

TESIS DOCTORAL

**METODOLOGÍA DE CONTROL INDEPENDIENTE DE
INSTRUMENTOS Y EXPERIMENTOS PARA SU
DESPLIEGUE EN LABORATORIOS REMOTOS**



UNIVERSIDAD DE DEUSTO

FACULTAD DE INGENIERÍA

Presentada por D. UNAI HERNÁNDEZ JAYO
dentro del Programa de Doctorado en INGENIERÍA INFORMÁTICA Y
TELECOMUNICACIÓN

Dirigida por el Dr. D. JAVIER GARCÍA ZUBÍA

El Director

El Doctorando

BILBAO, MAYO DE 2012

A mis Aitas
y
a Marta

*'Cause tramps like us,
baby we were born to run*

~Bruce Springsteen~

Resumen

Los laboratorios remotos actuales, especialmente los centrados en electrónica analógica, presentan unas altas prestaciones y fiabilidad de uso, pero para llegar a estos niveles han tenido que desarrollar soluciones propietarias que restringen su despliegue en otras universidades y países, lo que degrada la característica principal de un laboratorio remoto: la compartición de recursos y la sostenibilidad. La solución presentada mantiene los estándares de calidad de las implementaciones actuales, pero desde una perspectiva de estandarización que favorece el despliegue efectivo del laboratorio remoto fuera del ámbito en el que fue diseñado.

En el ámbito concreto de la electrónica analógica tiene lugar el desarrollo de este trabajo de investigación, el cual propone un método de control orientado a facilitar el despliegue y mantenimiento de la solución implementada al independizar los instrumentos y experimentos del laboratorio respecto del software desarrollado para su control.

Esta nueva metodología de control es el resultado de los estudios y análisis llevados a cabo en base a las soluciones ya implantadas y desplegadas en otros laboratorios remotos de referencia, dando lugar a una serie de desarrollos cuyo carácter innovador es aportado por la integración y combinación de distintas tecnologías y aplicaciones implementadas específicamente para este trabajo, obteniendo un método único y no desplegado hasta ahora en el campo de la experimentación remota.

Todas las soluciones y desarrollos realizados en este trabajo han sido evaluados en un escenario operativo real, lo que ha permitido validar el método propuesto y corroborar las mejoras derivadas de uso respecto a otras alternativas, obteniendo una serie de conclusiones que contribuyen a diseñar y obtener laboratorios remotos de electrónica analógica que permiten construir y caracterizar circuitos electrónicos con las mismas prestaciones que si las tareas de conexión y toma de medidas se realizaran en un laboratorio tradicional.

Abstract

Current remote laboratories, especially those devoted to Analog Electronics, present high quality features, are easy to use and reliable. However, this satisfactory performance level has been attained by means of developments that use proprietary solutions that hinder the deployment of remote laboratories in other institutions and countries, and thereby degrade their main characteristic; namely resource sharing and sustainability. While maintaining the quality standards of existing implementations, the solution presented herein, adopts a standardization perspective that facilitates the deployment of remote labs outside the context they were built in. scope in which it was created.

The current research work developed within this context, and specifically related to the teaching of Analog Electronics, proposes a control method that facilitates the deployment and maintenance of remote laboratories by making their instruments and experiments independent from the software applications that control them

This new methodology is the result of the analysis of previous solutions tested and validated in distinguished remote laboratories. The study led to a set of innovative developments combining and integrating different technologies and applications and specifically implemented as part of this doctoral thesis, giving as an end result a unique method never deployed before in the field of remote experimentation.

The improvements provided by the proposed method have been corroborated by validation of a prototype developed for, and tested on, real operating scenarios with real users.

The results of our research are an important contribution to the design and deployment of remote labs for Analog Electronics allowing users to build and characterize electronic circuits with the same performance as if the tasks of connection and measurement were carried out in a traditional hands-on lab.

Agradecimientos

En primer lugar quiero agradecer a Javier García Zubía que me invitara a unirme a él y formar parte del equipo de WebLab-Deusto. Con su ayuda elegimos el tema y encontramos el punto al que queríamos llegar al final de este trabajo. Pero sobretodo quiero darle las gracias por su apoyo, los medios de los que me ha provisto y el exhaustivo trabajo de corrección de este documento que ha llevado a cabo.

Gran parte del peso del WebLab-Deusto recae sobre los hombros de Pablo Orduña, quien me ha explicado mil y una veces sus entresijos y que además me ha ayudado a testear el sistema desarrollado incluyéndolo dentro de la estructura del WebLab-Deusto. ¡Gracias por todo tu tiempo Pablo! En este punto también quiero dar las gracias a Ingvar Gustavsson, Kristian Nilsson y Johan Zackrisson por su ayuda y colaboración en el proyecto VISIR.

Agradecer también a todos aquellos que me han ayudado a poder aislarme en la universidad y poder centrarme en el desarrollo de mi tesis, dejando quizás de lado algunas de mis tareas. También quiero dar las gracias a todos los compañeros que me preguntaban cada poco para cuándo acababa la tesis. Quizás las ganas de dejar de escuchar la pregunta también me ha motivado a acabar antes. Y motivación es lo que se necesita en una tarea como esta.

Gracias a todos los que participasteis en las pruebas y poder así recoger datos y números para realizar el análisis de funcionamiento del sistema. Sin ellos, parte de la experimentación no hubiera sido posible.

Rober, JuanMa gracias por vuestra compañía y amistad en la universidad y fuera de ella, por todo lo que hemos hecho juntos y más importante, lo que nos queda aún por hacer.

Alberto, Nekane, Chefo, JosuKa, Verónica, Inés, Anselmo, Begoña R., Pablo F., Pili, etc. etc. es un placer venir a trabajar y encontrarnos por los pasillos o en la mesa de enfrente.

Pero sobretodo, gracias, mil gracias a mis padres. Gracias a ellos, a su esfuerzo y a su compañía estoy donde estoy y llegaré a donde llegaré. Aita, especialmente esto va para ti. Creo que Tamara y yo nunca te agradeceremos lo suficiente todo el tesón, constancia y capacidad de sacrificio que con tu ejemplo nos has enseñado. Y Ama, gracias por parecerte tanto a quien te pareces... sería difícil añadir algo más...

Y finalmente a Marta. Todo comenzó con una planta y fíjate, Ayala casi ya está aquí. Gracias por ser mi compañía y soporte durante todo este camino, pues nada de esto hubiera sido posible sin tu tranquilidad y sonrisa. Gracias.

Índice

1	Introducción y contextualización.....	1
1.1	¿Qué es un laboratorio remoto?	6
1.1.1	Tipos de laboratorios remotos.....	9
1.1.2	Arquitectura de un laboratorio remoto	18
1.2	Contextualización del trabajo de investigación	22
1.2.1	Contexto tecnológico del trabajo de investigación	24
1.3	Descripción de la problemática	30
1.4	Hipótesis y objetivos.....	35
1.5	Metodología de investigación	38
1.6	Organización de la memoria	40
2	Recursos tecnológicos y antecedentes.....	43
2.1	Análisis de tecnologías de control	45
2.1.1	Tecnologías para el desarrollo del software de control.....	48
2.1.2	Interfaces de control de instrumentación.....	54
2.1.2.1	Selección del interfaz de control de instrumentos	63
2.1.3	Tecnologías para el control de experimentos	68
2.1.3.1	Selección del sistema de control de experimentos.....	73
2.1.4	Conclusiones.....	74
2.2	Antecedentes de metodologías de control en laboratorio remotos ..	76
2.2.1	NetLab. University of South Australia	77
2.2.1.1	Arquitectura del NetLab.....	78
2.2.1.2	Aplicación de control en NetLab.....	78
2.2.1.3	Control de experimentos e interfaz de control en NetLab ..	80
2.2.1.4	Resumen método de control en NetLab.....	82
2.2.2	RemotElectLab. Polytechnic Institute of Porto	83
2.2.2.1	Arquitectura del RemotElectLab	83

2.2.2.2	Aplicación de control en RemotElectLab.....	84
2.2.2.3	Control de experimentos e interfaz de control en RemotElectLab	87
2.2.2.4	Resumen método de control en RemotElectLab.....	93
2.2.3	ISILab. Universidad de Génova	93
2.2.3.1	Arquitectura del ISILab	94
2.2.3.2	Aplicación de control e interfaz de control en ISILab.....	96
2.2.3.3	Control de experimentos en ISILab	98
2.2.3.4	Resumen método de control en ISILab	99
2.2.4	Proyecto VISIR. Blekinge Institute of Technology.....	100
2.2.4.1	Arquitectura del VISIR.....	102
2.2.4.2	Aplicación de control en VISIR	103
2.2.4.3	Control de experimentos e interfaz de control en VISIR....	104
2.2.4.4	Resumen método de control en VISIR	108
2.2.5	iLab. Massachusetts Institute of Technology.....	109
2.2.5.1	Arquitectura del iLab.....	111
2.2.5.2	Aplicación de control en iLab.....	112
2.2.5.3	Control de experimentos e interfaz de control en iLab	113
2.2.5.4	Resumen método de control en iLab.....	114
2.2.6	Otras plataformas de experimentación remota.....	115
2.2.6.1	LEDFE. Universidad Nacional de Rosario	116
2.2.6.2	eVALIDATE. Dayalbagh Educational Institute.....	118
2.2.6.3	RwmLAB. Western Michigan University	120
2.2.6.4	Resumen de tecnologías usadas en estas metodologías	122
2.3	Análisis comparativo de las metodologías estudiadas.....	122
2.3.1	Tecnologías empleadas en cada método de control	124
2.3.2	Tipo de experimentación de cada método de control.....	127
2.3.3	Acciones posibles con cada método de control.....	135
2.3.4	Sistema de control empleado en cada método de control	138

2.4	Conclusiones	142
3	Descripción del método de control propuesto	149
3.1	Descripción general del método de control propuesto.....	150
3.1.1	Soluciones planteadas a los problemas descritos.....	154
3.1.1.1	Soluciones desarrolladas para la aplicación de control	154
3.1.1.2	Soluciones empleadas en el interfaz de control	164
3.1.1.3	Soluciones para el control de los experimentos	169
3.1.2	Resumen de soluciones y aportaciones del método de control propuesto	181
3.2	Aplicación software de control de la solución propuesta	184
3.2.1	Servidor de Medidas	184
3.2.1.1	Configuración del Servidor de Medidas.....	190
3.2.2	Servidor de Equipos y Experimentos.....	194
3.2.2.1	Configuración del Servidor de Equipos y Experimentos ..	202
3.2.3	Protocolo de comunicación entre ambos servidores	205
3.3	Sistema de control y configuración de experimentos de la solución propuesta.....	211
3.3.1	Ejemplo de configuración y realización remota de un circuito bajo prueba.....	217
3.3.1.1	Configuración de los ficheros del Servidor de Equipos y Experimentos	218
3.3.1.2	Configuración del <i>Circuit Solver</i> o Instructor virtual.....	221
3.3.1.3	Control de la matriz de conmutación	225
3.3.1.4	Módulos de expansión para componentes	230
4	Evaluación de la solución.....	235
4.1	Escenario de evaluación	236
4.1.1	Integración del método propuesto en WebLab-Deusto	237
4.2	Pruebas de evaluación del método propuesto.....	238
4.2.1	Operación con instrumentos intercambiables	240

4.2.2	Operación con usuarios y experimentos reales.....	244
4.2.3	Tiempos de respuesta del método de control.....	251
4.2.4	Comparativa de medidas respecto a escenario manual y laboratorio VISIR	256
4.2.5	Caracterización del sistema de control de experimentos	264
4.2.6	Rendimiento del sistema de control respecto al sistema empleado en el laboratorio VISIR	267
4.2.7	Medida de valores de tensión media en señales periódicas.....	275
4.3	Conclusiones de los resultados obtenidos.....	277
5	Conclusiones y líneas futuras de trabajo	281
5.1	Publicaciones realizadas durante el trabajo de investigación	285
5.2	Líneas futuras de trabajo.....	286
	Referencias bibliográficas	289
	Apéndices.....	307
A	Configuración y uso de drivers IVI	309
A.1	Uso de drivers IVI	312
A.2	Configuración de drivers IVI.....	315
A.3	Ejemplo de uso de drivers IVI para el control de un instrumento.....	321
B	Configuración de la red de instrumentos mediante LXI.....	323

Índice de Figuras

Figura 1.1: Distribución de laboratorios remotos por disciplinas	10
Figura 1.2: Arquitectura básica cliente-servidor de un laboratorio remoto.....	18
Figura 1.3: Contextualización del trabajo de investigación.....	22
Figura 1.4: Motivaciones de empleo y desarrollo de laboratorio remotos.....	24
Figura 1.5: Contexto tecnológico en un laboratorio remoto.....	25
Figura 1.6: Contexto experiencial del trabajo de investigación	26
Figura 1.7: Sistemas que forman el método de control propuesto	28
Figura 1.8: Pasos en el proceso de experimentación	29
Figura 1.9: Esquema de objetivos del trabajo de investigación realizado	32
Figura 1.10: Esquema de la metodología de investigación seguida.....	40
Figura 2.1: Tecnologías para el desarrollo de laboratorios remotos	46
Figura 2.2: Diagrama de componentes utilizados para el control de instrumentación.....	53
Figura 2.3: Alternativas para la configuración de la comunicación con los instrumentos	55
Figura 2.4: Evolución de los interfaces de control	60
Figura 2.5: Interfaz web de un generador de funciones LXI.....	61
Figura 2.6: Clases de instrumentos LXI	62
Figura 2.7: Ejemplo de matriz de conmutación	70
Figura 2.8: Ejemplo de sistema comercial de matriz de conmutación comercial.....	71
Figura 2.9: Página web del NetLab.....	77
Figura 2.10: Arquitectura del laboratorio remoto NetLab.....	78
Figura 2.11: Interfaz para el diseño de circuitos	79
Figura 2.12: Configuración de la matriz de conmutación NetLab	80
Figura 2.13: Estructura de la matriz de relés y capa de conmutación	81
Figura 2.14: Arquitectura software del RemotElectLab.....	84
Figura 2.15: Interfaz de usuario del RemotElectLab	85
Figura 2.16: Servidor de experimentos del RemotElectLab	86
Figura 2.17: Equipos y componentes del RemotElectLab.....	90
Figura 2.18: Circuito bajo prueba del RemotElectLab.....	91
Figura 2.19: Website del laboratorio remoto ISILab.....	94
Figura 2.20: Arquitectura del Laboratorio Virtual Compartido	95
Figura 2.21: Arquitectura Software del ISILab.....	95
Figura 2.22: Esquema XML de un fichero de configuración del ISILab	97
Figura 2.23: Placa base ISIBoard del laboratorio ISILab	98
Figura 2.24: Página Web principal del proyecto VISIR.....	101
Figura 2.25: Solución software de la plataforma VISIR	103
Figura 2.26: Instrumentos y servidor de equipos de la plataforma VISIR.	104
Figura 2.27: Configuración de los componentes disponibles en la plataforma VISIR.....	108
Figura 2.28: Arquitecturas disponibles en ISA	112
Figura 2.29: Estructura del Lab Server	114
Figura 2.30: Arquitectura del laboratorio LEDFE.....	117
Figura 2.31: Interfaz de la plataforma desarrollada en eVALIDATE	119
Figura 2.32: Interfaz de usuario del RwmLAB.....	121

Figura 2.33: Tipos de experimentos en función de su complejidad.....	132
Figura 3.1: Problemáticas a atajar mediante el método de control propuesto	151
Figura 3.2: Arquitectura física del método propuesto.....	152
Figura 3.3: Funciones de la aplicación de control contempladas en el método de control propuesto.....	153
Figura 3.4: Contexto de la aplicación de control en el método desarrollado	156
Figura 3.5: Arquitectura lógica de la aplicación de control de la solución propuesta.	159
Figura 3.6: Flujo de información para el control software de los experimentos	163
Figura 3.7: Configuración de la red de instrumentación mediante LXI.....	167
Figura 3.8: Configuración de la matriz 8x32 resultante.....	172
Figura 3.9: Ejemplo de toma de medidas de tensión sobre señal periódica	175
Figura 3.10: Ejemplo de circuitos formados con tres resistencias	177
Figura 3.11: Ejemplo de configuración de circuitos con diodos.....	178
Figura 3.12: Circuito ejemplo para la medida de intensidades	179
Figura 3.13: Arquitectura lógica del Servidor de Medidas	185
Figura 3.14: Secuencia de ejecución de procesos en el Servidor de Medidas.....	190
Figura 3.15: Arquitectura lógica del Servidor de Equipos y Experimentos	196
Figura 3.16: Llamada al algoritmo de control de cada instrumento.....	199
Figura 3.17: Secuencia de ejecución de procesos del Servidor de Experimentos y Equipos.	201
Figura 3.18: Pasos para la configuración del Servidor de Equipos y Experimentos	204
Figura 3.19: Representación de un esquema eléctrico	212
Figura 3.20: Sistema de interconexión de las matrices de conmutación	213
Figura 3.21: Ejemplo de creación de un circuito simple.....	215
Figura 3.22: Ejemplo de circuito: rectificador de media onda	218
Figura 3.23: Instructor virtual del Servidor de Medidas	222
Figura 3.24: Ejemplo de conexión serie de 2 componentes en la matriz de conmutación.....	227
Figura 3.25: Ejemplo de formación de comandos de control de la matriz.....	228
Figura 3.26: Procesos para la construcción física del circuito	229
Figura 3.27: Esquema de conexionado de los módulos de expansión	230
Figura 3.28: Vista frontal y trasera de los módulos de expansión	231
Figura 3.29: Imágenes de los módulos de expansión diseñados y desarrollados.....	232
Figura 4.1: Integración del método de control propuesto en WebLab-Deusto	238
Figura 4.2: Equipos utilizados en las pruebas del método de control propuesto.....	241
Figura 4.3: Paneles frontales de instrumentos del laboratorio	243
Figura 4.4: Dispersión de error en medidas prueba 1.....	251
Figura 4.5: Tiempos de respuesta del sistema de control respecto a diferentes tipos de circuitos.....	253
Figura 4.6: Tiempo de respuesta sistema VISIR vs. Sistema propuesto.....	256
Figura 4.7: Dispersión error en medida de frecuencia y tensión AC.....	259
Figura 4.8: Dispersión error en medida de resistencias	261
Figura 4.9: Dispersión error en medida tensión DC	262
Figura 4.10: Esquema de medida para caracterización de resistencias	264
Figura 4.11: Ejemplo de relé cerrado mediante servicio web del mainframe 34480 ^a	265
Figura 4.12: Ejemplo de interferencia de resistencias parásitas en medida de resistencias ..	266
Figura 4.13: Ejercicio de rendimiento del sistema de control propuesto	268

Figura 4.14: Medidas de intensidad adicionales en el test de rendimiento	268
Figura 4.15: Componentes y cortocircuitos necesarios en el laboratorio VISIR	269
Figura 4.16: Ejemplo de conexionado de componentes de la plataforma VISIR.....	269
Figura 4.17: Relación componentes necesarios para la realización de circuitos.....	274
Figura 4.18: Componentes y cortocircuitos necesarios para la realización de circuitos.....	274
Figura A.1: Evolución de los estándares de control de instrumentos	309
Figura A.2: Arquitectura lógica del método de control propuesto	314
Figura A.3: Arquitectura lógica del IVI Configuration Store.....	316
Figura A.4: Configuración del alias VISA del instrumento	317
Figura A.5: Configuración del Recurso Hardware del driver	318
Figura A.6: Configuración del Módulo Software del driver.....	319
Figura A.7: Configuración del Nombre Lógico del driver	320
Figura A.8: Configuración de dos instrumentos con mismo Nombre Lógico.....	321
Figura A.9: Ejemplo de uso de driver IVI.....	322
Figura B.1: Ejemplo pantalla de configuración de red en un equipo LXI	325

Índice de Tablas

Tabla 1.1: Definiciones de laboratorios remotos.....	6
Tabla 1.2: Caracterización de laboratorios.....	7
Tabla 1.3: Ejemplos de laboratorios en base al tipo de experimentación remota posible	13
Tabla 1.4: Ejemplos de laboratorios en función de las actividades remotas posibles.....	15
Tabla 2.1: Análisis comparativo entre las tecnologías de control para laboratorios remotos..	65
Tabla 2.2: Resumen tecnologías empleadas por en el método de control NetLab.....	82
Tabla 2.3: Resumen tecnologías empleadas por en el método de control RemotElectLab	93
Tabla 2.4: Resumen tecnologías empleadas por en el método de control ISILab.....	100
Tabla 2.5: Resumen tecnologías empleadas por en el método de control VISIR.....	109
Tabla 2.6: Resumen tecnologías empleadas por en el método de control iLab	115
Tabla 2.7: Resumen tecnologías empleadas en LEDFE, eVALIDATE y RwmLab.....	122
Tabla 2.8: Tecnologías de control hardware y adaptación a nuevos instrumentos	127
Tabla 2.9: Tipo de experimentos e interacción en cada plataforma y tolerancia a fallos	130
Tabla 2.10: Tipo de experimentos de cada plataforma y complejidad de los mismos	134
Tabla 2.11: Acciones habilitadas en los métodos de control analizados	137
Tabla 2.12: Sistema de control de experimentos empleados y características	141
Tabla 2.13: Especificaciones a satisfacer por el método de control deseado.....	145
Tabla 3.1: Tecnologías empleadas en el método de control propuesto.	154
Tabla 3.2: Soluciones para independizar la aplicación de control de los instrumentos	159
Tabla 3.3: Soluciones para independizar el software de control de los experimentos	164
Tabla 3.4: Soluciones adoptadas en el interfaz de control.....	170
Tabla 3.5: Soluciones adoptadas en el sistema de control de experimentos	180
Tabla 3.6: Identificadores de los instrumentos en el método propuesto.....	206
Tabla 3.7: Formas de onda definidas por el estándar IVI para generadores de señales.....	208
Tabla 3.8: Comandos para construir un rectificador de media onda y filtro a la salida	229
Tabla 4.1: Conjunto de pruebas y objetivos y soluciones a verificar.....	239
Tabla 4.2: Juego de ejercicios utilizados con usuarios reales	245
Tabla 4.3: Resultados obtenidos sobre el ejercicio 1, medidas con osciloscopio	247
Tabla 4.4: Resultados obtenidos en ejercicios 2 y 3 sobre medidas de resistencias.....	248
Tabla 4.5: Resultados obtenidos en el ejercicio 4 sobre medidas de tensiones	249
Tabla 4.6: Resultados obtenidos en el ejercicio 5 sobre medidas de frecuencia	250
Tabla 4.7: Tiempos de respuesta del sistema de control propuesto y el sistema VISIR	254
Tabla 4.8: Comparativa realizada respecto a la plataforma VISIR en el ejercicio 1	258
Tabla 4.9: Comparativa realizada respecto a la plataforma VISIR en el ejercicio 5	258
Tabla 4.10: Comparativa realizada respecto a la plataforma VISIR en los ejercicios 2 y 3.....	260
Tabla 4.11: Comparativa realizada respecto a la plataforma VISIR en el ejercicio 4	262
Tabla 4.12: Caracterización del sistema de control de experimentos.....	265
Tabla 4.13: Valores obtenidos para la validación de la prueba de rendimiento del sistema. .	271
Tabla 4.14: Set de circuitos para prueba de rendimiento de la matriz de conmutación.....	273
Tabla 4.15: Medida de tensión media de señal periódica con Offset 0V	276
Tabla 4.16: Medida de tensión media de señal periódica con Offset 1V	277
Tabla A.1: Clases de instrumentos y drivers definidos en el estándar IVI	311

*Capítulo***1****Introducción y
contextualización**

Aunque la educación a distancia tiene sus orígenes a mediados del siglo XIX, ha sido la gran revolución tecnológica causada por el nacimiento de Internet a comienzos de los años 90 la que ha propiciado la expansión y la llegada de la educación a distancia a casi cualquier parte del mundo. Con este nuevo modelo educativo surge la necesidad no solo de poner al alcance del alumno contenidos teóricos, sino también contenidos prácticos, lo que ha favorecido el desarrollo y despliegue de los laboratorios remotos como herramienta de aprendizaje para facilitar el acceso a distancia de los contenidos y elementos prácticos de las materias.

Así pues, el primer capítulo de esta tesis doctoral tiene como objetivo definir el marco en el cual se ha desarrollado el trabajo de investigación que aquí presentamos. Para ello vamos a proporcionar una descripción de qué es un laboratorio remoto, identificando cada uno de los niveles o capas que forman su arquitectura. A partir de esta estructura mostraremos los niveles en los que se ha centrado este trabajo y posteriormente describiremos la problemática existente en dicho dominio. Este capítulo finaliza describiendo las hipótesis y objetivos a alcanzar, así como la metodología de investigación seguida para tal fin.

Los primeros cursos a distancia de los que se tiene constancia tuvieron lugar en 1840 cuando Sir Isaac Pitman, el inventor de la taquigrafía, tuvo la idea de comenzar a dar cursos por correspondencia [Mat99]. Poco tiempo después, la Universidad de Londres fue el primer centro universitario en el mundo en ofrecer cursos a distancia a través de su *External System*, establecido en el año 1858 [Uni10], el cual, a pesar del éxito alcanzado, el tipo y el número de cursos ofrecidos, permaneció limitado a unas pequeñas áreas de conocimiento, principalmente debido al poco grado de interacción entre el instructor o profesor y el estudiante, ya que en aquella época, el intercambio de materiales únicamente era posible a través de cartas manuscritas, cuya entrega dependía siempre del servicio postal de correos.

En la segunda mitad del siglo XX, el panorama de los cursos a distancia comenzó a cambiar con la fundación en 1969 de la *United Kingdom's Open University* (OU), la cual ofrecía cursos en los cuales se utilizaban unas técnicas de enseñanza mixtas basadas en cursos presenciales y a distancia. A pesar de tener un funcionamiento similar a los cursos ofrecidos por la Universidad de Londres, en este caso los cursos eran mucho más diversos y contaban con textos y materiales cuidadosamente confeccionados así como de grabaciones de audio y video, complementadas con emisiones abiertas de radio y televisión. Adicionalmente, se organizaban sesiones a través de comunicaciones telefónicas en las que participaban el profesor y un alumno o bien un grupo de alumnos, por lo que podían mantener una comunicación fluida entre ellos [SAG10].

Sin embargo, no fue hasta comienzos de la década de los 90 cuando se introdujo el concepto de e-learning y tuvo lugar la aparición de los primeros laboratorios remotos [AB96] [AMM91]. La idea del e-learning (anglicismo de aprendizaje electrónico) estaba basada en el uso de un nuevo y potente medio de comunicación: Internet, el cual abría las puertas por primera vez a una interacción a gran escala entre el estudiante y el profesor. Internet en un primer momento facilitó la virtualización de los contenidos de los libros, haciendo posible su consulta a través de la red desde cualquier parte del mundo y en cualquier momento. Así comenzaron a utilizarse las herramientas tecnológicas asociadas a Internet para desarrollar soluciones que permitieran realzar y mejorar el acceso al conocimiento y al aprendizaje [Ros01]. Internet se afianzó como una herramienta global para la diseminación de la información y un medio para la colaboración y la

interacción entre las personas y las computadoras sin importar su localización geográfica [LCC+10].

Gracias a Internet, la educación a distancia ha ido convirtiéndose poco a poco en una alternativa a veces y en otra, en un complemento a la educación presencial de los métodos tradicionales de enseñanza, dando lugar al concepto de aprendizaje semipresencial o más conocido en su término anglosajón *blended-learning* o simplemente b-learning. Este auge ha sido debido a las principales ventajas del e-learning: los estudiantes se benefician gracias a los cursos a distancia de una reducción o eliminación total de desplazamientos, por lo que las oportunidades de acceso a los recursos educativos aumentan, así como la disponibilidad temporal de los mismos. Además, mediante las tecnologías puestas al servicio de la educación a distancia, aquellas actividades que requieren la colaboración entre alumnos pueden realizarse también de manera remota, añadiendo la libertad de horarios, por lo que las actividades pueden ser fácilmente planeadas y ejecutadas en el tiempo [GNR05].

Por consiguiente y gracias a las tecnologías actuales que posibilitan un acceso a la información de una forma cada vez más rápida y ubicua [Bab06], la educación a distancia y las plataformas de aprendizaje online están aumentando rápidamente. Como ejemplo, en el año 2004 la matriculación en cursos a distancia en las universidades de Estados Unidos aumentó en 2,4 millones de alumnos, un 18% más que en el año anterior [Bak06].

En el caso de España, en el curso 2004-2005 la Universidad Nacional de Educación a Distancia (UNED) aumentó el número de matriculaciones un 2,28% respecto al año anterior, que junto a la Universidad Politécnica de Valencia (UPV), la cual tuvo un incremento del 1,59%, fueron las únicas universidades públicas que aumentaron su número de alumnos. En el caso de las universidades privadas, en el mismo año, la única universidad que presentó un aumento de matriculaciones fue la Oberta de Catalunya, que lo hizo en un 17,9% [Edu05]. Como se puede discernir, las universidades que más aumentaron (y casi las únicas, a excepción de la UPV) son aquellas que ofertan titulaciones mediante cursos a distancia. En el curso 2006-2007 se seguía manteniendo la misma tendencia. Comparando los datos de los cursos 2004/2005, 2005/2006 y 2006/2007 la Universidad Nacional de Educación a Distancia (UNED) encabeza la lista de universidades con mayor

número de alumnos matriculados, con 140.529, por delante de la Complutense de Madrid (78.510), la Universidad de Sevilla (58.202) y la de Granada (56.215). Las dos universidades a distancia (UNED y Universitat Oberta de Catalunya) representan el 12,7% del total de matrículas, un 0,8% más que en el curso anterior [Est08]. En cambio, datos más recientes indican un descenso en el total de matriculaciones de alumnos en las universidades españolas, pasando de ser 1.371.810 en el curso 2008-2009 a 1.214.746 en el curso 2009-2010 (un 11% menos). Aún así, sigue habiendo una gran diferencia entre la UNED con 114.360 alumnos matriculados en el curso 2009-2010 (un 20% menos que el curso anterior) con la siguiente universidad con más matriculaciones, la Universidad Complutense de Madrid con 63.239 (un 15% menos) o la Universidad de Granada con 56.356 (siendo una de las pocas universidades que han experimentado un leve aumento de alumnos con 325 más respecto al curso anterior), manteniendo la Universidad Oberta de Catalunya la misma progresión que el resto de universidades con 33.601 alumnos (un descenso del 15% en las matriculaciones) [Est10].

El desarrollo tecnológico de laboratorios remotos ha presentado avances significativos en los últimos años a tenor de los numerosos proyectos de investigación que se desarrollan en este ámbito, auspiciados por la Unión Europea, como por ejemplo las iniciativas DYNACORE [RGR+99], MARVEL [MF03], PEARL [SCC+04], CYBERLAB [ZE08] y PEMCWebLab [BFR08]. El impacto de las tecnologías de la información y la comunicación ha propiciado la creación de redes educativas, como es el caso de la European Schoolnet, la red de excelencia PROLEARN (www.prolearn-online.com) o las iniciativas directamente enfocadas en el desarrollo y difusión de la experimentación remota, como es el caso de la red australiana de laboratorios remotos Labshare (<http://www.labshare.edu.au/>) y el consorcio de universidades unidas en torno al proyecto VISIR [GZH+07]. A nivel internacional destacan el Global Online Laboratory Consortium (GOLC, <http://online-lab.org/>), la Red de Laboratorios Remotos RexNet (<http://www.rexlab.net/>) y especialmente la plataforma iLab (<http://ilab.mit.edu/>) desarrollada por el MIT. También encontramos en la Red repositorios web de laboratorios remotos como es el caso del proporcionado por el proyecto europeo LiLa (<http://www.lila-project.org>) y Lab2go (<http://www.lab2go.net/>), cuyo objetivo es proporcionar un entorno web común para compartir recursos y experiencias relativas a la

experimentación remota, tanto para desarrolladores, como profesores y estudiantes de diversas áreas de conocimiento.

La experimentación remota también ha sido un tema sujeto a numerosas publicaciones durante los últimos años. En un estudio comparativo sobre la temática versada en laboratorios remotos y tradicionales y realizada la búsqueda en tres bases de datos (IEEE, ACAM, y ScienceDirect), se encontró que un área importante de trabajo engloba los aspectos técnicos del desarrollo de laboratorios remotos [MN06]. Al mismo tiempo, se han desarrollado congresos y publicaciones periódicas orientadas exclusivamente a la ingeniería y experimentación remota, como es el caso del *International Conference of Remote Engineering & Virtual Instrumentation* (REV, <http://www.rev-conference.org/>) patrocinado por la *International Association of Online Engineering* (IAOE), y su publicación periódica el *International Journal of Online Engineering* (iJOE). Respecto al Institute of Electrical and Electronics Engineers (IEEE), asociación internacional de referencia, podemos citar los *Transactions on Learning Technologies*, publicación indexada en la *ISI Web of Knowledge*, así como los *special issues* dedicados a la experimentación remota de publicaciones como el *Transactions on Industrial Electronics*, el *Transactions on Industrial Informatics* o el *Transactions on Education*. Y por otra parte, la editorial IGI Global (<http://www.igi-global.com>), especializada en publicaciones de divulgación científica cuenta en su catálogo con una línea editorial dedicada a temas relacionados con el e-learning y cuenta entre sus números con algunos específicos relativos a los laboratorios remotos [AAJ12] [RZH+12] [OGL+11].

Así pues, cada vez más la educación a distancia en el ámbito universitario está teniendo más repercusión en el panorama tanto nacional como internacional. Pero la transición de un sistema educativo presencial a un entorno remoto o virtual no es sencilla y presenta paradigmas con sus propias dificultades [Sch98]. Ciertas disciplinas académicas no requieren grandes recursos tecnológicos para poner a disposición de los alumnos sus contenidos y prácticamente con la virtualización de sus contenidos es suficiente. Pero en el caso de las ciencias técnicas y las ingenierías las necesidades van más allá, principalmente por su alto contenido de formación práctica. Si es sencillo proveer a un alumno de contenidos teóricos a través de un entorno web, no lo es tanto cuando se desea proporcionar al alumno entornos y herramientas para acceder a dicha formación práctica, ya que las

necesidades y los requisitos respecto a una formación eminentemente teórica son muy distintos [GS08]. Es por tanto en este escenario en el que hacen aparición los laboratorios remotos como herramientas puestas al servicio de la educación a distancia.

1.1 ¿Qué es un laboratorio remoto?

Pero, ¿qué es un laboratorio remoto? Se pueden encontrar numerosas definiciones de lo que es un laboratorio remoto tal y como se constata en la Tabla 1.1, elaborada en base a los investigadores de referencia en esta área de conocimiento. Como vemos, no existe una definición que sea universalmente aplicable y entendible cuando se habla de conceptos como laboratorio remoto, weblab (o web-lab), laboratorio virtual, tele-lab y online lab. Es más, muchas veces estos términos son usados como sinónimos e incluso en algunas publicaciones no queda claro cuáles son las diferencias entre todas estas herramientas.

Definición	Referencia
Son plataformas basadas en Internet que proporcionan acceso a bancos de pruebas experimentales reales, no disponibles en el lugar físico en el que se encuentra el usuario	[GB09]
Son laboratorios que utilizan datos reales obtenidos desde la experimentación realizada a través de un interfaz web.	[CNE ⁺ 04]
La idea general de un laboratorio remoto es habilitar el acceso a los laboratorios físicos o a puestos desde lugares distantes a los mismos, haciendo uso de una infraestructura de comunicación adecuada.	[ME06]
Un laboratorio remoto se corresponde a aquella situación en la que el control y la observación de los instrumentos y del objeto bajo prueba del experimento son proporcionados por un ordenador, el cual es accedido remotamente gracias a una red de comunicaciones específica	[AGS ⁺ 07]
Los laboratorios remotos son instalaciones experimentales que se pueden acceder a través de Internet, permitiendo a los estudiantes y educadores llevar a cabo experimentos desde cualquier lugar del mundo en cualquier momento.	[Ala04]
Un laboratorio remoto es un sistema de apoyo a la docencia basado en un laboratorio virtual y de tele-presencia accesible a través de una red basada en protocolos TCP/IP que permita al alumno practicar de una forma lo más similar posible a como si estuviese en las dependencias del laboratorio, dándole la posibilidad de manejar las simulaciones o interactuar con las plantas reales.	[DSM00]

Tabla 1.1: Definiciones de laboratorios remotos

		Acceso	
		Local	Remoto
Equipo	Físico (Real)	Laboratorio Manual	Laboratorio Remoto
	Virtual (Simulado)	Laboratorio Virtual	Laboratorio Virtual Remoto

Tabla 1.2: Caracterización de laboratorios

Realmente, encontrar una definición es una tarea complicada. Casi todos los laboratorios relacionados con la ingeniería llevan asociados el uso de un computador. Incluso hoy en día, hay muchos equipos de laboratorio que son controlados y manejados a través de un interfaz que reside en un ordenador. En esta situación, el modo en que se accede al equipo del laboratorio no difiere mucho en el caso en que el alumno se encuentre físicamente junto al equipo o bien acceda a su control mediante ese interfaz virtual, pero de manera remota.

Así pues y con el objetivo de diferenciar los tipos de laboratorios mencionados previamente e identificar aquel en el cual se va a desarrollar este trabajo de investigación, se va a realizar una comparativa entre laboratorio manual y virtual (simulado) y laboratorio local y distribuido. Los criterios siguientes permitirán establecer una primera orientación respecto a la naturaleza del laboratorio (físico o virtual) y el modo de acceso de los alumnos (local o remoto), lo que permite analizar la Tabla 1.2 [Dor02], y extraer las siguientes reflexiones:

- Dependiendo del tipo de interacción del usuario con el experimento, ésta puede ser de dos tipos:
 - El usuario puede controlar directamente los dispositivos y la instrumentación, interacción que se corresponde con la llevada a cabo en el laboratorio tradicional.
 - El usuario controla los dispositivos y la instrumentación a través del interfaz proporcionado por un ordenador, utilizando instrumentación virtual o entornos mixtos virtuales-reales.
- Atendiendo a la naturaleza de los experimentos que se pueden llevar a cabo en el laboratorio, estos podrán estar basados en la utilización de dispositivos y equipos físicos o en modelos simulados de los dispositivos o equipos físicos.

- Y finalmente, se contemplan dos tipos de situaciones para la localización de los usuarios y los laboratorios, a saber, ambos en la misma localización o bien en diferentes localizaciones, es decir, separados geográficamente.

Este trabajo de investigación se centra en dar solución a unas problemáticas concretas que aparecen en aquella situación en la que un usuario requiere el acceso remoto a los recursos físicos del laboratorio, es decir, a través de un laboratorio remoto. En un escenario típico de experimentación remota, un alumno está en una localización diferente a donde se encuentran los recursos físicos del laboratorio y utiliza un acceso web para interactuar con ellos. Es decir, un laboratorio remoto lleva intrínseco el concepto de acceso a los dispositivos físicos desde un lugar situado lejos de ellos, gracias a una infraestructura de comunicaciones y sistemas de control de los experimentos y equipos desarrollados a tal efecto, con el objetivo de examinar y/o analizar su respuesta ante una serie de cambios generados bien por el propio sistema o bien por el propio alumno [CDF02]. Veremos más adelante como la naturaleza, el tipo y las propias tecnologías utilizadas en el desarrollo del laboratorio determinarán las acciones que el alumno podrá realizar durante su sesión práctica en el laboratorio de manera remota [GOL⁺07].

Un laboratorio remoto se trata de una herramienta de aprendizaje usada principalmente como soporte en plataformas educativas relacionadas con las ciencias y las ingenierías, en aquellos casos en los que desarrollar ciertos trabajos en las instalaciones del campus no es fácil (disponibilidad de equipos, horarios, etc.), o bien el laboratorio remoto se ofrece como un complemento a las sesiones de laboratorio presenciales, permitiendo a los estudiantes repetir un ejercicio o un experimento de manera remota, tantas veces como deseen, eliminando las restricciones en cuanto a distancia y tiempo, ya que a través de Internet los alumnos pueden acceder al laboratorio cuando quieran y desde donde quieran [PA07] [GPO⁺10].

De este modo y tratando de dar respuesta a la pregunta planteada al principio de esta sección, un laboratorio remoto puede ser definido como un recurso educativo emergente, que facilita la distribución remota de contenidos prácticos a través de la web y permite al usuario configurar, controlar y/o monitorizar parámetros físicos reales del experimento bajo prueba.

1.1.1 Tipos de laboratorios remotos

Prácticamente hay tantos tipos de laboratorios remotos como materias de enseñanza, ya que se podría llegar a desarrollar un laboratorio remoto que posibilitara el acceso a prácticamente cualquier tipo de experimento, ya sea en el campo de la química [CSS+06], de la física [CPA+00], de la robótica [CJB+06], o incluso de la energía nuclear como en el caso del laboratorio remoto basado en un reactor nuclear ofrecido por el MIT [HDB+08]. Esto significa que el uso de los laboratorios remotos no está restringido para su aplicación en un único entorno educativo, sino que puede utilizarse para controlar más equipos que aquellos que son controlados habitualmente por un ordenador. Así tal y como observamos en la Figura 1.1, podemos encontrar laboratorios remotos en muchas áreas de conocimiento, siendo la automática, la electrónica y la robótica las disciplinas de las que más laboratorios se encuentran referencias en la literatura especializada [GA11] [GRB10] [REV10] [GB09] [REV09] [GFB+08] [GFB+08].

Como se puede ver, los resultados corroboran las ideas tratadas anteriormente: la gran mayoría de laboratorios remotos están relacionados con carreras técnicas de ingeniería, en las que es importante y necesario que los alumnos lleven a cabo ejercicios prácticos de laboratorio.

En el repositorio web de laboratorios remotos Lab2go mencionado anteriormente y accesible en <http://www.lab2go.net/>, se puede ver también como la gran mayoría de laboratorios aquí recogidos pertenecen también a disciplinas relacionadas con las ciencias y la ingeniería.

Este trabajo de investigación va a estar concentrado en aquellos laboratorios remotos que han sido diseñados para su uso en el ámbito de la electrónica analógica, y es por tanto esta temática en la que se va a realizar una clasificación de los tipos de laboratorios actualmente existentes. Esta catalogación servirá para posteriormente enmarcar cada uno de los laboratorios remotos que se van a analizar y así contextualizar también la solución propuesta.

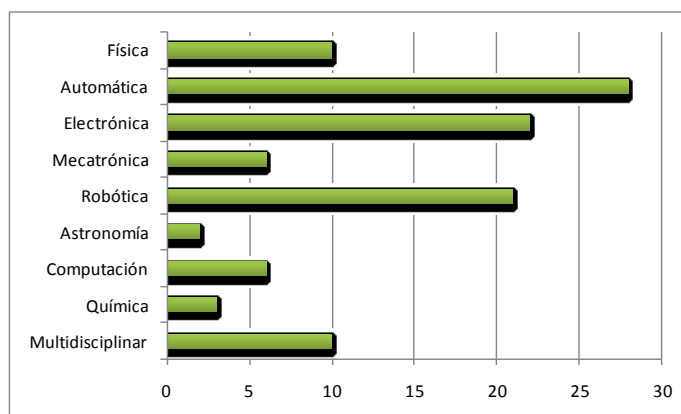


Figura 1.1: Distribución de laboratorios remotos por disciplinas

Realmente no es posible hacer una única catalogación de los laboratorios que actualmente se encuentran activos, pues los criterios para realizarla pueden ser muy dispares, desde los meramente temáticos hasta los que ahondan en las tecnologías de desarrollo utilizadas. Es por ello que a través de la información recopilada a este respecto hemos realizado una serie de clasificaciones en las que puedan acomodarse prácticamente todos los desarrollos realizados en el ámbito de la experimentación remota orientada a la electrónica.

A) Clasificación en función del tipo de experimentación remota.

Esta manera de ordenar los laboratorios remotos hace referencia al modo en que se le ofrecen al usuario las herramientas disponibles en el laboratorio y por tanto el tipo de experimentación ofrecida. Esta clasificación profundiza en la anteriormente realizada y que aparece reflejada en la Tabla 1.2, donde se caracterizan los laboratorios en función de su naturaleza (local o remota). Así pues, si dejamos a un lado los laboratorios tradicionales o presenciales nos encontramos tres tipos de laboratorios: laboratorios virtuales, laboratorios remotos y laboratorios híbridos o laboratorios remotos virtuales [GB09] [MN06].

Los laboratorios virtuales tienen como principio de funcionamiento la simulación remota y por tanto ofrecen un entorno de experimentación seguro que no requiere que el usuario tenga conocimientos avanzados sobre el comportamiento del experimento. Esto ayuda a inicializar al alumno en ciertos conocimientos básicos, evitando la incertidumbre o incluso el temor a

dañar los equipos que puede sentir al estar frente a instrumentos reales del laboratorio.

Este tipo de laboratorios puede ser accesible a través de un sitio web en el cual se ejecuta la simulación, o bien pueden ser laboratorios más interactivos en los que un ordenador juega el papel de servidor conteniendo una aplicación que simula un equipo o un sistema, a la que los alumnos se conectan, bien desde el propio laboratorio o bien desde un lugar remoto. Un ejemplo de este tipo de laboratorios es utilizado en la Universidad de Maribor para el aprendizaje del control de robots industriales [Tza09].

Aunque este tipo de laboratorios no son tan efectivos como la experimentación manual, una ventaja obvia de este tipo de experimentación es la flexibilidad y el amplio abanico de ejercicios prácticos que se les ofrece a los alumnos, ya que les permite actuar sobre sistemas que podrían ser complejos o encontrarse en entornos adversos sin estar expuestos a ningún tipo de peligro para ellos o los equipos, y que serían complicados de llevar a cabo en un laboratorio de la universidad. Al mismo tiempo, los estudiantes pueden repetir una y otra vez el experimento tantas veces como quieran con la oportunidad de ir aprendiendo de sus propios errores.

Por otro lado, en muchas ocasiones los paquetes software de simulación se encuentran adaptados al entorno pedagógico incluyendo manuales, tutoriales o secuencias explicadas paso a paso que ayudan al estudiante a centrarse en los conceptos importantes del experimento [DP07]. El proyecto "*Distance Learning on Digital Systems*", desarrollado por la Tallin University of Technology y la Technical University of Ilmenau, es un ejemplo de este tipo de entornos basados en simulaciones software. En sus páginas web se dispone de contenidos teóricos, textos con hipervínculos a otras fuentes de información, aplicaciones interactivas basadas en experimentos prácticos, pruebas de test para la autoevaluación de los alumnos y diferentes ejercicios que orientan al alumno durante su autoaprendizaje [UJK⁺07]. A favor de este tipo de experimentación se encuentra la reducción de costes, ya que por medio de un único paquete de simulación y del desarrollo de herramientas para hacer posible su acceso on-line se puede dar servicio a un número elevado de alumnos.

Los laboratorios remotos "manuales", o simplemente laboratorios remotos, proporcionan a los estudiantes acceso a los recursos físicos de un

laboratorio a través de Internet. Este tipo de laboratorios ya se han tratado al comienzo de esta sección en profundidad, por lo que no ahondaremos más en sus características y posibilidades.

Finalmente, los laboratorios híbridos o laboratorios remotos virtuales son una combinación de los dos tipos de laboratorios anteriores. Entre los diferentes tipos de experimentación remota disponibles, algunos autores consideran a este tipo de laboratorios como los más eficientes desde el punto de vista educativo y de investigación [GB09]. Por una parte, estos laboratorios proporcionan un amplio repertorio de experimentos frente a las limitaciones que muchas veces aparecen al tratar de hacer disponible a través de Internet ciertos ejercicios y sistemas prácticos. Estos laboratorios permiten además reducir los riesgos de mal uso de equipos y experimentos remotos debido a la inexperiencia del usuario, haciendo que primeramente practique con simuladores que recrean el posterior ejercicio a desarrollar sobre los equipos físicos. Así, en muchas situaciones es difícil distinguir si la aplicación web que se está ejecutando está actuando sobre un equipo físico controlado así remotamente o bien se trata solamente un simulador web. De hecho, esto aporta una ventaja ya que ciertos laboratorios pueden recurrir a un modo de operación que les permite intercambiar el control sobre un sistema físico o bien actuar sobre el mismo sistema de manera simulada y controlada. Este es el caso del laboratorio SMCRVI, el cual combina LabVIEW y Matlab para mostrar de manera didáctica a los alumnos cómo se realiza el control de un proceso industrial [CAP05].

De manera resumida, la Tabla 1.3 muestra un ejemplo de cada uno de estos laboratorios, en los cuales el alumno siempre utiliza un ordenador como medio para interactuar con el experimento, ya sea éste virtual, real o una mezcla de ambos sistemas.

B) Clasificación en función de las actividades remotas posibles

Según la clasificación realizada por el Dieter Müller [ME07], los laboratorios remotos se pueden catalogar en función de los siguientes tipos de acciones llevadas a cabo por el usuario: acciones relativas a la lectura remota de información, acciones de escritura remota y acciones de lectura y escritura remota.

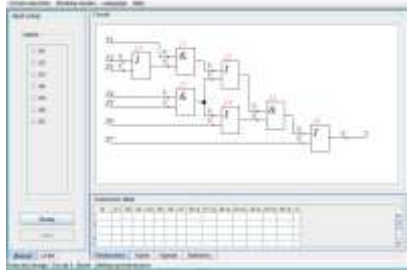
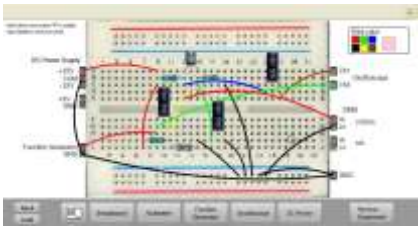
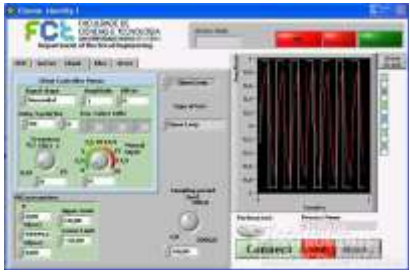
Tipo de experimentación remota	Ejemplo laboratorio	Web
<p>Laboratorios virtuales</p> <p>Basados en simulaciones realizas de manera remota</p>		<p>DILDIS Project, Universidad de Tallin e Ilmenau [UJK*07] http://www.pld.ttu.ee/dildis/</p> <p>Laboratorio virtual orientado al aprendizaje de testeo y diagnóstico de circuitos digitales de microelectrónica</p>
<p>Laboratorios remotos</p> <p>Permiten interactuar de manera remota sobre recursos físicos reales</p>		<p>Laboratorio VISIR [GZH*07] http://openlabs.bth.se/electronics/</p> <p>Permite crear circuitos de electrónica analógica de manera remota y testarlos utilizando instrumentación real de laboratorio</p>
<p>Laboratorios híbridos</p> <p>Combinan simulaciones en entornos virtuales e interacción con dispositivos reales</p>		<p>Laboratorio remoto SMCRVI, Universidad Uninova [CA05]</p> <p>Permite controlar de manera remota procesos industriales mediante aplicaciones en LabVIEW y MatLab</p>

Tabla 1.3: Ejemplos de laboratorios en base al tipo de experimentación remota posible

Los laboratorios que permiten únicamente la lectura de información son los denominados laboratorios de instrumentación, en los que el alumno es un mero observador del experimento y no puede realizar ninguna acción sobre el mismo. Generalmente se tratan de aplicaciones en las que se muestra algún fenómeno físico y que son usadas por el profesor como complemento adicional a la explicación teórica [BPP+07].

Otro tipo de laboratorios es aquel que permite la modificación o “escritura” de algunos de sus parámetros. En ellos el usuario dispone de las herramientas necesarias para realizar cambios bien en los instrumentos que forman parte del ejercicio práctico, o bien en los propios componentes del experimento y así advertir los cambios que sus acciones producen en el comportamiento del experimento. En estos laboratorios el objetivo es que el

usuario tenga la oportunidad de cambiar la ejecución del experimento para analizar y evaluar su comportamiento ante los cambios ocasionados. Un ejemplo de estos experimentos se puede encontrar en el laboratorio *Automatic Control Lab* diseñado por la Universidad de Siena, en donde el usuario dispone de una serie de experimentos orientados a conocer el funcionamiento de los diferentes procesos de control [CPV03].

Finalmente, los laboratorios de lectura y escritura son aquellos en los que se permite al usuario un control total sobre el funcionamiento del experimento. Es decir, el usuario puede cambiar desde la lógica del experimento hasta hacer modificaciones en la configuración de los instrumentos disponibles. Como ejemplo, el laboratorio VISIR del Blekinge Institute of Technology permite a los usuarios construir físicamente un circuito electrónico de manera remota, seleccionando los componentes oportunos, su conexión y posteriormente, dispone de la instrumentación necesaria para el análisis del comportamiento del circuito [GZH+07].

García-Zubía et al. [GLH+07b] realiza una clasificación de los laboratorios similar a la de Dieter Müller, de modo que asemeja las acciones de lectura remota de información a aquellas relativas a la observación de resultados a través de los instrumentos del laboratorio. Las acciones de escritura son aquellas que permiten modificar los parámetros del laboratorio y finalmente añade una acción más que es posible llevar a cabo en los laboratorios en los que el usuario tiene la oportunidad de alterar no solo los parámetros sino también la propia lógica del experimento. La Tabla 1.4 muestra algunos ejemplos de laboratorios relativos a esta clasificación.

Aquellos laboratorios denominados de instrumentación son en los que se dispone de una serie de experimentos diseñados de tal manera que el usuario únicamente puede excitar sus entradas (mediante interruptores virtuales, generadores de funciones, etc.) y observar sus salidas ya sean virtuales o reales (diodos LEDS, señales en un osciloscopio, etc.), haciendo uso de una webcam o bien de un interfaz que es capaz de interpretar y representar estas salidas en un entorno web.

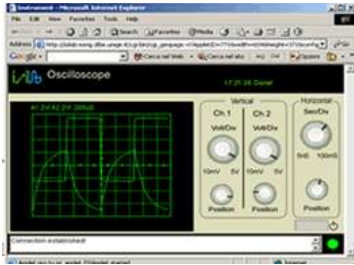


Tipo actividad remota posible	Ejemplo laboratorio	Web
<p>Control únicamente de la instrumentación.</p> <p>Acción de lectura de las entradas y salidas</p>		<p>ISILab, University of Geneve [BCS05a] http://isilab.dibe.unige.it/</p> <p>Permite únicamente modificar las entradas del circuito y visualizar los cambios producidos en el circuito</p>
<p>Control de los parámetros de funcionamiento del laboratorio.</p> <p>Acciones de lectura y escritura en el laboratorio</p>		<p>Automatic Control Lab, University of Siena [CPV03] http://www.dii.unisi.it/~control/act/home.php</p> <p>Permite modificar en cada experimento ciertos parámetros del control y observar los resultados</p>
<p>Control de la lógica interna del experimento</p>		<p>PLD WebLab-Deusto, Universidad de Deusto [GAH08] http://www.weblab.deusto.es</p> <p>Permite descargar un fichero sobre un dispositivo programable real y comprobar su funcionamiento modificando las entradas y visualizando las salidas producidas ante el programa descargado</p>

Tabla 1.4: Ejemplos de laboratorios en función de las actividades remotas posibles

En los laboratorios de instrumentación el usuario es prácticamente un mero espectador, en el que únicamente realiza un ejercicio de lectura sobre las salidas, ya que o bien solo tiene la oportunidad de modificar ciertos parámetros de los instrumentos que rodean el experimento y visualizar los resultados, o bien únicamente puede capturar la medida de ciertos parámetros físicos del entorno recogidos por sensores. Un ejemplo de este tipo de laboratorios se puede encontrar en el Institute of Automation de Maribor (Eslovenia) [BM06], en el cual se controlan mediante LabVIEW la temperatura y la humedad de una cámara de ensayos WEISS SB1 300, la cual se usa para medir la respuesta y funcionamiento de instrumentos cuando se les cambia las características del entorno.

Otro tipo de laboratorios son aquellos en los que se pueden modificar los parámetros de la lógica del sistema a controlar. El caso más claro es el de

un sistema de control PID donde el usuario no puede modificar la estructura del regulador pero sí sus parámetros de control. En este tipo de laboratorios el usuario se limita a modificar alguno de los parámetros que intervienen en el sistema y observar los cambios en la respuesta del mismo. Es decir, en este caso el usuario es capaz de escribir datos en el laboratorio y leer las respuestas. Una de las aplicaciones más extendidas en las que se hace uso de estos laboratorios son aquellas en las que se quiere controlar un sistema automático complejo a través de una aplicación SCADA (Supervisory Control And Data Acquisition) como el mostrado en [AA09].

Finalmente nos encontramos aquellos laboratorios en los que es posible modificar la lógica y los parámetros del sistema a controlar. Un ejemplo sencillo sería una maqueta didáctica clásica controlada por un microcontrolador que ha sido programado remotamente por el alumno. Lo mismo podría decirse para un procesador digital de señales o un sistema controlado por PC y programado con LabVIEW, etc. En este tipo de laboratorios los alumnos pueden realizar sus diseños en casa y una vez compilados pueden a través de una aplicación web descargar su código sobre el dispositivo remoto y testear el funcionamiento de su programa mediante la modificación de ciertas entradas (interruptores, por ejemplo), y visualizar la salida a través de una webcam u otros elementos software capaces de interpretar los cambios producidos en el experimento. Un ejemplo de este tipo de laboratorios es el diseñado por el grupo WebLab-Deusto en el que se disponen de varios laboratorios remotos para su uso docente en asignaturas relacionadas con la programación de microcontroladores [GAO⁺10].

C) Clasificación en función del tipo de experimentos

Desde el punto de vista de los experimentos éstos se pueden clasificar en experimentos por lotes, experimentos interactivos y experimentos basados en sensores [HAL⁺08].

Los experimentos por lotes son aquellos en los que los usuarios envían al laboratorio procesos que se ejecutan remotamente y donde el experimento devuelve los resultados sin que el usuario pudiera interactuar con él durante un tiempo determinado. Este modo de funcionamiento implica que no hay nunca dos usuarios utilizando el mismo experimento a la vez, sino que el primer usuario ejecuta todo su trabajo con un experimento,

el siguiente usuario con otro experimento y el siguiente con otro. Es decir, el sistema está preparado para gestionar una cola de usuarios que esperan su turno para poder acceder al laboratorio.

Un ejemplo de este tipo de experimentos lo encontramos en el laboratorio iLab desarrollado en el Massachusetts Institute of Technology [ABM⁺02] en el que los alumnos pueden caracterizar una serie de semiconductores mediante la preparación previa de un test de control que se realiza mediante el uso de un editor interactivo previo a que se ejecute la caracterización del componente

Los experimentos interactivos son aquellos en los que el usuario monitoriza y controla uno o más parámetros del experimento durante su ejecución en tiempo real. Un ejemplo de estos experimentos es el MIT's Teach-Spin Lab [ACH⁺03], en el que los alumnos pueden modificar dinámicamente el valor de la frecuencia y la amplitud de la corriente con la que se excita una bobina y así observar y medir el movimiento del imán que se encuentra suspendido dentro de la bobina.

Finalmente los experimentos basados en sensores son aquellos en los que el usuario monitoriza o analiza en tiempo real los datos obtenidos del experimento, pero sin interferir en el fenómeno que se esté produciendo. Un ejemplo son todas aquellas aplicaciones que proporcionan al usuario información de las condiciones meteorológicas de un determinado lugar a través de la información recogida por sensores monitorizados de manera remota [AS02].

Es importante definir el tipo de experimentos que se van a ofrecer a los usuarios pues en base ellos se diseñará el sistema de gestión de las conexiones o sistema de reservas en el laboratorio. De este modo:

- Los experimentos por lotes son aquellos en los que la configuración completa del experimento se realiza antes de que éste comience. Es decir, el usuario prepara en primer lugar una serie de comandos que después envía al experimento y espera a que éste le devuelva el resultado de dicha configuración.
- En los experimentos interactivos el usuario puede monitorizar y controlar uno o más aspectos del experimento durante la ejecución en tiempo real del mismo, cambiando así su comportamiento de manera dinámica.

- Los experimentos sensorizados o basados en sensores, son aquellos en los que el usuario monitoriza o analiza el comportamiento del experimento pero sin intervenir en él de ninguna manera, de modo que no puede ejercer ningún control sobre los parámetros del mismo.

1.1.2 Arquitectura de un laboratorio remoto

Un laboratorio remoto consiste en una plataforma tecnológica que puede permitir a un alumno remoto a través de su ordenador comunicarse, observar y/o controlar a distancia un experimento situado en las lejos de él, haciendo uso de Internet como medio de comunicación.

Para hacer posible la comunicación entre el usuario y el experimento, la arquitectura de un laboratorio remoto está basada en el paradigma cliente-servidor (Figura 1.2), en donde las tecnologías asociadas a Internet son el centro de los desarrollos actuales. Incluso utilizando diferentes alternativas tecnológicas, la mayoría de las soluciones están fundamentadas en esta simple arquitectura. En un extremo de la comunicación se sitúa el experimento junto con su equipamiento e instrumentos y un ordenador que permite por una parte el control del experimento y los instrumentos y por otra proporciona conexión a Internet, actuando como un servidor del laboratorio en un extremo de la comunicación. En el otro extremo, el usuario utilizará una aplicación cliente (en la mayoría de las situaciones, un navegador web o un plug-in desarrollado a tal efecto), que le ofrecerá la posibilidad de controlar y monitorizar remotamente el experimento.

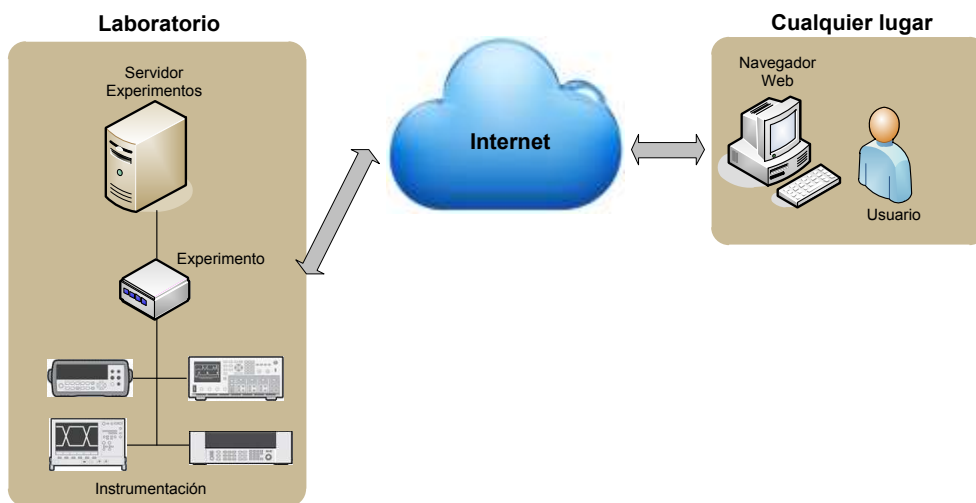


Figura 1.2: Arquitectura básica cliente-servidor de un laboratorio remoto

En algunos casos la aplicación cliente de un laboratorio remoto puede integrarse bien en los denominados Learning Management Systems (LMS) o bien en los más actuales entornos denominados Personal Learning Enviroments (PLE) [ED11], lo que facilita la implicación del laboratorio remoto en el proceso de aprendizaje al estar su acceso embebido en un espacio dedicado completamente al estudio de una materia concreta [GNR05].

Por lo tanto en la arquitectura de un laboratorio remoto denominaremos cliente al software a través del cual el usuario accede e interactúa con los experimentos disponibles. En función de la actividad a desarrollar, el cliente facilitará al estudiante las aplicaciones o servicios necesarios para, por ejemplo, enviar y/o leer un fichero al/desde el servidor, mostrar los resultados del experimento, permitir controlar y monitorizar los equipos del experimento o mostrar en tiempo real mediante un servicio de video y/o audio lo que sucede durante el ejercicio práctico [GOL+07].

Por consiguiente, el mejor cliente a proporcionar al usuario será aquel mediante el cual pueda realizar todas las actividades que llevaría a cabo en un laboratorio tradicional sobre el mismo experimento, sin ninguna restricción y a través de una aplicación estándar, como es un navegador de Internet. Para ello, las tecnologías de desarrollo deben permitir integrar en el navegador todas las aplicaciones y servicios necesarios para ello.

Análogamente, en la arquitectura de un laboratorio remoto denominaremos servidor al software que se encarga del control y gestión del experimento. El servidor está en comunicación directa tanto con el cliente remoto como con el experimento y los instrumentos que le rodean, por lo que es a través de él como accede el cliente al laboratorio. Así, y aunque el cliente es una parte importante del laboratorio remoto, es en el servidor donde más esfuerzos de diseño y desarrollo se concentran, siendo sus tareas principales:

- Gestión y administración de los usuarios del laboratorio. En un laboratorio remoto, al igual que en el tradicional, el profesor necesita poder realizar más acciones que los alumnos, como por ejemplo: puede crear sesiones de laboratorio con experimentos específicos, puede asignar materiales a cada sesión, asignar turnos a los alumnos, ver cuántas veces el alumno accede al laboratorio, etc.

- Gestión de las conexiones. El servidor será el encargado de determinar en primer lugar si el laboratorio puede ser usado por varios alumnos a la vez o se debe disponer de una cola de alumnos esperando a usar el laboratorio. Podrá disponer de un sistema de gestión de reservas mediante la cual se controlará la conexión al laboratorio de los alumnos durante un periodo de tiempo determinado.
- Autenticación, gestión de permisos, integridad y privacidad de las comunicaciones. El servidor deberá estar dotado de las herramientas y tecnologías necesarias para garantizar la seguridad en las comunicaciones, tanto de los datos intercambiados entre el cliente y el servidor, como proteger a este último de posibles ataques exteriores que pueden atentar contra los datos de los usuarios almacenados.
- Control y gestión de los experimentos disponibles en el laboratorio. El servidor es el encargado de gestionar la comunicación tanto con el experimento como con la instrumentación y demás componentes disponibles en el laboratorio, por lo que deberá implementar los protocolos de comunicación necesarios para el control de los mismos.

Siendo el cliente y el servidor los elementos principales de un laboratorio remoto, también son necesarios los sistemas de comunicaciones que intervienen en el proceso:

- Comunicación cliente-servidor: en la gran mayoría de los laboratorios remotos existentes y en todos los casos analizados en la sección 1.1.1, la comunicación entre el alumno y los elementos sujetos a experimentación del laboratorio se realiza utilizando Internet como plataforma para el intercambio de información entre ambos extremos de la comunicación. Existen hoy en día multitud de tecnologías para implementar dicho enlace de comunicaciones, pero debemos recordar que cualquier transacción de información realizada a través de la World Wide Web está basada en el protocolo HTTP, por lo que la tecnología utilizada para la implementación de la comunicación entre el cliente y el servidor deberá tener en consideración los requisitos establecidos para ellos por HTTP.
- Comunicación servidor-experimento: como se ha dicho anteriormente, entre las tareas del servidor se encuentra controlar los experimentos del laboratorio, los cuales pueden estar formados por el sistema bajo prueba así como la instrumentación necesaria para realizar los test y

las medidas correspondientes al ejercicio. Para ello, el servidor debe disponer de un conjunto de algoritmos de control así como de un sistema de comunicaciones que permita por una parte ejecutar dicho control sobre los experimentos y los equipos, y por otra recoger los datos generados durante las pruebas realizadas y que serán enviados al cliente del laboratorio remoto para ponerlos a disposición del usuario.

Vemos por tanto cómo, aún pareciendo sencillo el paradigma cliente-servidor, son muchos los sistemas necesarios para obtener un laboratorio remoto que satisfaga las necesidades y requisitos para ser una herramienta educativa de calidad e integrada en la docencia universitaria. De este modo su diseño deberá cumplir con una serie de expectativas tecnológicas, lo que hace que la arquitectura básica mostrada en la Figura 1.2 pueda verse modificada en función de la aplicación y funcionalidades de las que se quiera dotar al laboratorio [Alv11]. Además, al ser un laboratorio remoto un servicio tecnológico que debe estar integrado en las aplicaciones de la universidad, este requiere una serie de infraestructuras y facilidades que van ser proporcionadas por la propia institución. Deberá prestarse atención a aquellos requisitos relacionados con la visibilidad del laboratorio desde el exterior, ya que el alumno tiene que acceder de manera remota a recursos localizados físicamente en los laboratorios de la universidad, por lo que se le deberán aplicar las mismas políticas de gestión de acceso y seguridad empleadas en el resto de aplicaciones mantenidas y soportadas por los servicios informáticos de la universidad.

Adicionalmente, el laboratorio remoto a desplegar deberá cumplir una serie de características que implicarán que cualquier alumno pueda participar en el laboratorio sin restricciones respecto al tipo de aplicaciones y plataformas con la que esté habituado a trabajar. Es recomendable que el usuario disponga de herramientas o utilidades que hagan que sus acciones en el laboratorio sean sencillas y que le transmitan sensaciones similares a las que tendría en el laboratorio de la universidad, lo que le ayudará a mejorar su experiencia de aprendizaje. Para obtener una visión más amplia y precisa de estas necesidades y requisitos que deben cumplir los desarrollos llevados a cabo para construir un laboratorio remoto se pueden consultar los trabajos realizados por Garcia-Zubia et al. en [GLO+06a] y [GLH+07b] y especialmente en [GOL+09].

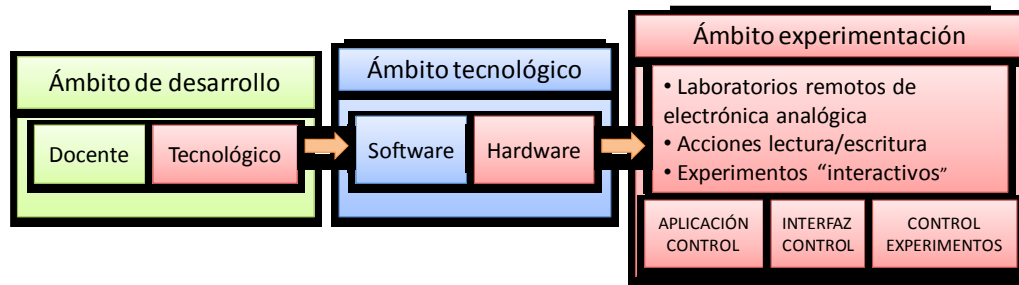


Figura 1.3: Contextualización del trabajo de investigación

Por consiguiente el análisis, diseño y desarrollo de un laboratorio remoto es una tarea complicada que debe hacerse de manera concisa y minuciosa, ya que el conjunto de tecnologías utilizadas para su construcción juega un papel decisivo en su viabilidad, su estabilidad en el tiempo y su aprovechamiento y rendimiento como herramienta educativa [HAE⁺07]. De este modo si bien el desarrollo de un laboratorio remoto no es el objeto de este trabajo de investigación, sí lo es el proporcionar una solución concreta a una serie de problemas que aparecen al desarrollar partes específicas de la arquitectura de un laboratorio remoto y que afectan a su posterior despliegue y uso. Por lo tanto, en la siguiente sección describiremos el contexto en el que se ha desarrollado nuestro trabajo de investigación.

1.2 Contextualización del trabajo de investigación

La contextualización del trabajo de tesis doctoral que presentamos aparece representada gráficamente en la Figura 1.3. En primer lugar analizaremos el punto de vista de desarrollo desde el cual se enfoca este trabajo. A continuación fijaremos la atención en el área tecnológica en la que nos hemos centrado, y ya en el marco concreto de la experimentación remota, especificaremos sobre qué tipo de laboratorio trabajaremos y cuáles son los sistemas propios de un laboratorio remoto en los cuales desplegaremos la metodología de control propuesta.

Las razones por las que se comienza a desarrollar un laboratorio remoto pueden ser muy diversas pero principalmente pueden ser agrupadas en dos: motivos docentes o motivos tecnológicos. Desde el punto de vista docente el laboratorio remoto es empleado por alumnos y profesores. En la gran mayoría de las situaciones el laboratorio es un complemento a las actividades que el alumno realiza en la universidad, pero también podrá ser utilizado por el profesor para alternar explicaciones teóricas con

demostraciones prácticas sin tener que desplazar a todos los alumnos a los laboratorios. Además la inclusión de un laboratorio remoto como herramienta docente proporciona las ventajas inherentes a su naturaleza: alta disponibilidad de horario y accesos, autogestión del tiempo, etc. [GCC+07]

No debemos olvidar tampoco que la formación práctica es sumamente importante en la educación de una carrera técnica [GRB10], por lo que los laboratorios remotos pueden jugar durante este proceso un papel fundamental [GB09]. Tomando como referencia la *Accreditation Board for Engineering and Technology* (ABET), responsable de los procesos de acreditación de los estudios de ingeniería en Estados Unidos, se puede aproximar la calidad docente de los ejercicios prácticos en el laboratorio según los objetivos marcados por esta institución [Fei02]. Por tanto, las herramientas docentes empleadas durante dichos ejercicios, ya sean manuales/locales o remotas, deberán tratar de ayudar a alumnos y profesores a desarrollar los objetivos y competencias dictadas según esta agencia de acreditación.

De este modo, si bien las motivaciones técnicas que conllevan el diseño y despliegue de un laboratorio remoto pueden tener o no su origen en satisfacer unas necesidades educativas, su meta debería ser siempre obtener una herramienta docente de calidad al servicio de profesores y alumnos. Para ello investigadores, diseñadores y administradores de los laboratorios remotos deberán trabajar con profesores que les ayuden a comprender el contexto y las necesidades derivadas del uso del laboratorio remoto como herramienta educativa.

Vemos por tanto como ambas cuestiones están directamente relacionadas pues para alcanzar ciertos objetivos educativos durante la realización de una actividad práctica remota será necesario emplear una herramienta tecnológicamente óptima. Es decir, la selección de las tecnologías empleadas, así como de las estrategias utilizadas en el diseño y desarrollo del laboratorio remoto determinaran el uso que se pueda hacer posteriormente del mismo como herramienta docente.

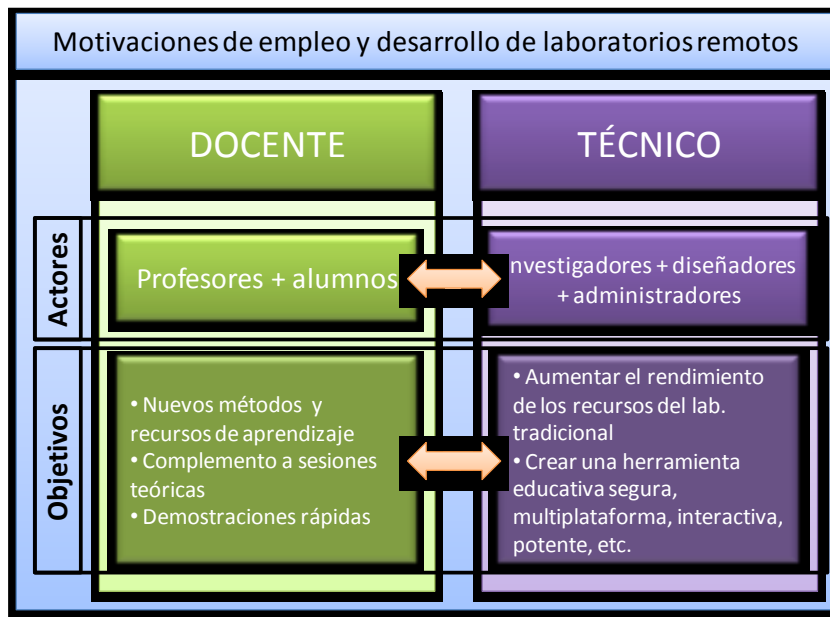


Figura 1.4: Motivaciones de empleo y desarrollo de laboratorio remoto

Gráficamente las perspectivas de uso y desarrollo de un laboratorio remoto, así como los actores que participan en cada una de ellas y los objetivos que cada colectivo persigue aparecen reflejados en la Figura 1.4, en donde se observan también las relaciones existentes ambas visiones.

Así pues, en este trabajo de tesis nos vamos a centrar exclusivamente en ciertos aspectos técnicos relativos a la implementación de un laboratorio remoto empleado durante el aprendizaje de electrónica analógica, pero sin olvidar que las decisiones técnicas tomadas repercutirán en la calidad docente del laboratorio remoto en que se integren. En la siguiente sección analizamos el área tecnológica específica que abarcaremos con este trabajo.

1.2.1 Contexto tecnológico del trabajo de investigación

Cuando se piensa en un desarrollo tecnológico en el ámbito de la ingeniería, y más si se trata del área de la electrónica, la informática o las comunicaciones, siempre aparecen dos conceptos o cuestiones principales: hardware y software, es decir, una aplicación que gestiona, controla o monitoriza un dispositivo físico. En el caso de un laboratorio remoto también se puede aplicar esta dicotomía, aunque con ciertas apreciaciones pues la barrera entre el hardware y software puede ser difusa. Para aclarar esta idea y así seguir contextualizando nuestro trabajo fijémonos en la Figura 1.5.

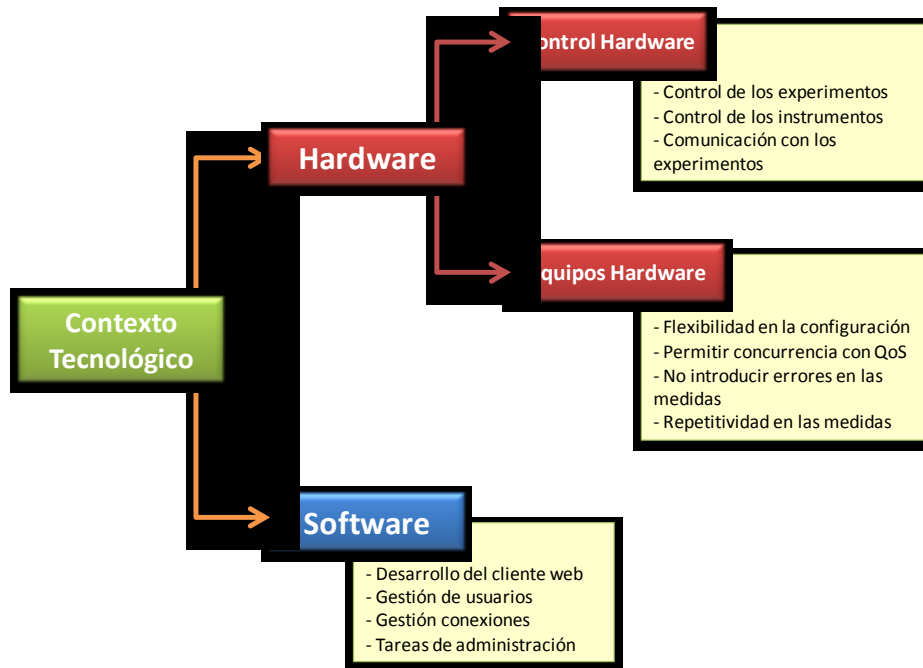


Figura 1.5: Contexto tecnológico en un laboratorio remoto.

Una primera aproximación podría ser la diferenciación entre software y hardware, pero durante este trabajo esta división será realizada con matices. Así, por desarrollo software en un laboratorio remoto se entiende aquellas aplicaciones destinadas al desarrollo del cliente, del servidor y de aquellas funciones de gestión que en estas entidades se desarrollan. En cambio bajo el epígrafe hardware englobaremos, junto con los equipos, instrumentos y experimentos, también aquellas aplicaciones software que permiten controlar los dispositivos físicos del laboratorio. De este modo, este trabajo de investigación está orientado a proporcionar soluciones en estos dos últimos epígrafes: el control hardware de los equipos y componentes del laboratorio remoto.

Obviamente, los objetivos y problemas a solucionar en cada caso son diferentes y serán analizados posteriormente en la sección 1.3, pero a modo de introducción podemos decir que el control hardware debe proporcionar herramientas para el control y comunicación con los instrumentos y experimentos que permitan una rápida y sencilla reconfiguración. Por su parte, los equipos seleccionados deberán permitir al alumno realizar las mismas acciones que si se encontrara en un laboratorio manual, proporcionando para ello respuestas y medidas veraces, sin que su localización remota introduzca ninguna anomalía o error en las medidas,

más allá de los que se pueden producir en una sesión normal en el laboratorio manual.

Para finalizar la contextualización de nuestro trabajo, y determinar así los sistemas o bloques de un laboratorio remoto en los cuales vamos a concentrarnos, vamos a partir de la arquitectura definida en la sección anterior. Tomando como referencia la Figura 1.2, si se representan ahora gráficamente cada uno de los sistemas que forman parte de la arquitectura de un laboratorio remoto e indicamos en cada uno de ellos cuáles son sus funciones, se obtiene la Figura 1.6.

En primer lugar advertimos en esta representación gráfica que tanto el cliente como el sistema de comunicaciones entre el cliente y el servidor quedan totalmente fuera del alcance de nuestro trabajo. Del mismo modo sucede con todos los sistemas necesarios para realizar las funciones de gestión de usuarios, conexiones o permisos que tienen lugar en el servidor, es decir, lo que en la Figura 1.5 englobábamos como software del laboratorio remoto.

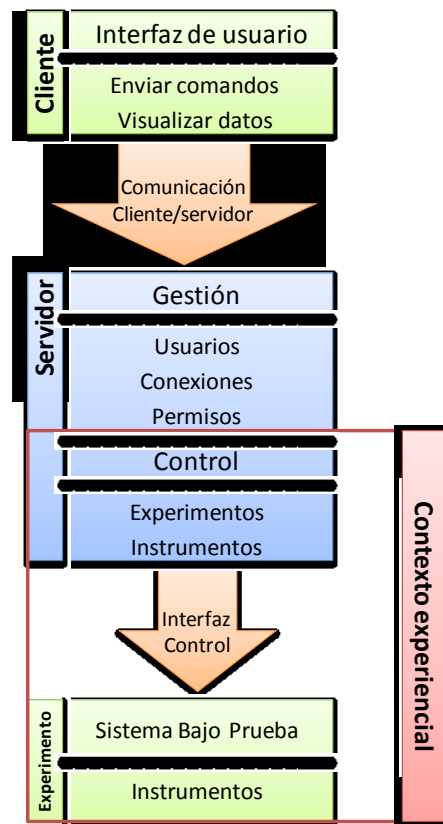


Figura 1.6: Contexto experiencial del trabajo de investigación

De este modo el contexto en el cual se ha desarrollado este trabajo estará enfocado a solucionar problemas relacionados con el control del hardware de un laboratorio remoto, los cuales se encuentran agrupados en los siguientes sistemas:

- Control de experimentos e instrumentos: comprende los algoritmos de control encargados de ejecutar sobre el sistema bajo prueba y los instrumentos los comandos enviados por el alumno, y posteriormente de devolverle la respuesta generada por cada uno de los elementos que integran el experimento.
- Interfaz de comunicaciones y control: es el sistema tanto hardware como software necesario para comunicar el servidor y el experimento con el objetivo de ejecutar sobre él los comandos enviados por el usuario y devolverle las medidas realizadas mediante los instrumentos, para que posteriormente sean mostradas en el cliente web.
- Experimento: es el equipamiento real al que el usuario accede remotamente. Está formado por el sistema bajo prueba y los instrumentos necesarios para llevar a cabo las medidas contempladas en el ejercicio al que está destinada la sesión práctica en la que se emplea el laboratorio remoto. Este trabajo está enmarcado en el área de los laboratorios remotos de electrónica, por lo que el sistema bajo prueba será un circuito electrónico.

El experimento deberá contar además con los mecanismos necesarios para poder seleccionar o construir diferentes circuitos electrónicos así como poder alimentarlos con diferentes señales y tomar medidas de su comportamiento en aquellos puntos que se consideren necesarios para comprender el funcionamiento del circuito.

Todas estas acciones se deberán realizar sin introducir ninguna modificación sobre el circuito ni ningún error en la captura de datos, aportando de este modo fiabilidad, repetitividad y calidad de servicio a las medidas y acciones realizadas por el usuario de manera remota.

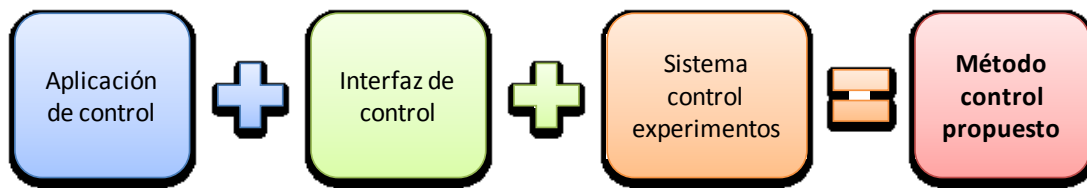


Figura 1.7: Sistemas que forman el método de control propuesto

Así, al conjunto de soluciones que describiremos posteriormente y que comprenden este trabajo de investigación, lo denominaremos de aquí en adelante método de control propuesto, el cual está formado por los sistemas mostrados en la Figura 1.7: una aplicación software para la configuración, control y monitorización de los experimentos de un laboratorio remoto de electrónica analógica, un interfaz de comunicaciones y control y un sistema hardware de control de experimentos diseñado especialmente para este método de control. La combinación de estos tres sistemas proporciona una alternativa tecnológica única y novedosa, que presentamos como solución a los problemas que aparecen durante el desarrollo y despliegue de estos sistemas como parte de la arquitectura de un laboratorio remoto.

Por otra parte, y en base a la clasificación sobre los tipos de laboratorios remotos realizada en la sección 1.1.1, ya hemos adelantado anteriormente que el contexto de aplicación de la solución desarrollada es el de los laboratorios remotos de electrónica. Es decir, no se trata de una solución para ser desplegada en laboratorios virtuales o híbridos, sino que el enfoque seguido durante el desarrollo del método propuesto plantea su uso con instrumentos y circuitos electrónicos puramente reales controlados remotamente por el usuario a través del cliente web.

Gracias a las soluciones propuestas, el usuario podrá realizar sobre los experimentos actividades de lectura y escritura así como de control de los instrumentos y de la lógica y parámetros del experimento prácticamente igual que si se encontrara en un laboratorio tradicional [CDF02]. Estas acciones serán posibles ya que mediante la configuración y desarrollos que posteriormente analizaremos, el usuario podrá construir partiendo de los componentes discretos un circuito electrónico, alimentarlo con distintos tipos de señales, y comprobar su funcionamiento en diferentes situaciones.

Para ello el alumno podrá configurar los instrumentos o el circuito bajo prueba añadiendo, eliminando o sustituyendo componentes del mismo

en función de las especificaciones de la práctica que se esté realizando en cada momento. Además, el usuario podrá también tomar medidas en diferentes puntos del circuito para posteriores análisis.

Así, el proceso de experimentación que el usuario llevara a cabo será el mostrado en la Figura 1.8: construcción del circuito, conexión y configuración de instrumentos, y observación del funcionamiento, pudiendo repetir estos pasos en el caso de realizar modificaciones sobre el circuito creado en el paso primero.

Respecto al tipo de experimentos, si bien el método de control está pensado para ser desplegado en laboratorios que contengan experimentos por lotes, el algoritmo de control implementado permite que el usuario tenga la perspectiva de estar realizando experimentos interactivos, ya que puede interactuar sobre el experimento de manera dinámica, obteniendo la respuesta a sus acciones sobre el mismo en tiempo real.

En resumen, el trabajo que aquí presentamos trata de resolver los problemas que aparecen a la hora de realizar el control de los experimentos proporcionados por un laboratorio remoto de electrónica mediante el cual se quiere proporcionar al alumno la capacidad de realizar las mismas acciones que en un laboratorio manual.

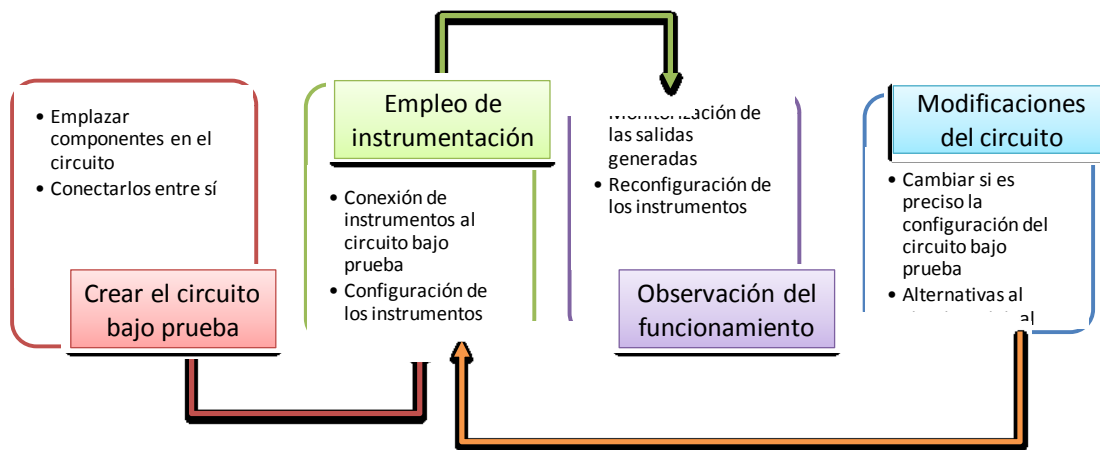


Figura 1.8: Pasos en el proceso de experimentación

En la siguiente sección se procede a describir dichos problemas, los cuales han sido hallados como fruto de los dos estudios que se encuentran incluidos en el capítulo segundo de este trabajo:

- Análisis de las tecnologías e interfaces que pueden ser utilizadas para el control de la instrumentación usada normalmente en laboratorios de electrónica.
- Estudio de las soluciones empleadas para el control de experimentos en otros laboratorios remotos de electrónica y que son referentes a nivel mundial en el campo de la educación a distancia de electrónica. Este estudio nos permite valorar diferentes alternativas, comparar las ventajas e inconvenientes de unas y otras, y razonar y justificar la solución propuesta.

1.3 Descripción de la problemática

Independientemente de la perspectiva utilizada para desarrollar un laboratorio remoto, el objetivo último debe ser proporcionar una herramienta de aprendizaje fácil de usar, de mantener y actualizar con nuevos equipos y experimentos, de modo que las medidas y resultados ofrecidos durante la experimentación sean reales y veraces.

Por tanto, si el resultado es un recurso educativo ofrecido por la universidad, este tendrá que satisfacer una serie de requisitos tanto desde el punto de vista pedagógico [MV07], como desde el punto de vista técnico en lo que se refiere a cómo acceden los alumnos al laboratorio, cómo interaccionan con los experimentos, el grado de usabilidad del sitio web creado, etc. [GOH+09] [OAD+07] [CCA02], aspectos que difieren en muchos casos de los requeridos para realizar el mismo ejercicio práctico en el laboratorio tradicional, tanto de recursos hardware para construir el experimento, como software para implementar el interfaz de acceso necesario [GFB+08].

Cada laboratorio debido a su tipología, a la asignatura que se encuentra vinculado, y en general a sus necesidades concretas de implementación, presenta una serie de características propias que hace que generalmente solamente pueda ser desplegado allí donde ha sido desarrollado y en el entorno para el cual ha sido creado. Mientras que en alguno de los laboratorios los experimentos pueden compartir ciertas

infraestructuras, en otros cada experimento requiere una arquitectura hardware y software específica. Estos aspectos contribuyen a que sea complicado compartir entre instituciones o incluso entre laboratorios de la misma área el uso de las plataformas educativas que posibilitan el acceso a los laboratorios remotos [LLI07].

Puede incluso suceder que un mismo laboratorio sea difícilmente modificable o ampliable, tanto desde el punto de vista de los instrumentos como de los experimentos. Esto puede ser debido a que el software desarrollado haya sido diseñado para ser operativo únicamente con unos instrumentos específicos, ya sea porque dependen del fabricante, del tipo de instrumento o del interfaz de control empleado. También puede suceder que dicho software únicamente permita trabajar con unos circuitos concretos y que la sustitución de unos por otros sea difícil y requiera la recodificación de la aplicación.

Por tanto el objetivo de este trabajo es obtener un método de control que siendo independiente de los instrumentos y los circuitos incluidos en los experimentos, permita modificar el hardware del laboratorio de manera sencilla sin la necesidad de alterar su software de control. Gráficamente estas ideas aparecen representadas en la Figura 1.9, en donde se ilustra cómo el eje principal de este trabajo es obtener un método de control que permita independizar los experimentos (instrumentos y circuitos bajo prueba) de su aplicación de control y gestión.

La necesidad de plantear y desarrollar un nuevo método de control surge después del análisis inicial del estado del arte que detallamos en el capítulo siguiente en el cual se han estudiado diferentes metodologías de control desplegadas en laboratorios remotos desarrollados en el ámbito de la electrónica analógica [LMV09] [BPP+07] [Gus06] [ACH+03] [MN02]. En este estudio nos hemos concentrando exclusivamente en los tres niveles contemplados en el alcance y contexto de este trabajo, por lo que se describirán a continuación los problemas detectados y que se desean solventar en cada uno de ellos.

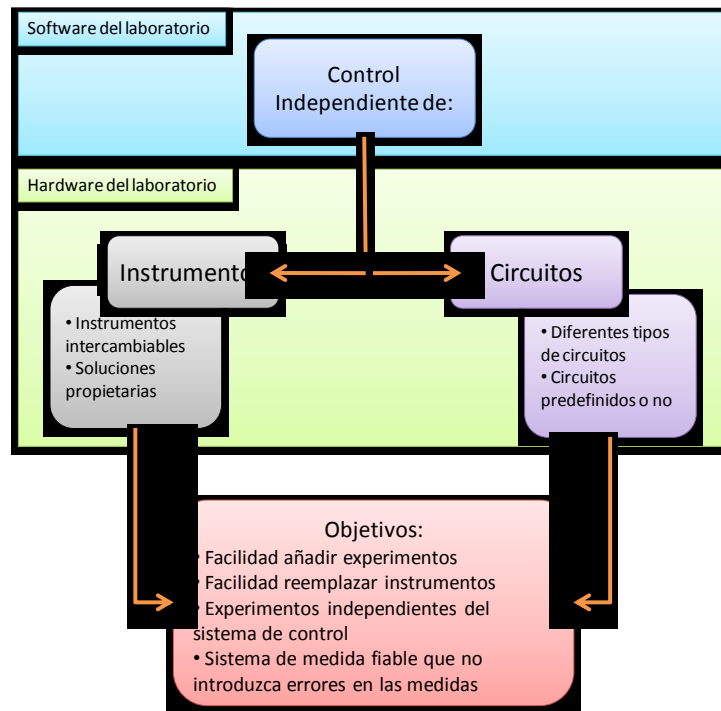


Figura 1.9: Esquema de objetivos del trabajo de investigación realizado

A) Problemática en la aplicación software de control

El principal problema a la hora de desarrollar el conjunto de algoritmos que controla los instrumentos y experimentos del laboratorio y que forma lo que se denominará de aquí en adelante aplicación o software de control es hacer posible que esta aplicación pueda ser empleada para controlar instrumentos de distinto tipo y proveedor, así como los circuitos bajo prueba que forman parte de los experimentos. Tenemos que tener en cuenta, que este software se encargará de gestionar los comandos enviados a los instrumentos y experimentos, así como de adecuar los datos obtenidos como resultado de la experimentación al formato que sea entendible posteriormente por el cliente, para así mostrar esta información al usuario del laboratorio remoto. Por lo tanto:

- Con diferentes tipos de instrumentos hacemos referencia a que el software de control desarrollado no dependa del fabricante del equipo sino que pueda ser utilizado con dispositivos comerciales de diferentes proveedores, siendo el rendimiento y comportamiento del experimento el mismo. Esta característica permite que en el laboratorio se puedan actualizar los instrumentos de los experimentos

de una manera rápida y sencilla, sin requerir la modificación del software de control. Esta característica permite también que el mismo software de control pueda ser utilizado por diferentes instituciones, utilizando cada una de ellas los instrumentos que mejor se adapten a sus necesidades, haciendo posible que la plataforma desarrollada pueda ser extensible y desplegable en más universidades, aparte de donde ha sido creada, pudiendo llegar a formar federaciones de laboratorios remotos.

- Con diferentes circuitos electrónicos hacemos referencia a que el mismo sistema de control permita trabajar con experimentos de diferente dificultad, adecuándose al aprendizaje del alumno y acompañándole durante su proceso educativo el máximo tiempo posible, sin tener que cambiar de herramienta, mejorando así el impacto en el proceso de aprendizaje [SY10] [GRB10] [CNE+04]. Es decir, en el caso de la electrónica, que el alumno pueda aprender desde cómo utilizar un osciloscopio a realizar montajes más complicados como puede ser un rectificador de media onda o un amplificador basado en transistores, pasando por la realización de montajes con resistencias. Para ello, el método de control empleado debe ser independiente del sistema bajo prueba, es decir, del tipo de circuito que el alumno esté testeando a través del laboratorio remoto.

B) *Problemática en el interfaz de control*

El principal problema consiste en proporcionar un interfaz de comunicaciones que sea fácilmente desplegable, extensible y con una calidad de servicio aceptable y adecuada a su uso en un laboratorio remoto:

- Fácilmente desplegable tanto en lo referente a la configuración del sistema de control como a las restricciones físicas impuestas por el mismo (número de equipos posibles, distancia entre los mismos o hardware especial). Esta característica permitirá que futuras actualizaciones de la instrumentación del experimento se puedan realizar de manera sencilla sin alterar el software de control diseñado previamente, por lo que para ello, el interfaz de comunicaciones también deberá ser transparente respecto de dicho software. Además,

reduciendo la necesidad de utilizar hardware especial, se puede reducir el precio de despliegue de los experimentos del laboratorio.

- Extensible respecto a la posibilidad de añadir nuevos instrumentos o reemplazar los instrumentos existentes por otros más actuales y que se adapten a las necesidades de las asignaturas en las que se emplee el laboratorio remoto.
- El interfaz de comunicaciones empleado, en combinación con el software de control desarrollado deberá proporcionar una calidad de servicio aceptable en lo que se refiere a velocidad y precisión en la respuesta, así como robustez frente a caídas o fallos en el sistema. Recordemos que el contexto de aplicación del método de control propuesto es su uso con experimentos por lotes, pero proporcionando al alumno la percepción de estar realizando un experimento interactivo, por lo que la velocidad de respuesta del sistema debe ser alta y la vulnerabilidad del sistema frente a fallos (caídas del servidor, fallos en las respuestas de los instrumentos) baja.

C) *Problemática en el control y configuración del experimento*

El problema principal aparece al querer proporcionar un sistema de control y configuración del experimento que permita la construcción física del circuito bajo prueba y su posterior testeo y validación por parte del alumno de manera remota, ejecutando para ello las mismas acciones que si se encontrara presencialmente junto al circuito. De este modo:

- Uno de los problemas que surgen a la hora de desarrollar laboratorios remotos y en concreto sus sistemas de control, es que se adoptan medidas propietarias que pueden ser difíciles de configurar e incluso de mantener. Por tanto, el problema a solucionar en esta área consiste en diseñar y desarrollar un sistema de control y configuración del experimento que haga uso en la medida de lo posible, únicamente de instrumentos y equipos comerciales y que puedan ser controlados de igual manera que el resto de instrumentos de medida del laboratorio. Hay que tener en cuenta que el sistema de control y configuración del experimento hace referencia al sistema que permite crear físicamente circuitos electrónicos de diferente complejidad y posteriormente llevar a cabo las pruebas y medidas necesarias para comprobar su funcionamiento.

- Por tanto para el testeo y validación la solución debe proporcionar mecanismos para poder interconectar los instrumentos al circuito, tanto para alimentarlo como para obtener los valores de las medidas en aquellos puntos que el alumno o instructor del laboratorio consideren oportuno. Estos valores serán capturados por los instrumentos, enviados al software de control a través del interfaz de control y posteriormente mostrados al cliente web como respuesta a las órdenes enviadas previamente por el alumno.
- El sistema implementado para la creación del circuito bajo prueba, así como para la conexión en el mismo de los instrumentos, no debe introducir errores significativos en las medidas ni interferir en el funcionamiento normal del circuito. Es decir, el hecho de realizar una experimentación remota no debe implicar una merma en la calidad del sistema de medida, por lo que éste deberá proporcionar repetitividad y fiabilidad en los resultados y por tanto errores controlados similares a los obtenidos en las prácticas presenciales y que pueden ser debidos a los propios componentes e instrumentos empleados.

En definitiva, el análisis de estos problemas es lo que ha motivado el desarrollo de un nuevo método de control que permita ser utilizado únicamente con instrumentos comerciales independientemente del fabricante de manera que sean intercambiables de una manera sencilla y en el que sea posible realizar experimentos de diferentes complejidad utilizando el mismo método de control para todos ellos.

Las metodologías existentes hoy en día se centran en dar soporte a un tipo concreto de experimentos, y en pocas ocasiones contemplan la utilización de instrumentos diferentes a los que usan actualmente, por lo que disponer de diferentes equipos a los planteados inicialmente requeriría rehacer una parte importante de la plataforma o bien su reestructuración completa. Este tipo de implementaciones son las que nos llevan a plantearnos las hipótesis y objetivos recogidos en la siguiente sección.

1.4 Hipótesis y objetivos

Tomando como referencia el contexto y alcance de este trabajo de experimentación, así como los problemas detallados anteriormente y que se

desea solventar con el método de control que aquí se presenta, la hipótesis de este trabajo de investigación se enuncia de la siguiente manera:

Es posible definir una metodología de control para ser desplegada en laboratorios remotos de electrónica analógica, que con independencia de los instrumentos y experimentos disponibles proporcione herramientas para la configuración de circuitos y la adquisición de medidas, obteniendo resultados lo más cercanos posibles a los adquiridos en un laboratorio manual.

Con el objetivo de validar la hipótesis expuesta, desarrollaremos un método de control para ser desplegado en un laboratorio remoto de electrónica que permita corroborar lo anteriormente expuesto, por lo que el objetivo general de este trabajo de tesis será:

Desarrollar un método para el control de experimentos en laboratorios remotos orientados a la electrónica analógica que permita al usuario practicar con circuitos y experimentos de manera eficiente tal y como si se encontrara en un laboratorio manual.

De este objetivo general derivan los siguientes objetivos específicos, los cuales están en línea con las acciones necesarias que llevaremos a cabo para la confirmación de la hipótesis enunciada primeramente:

- OE1. Diseñar una aplicación software para el control de experimentos adaptable a instrumentos de cualquier fabricante y circuitos electrónicos de diferente dificultad.*
- OE2. Proporcionar un interfaz de control entre el servidor y el experimento de modo que la respuesta del sistema sea de calidad y permita una fácil reconfiguración del mismo.*
- OE3. Proporcionar un sistema para el control de los experimentos basado en soluciones comerciales y que facilite la construcción y la validación de los circuitos bajo prueba mediante un sistema de medida fiable.*
- OE4. Validar el sistema de configuración y medida de circuitos propuesto mediante su despliegue en un laboratorio remoto funcional para su testeo con usuarios reales.*

Cada uno de los objetivos específicos trata de resolver las problemáticas concretas identificadas en niveles determinados de la arquitectura de un laboratorio remoto orientado a su uso en electrónica analógica. El primer objetivo hace referencia a la necesidad de disponer una lógica de control sobre el experimento que no esté basada ni en instrumentos ni en circuitos bajo prueba únicos o concretos.

El siguiente objetivo busca proporcionar al método de control un interfaz que comunique el servidor con el experimento, que sea fácilmente desplegable, sin necesidades restrictivas en lo referente a número de instrumentos o equipamiento específico y costoso, pero que proporcione una calidad de servicio aceptable y no interfiera de manera negativa en la experiencia del alumno en el laboratorio.

El tercer objetivo específico contempla la provisión mediante el método propuesto de un sistema de control del circuito bajo prueba y de los instrumentos que permitan al alumno construir circuitos, modificarlos y validar su funcionamiento mediante los instrumentos dispuestos a tal efecto, como si se encontrara en un laboratorio manual.

Finalmente, el último objetivo específico enunciado permitirá validar el método de control propuesto mediante su inserción en un laboratorio remoto completamente funcional y su uso con usuarios reales. Para ello, retomando el concepto de arquitectura de un laboratorio remoto y los sistemas excluidos del alcance de este trabajo (Figura 1.2 y Figura 1.6), añadiremos al método de control desarrollado un cliente para que los usuarios accedan al laboratorio y proporcionaremos al servidor del laboratorio el resto de funcionalidades necesarias para el correcto funcionamiento del laboratorio. Así podremos construir de manera remota varios circuitos sobre los que realizar medidas y comprobar mediante la comparación de estos resultados con los obtenidos en otros escenarios previamente validados, que estas medidas son correctas y que por tanto el sistema de control construido es fiable y no introduce errores significativos en las medidas.

Lograremos así satisfacer los objetivos específicos, y por tanto el objetivo general y así poder confirmar la hipótesis enunciada, mediante la consecución de los siguientes objetivos operativos:

- OO1. *Determinar los requisitos propios de los instrumentos de un laboratorio para diseñar un software de control adaptable a cada instrumento.*
- OO2. *Definir diferentes tipos de circuitos electrónicos para comprobar la validez del método de control.*
- OO3. *Establecer un interfaz de control entre el servidor y los experimentos.*
- OO4. *Construir y desplegar un sistema de control de experimentos para la construcción y validación de circuitos electrónicos*
- OO5. *Proporcionar mecanismos que hagan posible añadir nuevos experimentos y componentes de manera sencilla y rápida*
- OO6. *Diseñar y ejecutar experimentos con usuarios reales para la validación del método propuesto.*
- OO7. *Realizar pruebas de rendimiento y de toma de medidas sobre circuitos reales para comprobar su validez respecto a las adquiridas en un laboratorio remoto de referencia y en un laboratorio manual.*

Estos objetivos operacionales recogen las acciones más representativas que llevaremos a cabo durante el proceso de investigación y que concluirán con la obtención de un método de control único que satisfaga los objetivos e hipótesis planteadas al principio de este trabajo.

1.5 Metodología de investigación

El método de investigación llevado a cabo para realizar este trabajo de tesis ha sido desarrollado en función de la disponibilidad de infraestructuras y equipos requeridos, la accesibilidad a la información y a los expertos en la materia, así como la participación en publicaciones y proyectos de investigación desarrollados relacionados con el trabajo aquí presentado. En este contexto, para llegar a la consecución de los objetivos se diseñó previamente un plan de trabajo basado en las siguientes actividades:

1. Participación en la línea de investigación WebLab-Deusto¹ de la Universidad de Deusto.
2. Recopilación, estudio y análisis de las publicaciones recientes de los trabajos relevantes en el ámbito de la experimentación remota.
3. Participación en seminarios, talleres, conferencias y publicaciones relativas al trabajo del grupo de investigación WebLab-Deusto.
4. Construcción de una red de contactos de expertos mediante la participación en congresos, talleres, correspondencia electrónica y visitas a otros centros de investigación².
5. Estudio y despliegue de la plataforma VISIR en la Universidad de Deusto.
6. Diseño y desarrollo de cada una de las partes del modelo y metodología propuesta como alternativa tecnológica a las soluciones existentes.
7. Rediseño de la solución con la información obtenida como resultado de las actividades realizadas en los puntos anteriores.
8. Diseño y desarrollo del prototipo final para las pruebas y validación de la metodología propuesta.

La Figura 1.10 representa el proceso de investigación, indicando las fases seguidas así como los resultados obtenidos en cada una de ellas. Básicamente estas fases pueden enmarcarse dentro de una metodología de investigación del tipo investigación-acción, la cual consta de las siguientes fases:

- a) Identificación del problema: durante esta fase se determina la problemática y las cuestiones relevantes a solucionar, relacionadas con la problemática a tratar. Se identifican las alternativas posibles, así como las carencias de las soluciones existentes y las posibles mejoras a realizar.
- b) Plan de acción: una vez identificado el problema a solucionar, se identifican las posibles alternativas a seguir.

¹ Desde el año 2006, el autor pertenece al grupo de investigación WebLab-Deusto. Más información del grupo en www.weblab-deusto.es

² El autor ha sido investigador visitante en el Blekinge Tekniska Högskola (Ronneby, Suecia), como parte de su aprendizaje para el posterior despliegue de la plataforma VISIR en la Universidad de Deusto.

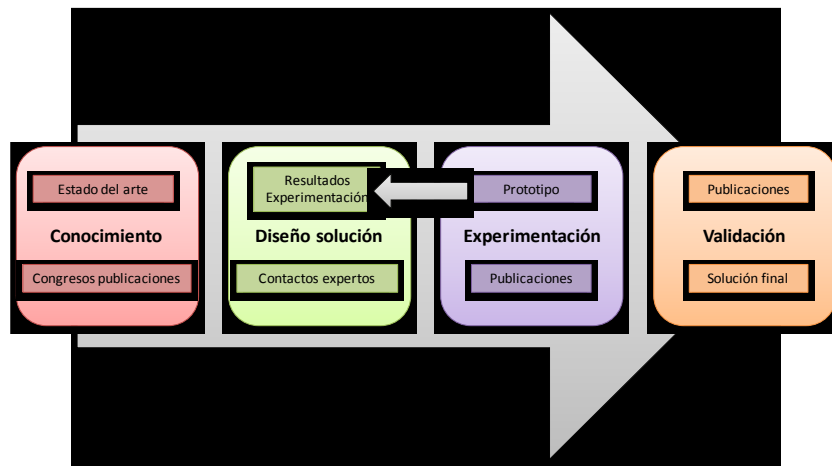


Figura 1.10: Esquema de la metodología de investigación seguida

- c) Desarrollo de acciones: se selecciona una de las alternativas identificadas en la fase anterior y se lleva a cabo de manera controlada y planificada.
- d) Evaluación y observación de los resultados: durante esta fase se observan los resultados obtenidos de las acciones llevadas a cabo y se analizan de manera crítica con el objetivo de evaluarlas y determinar su adecuación a la planificación realizada. Como resultado se pueden tratar nuevas cuestiones que pueden ser relevantes y que ayuden a mejorar el planteamiento inicial.

Este método de investigación puede ser repetido de manera cíclica con el objetivo de proporcionar una serie de reflexiones críticas y rigurosas que permitan obtener un proceso de mejora continuo en el que aparezcan nuevos retos e hitos a superar, obteniendo en cada iteración una solución más precisa que la anterior. Es por tanto un proceso de aprendizaje y reflexión cíclico.

1.6 Organización de la memoria

La memoria que presenta el trabajo doctoral realizado puede ser dividida en tres grandes apartados, los cuales representan los estadios o fases a través de los que se ha llevado a cabo el proceso de investigación: estudio de las tecnologías y soluciones ya adoptadas y relacionadas con los objetivos de nuestro trabajo, desarrollo de la solución y obtención de resultados en base a una experimentación práctica para la validación de la misma. Estas fases se han estructurado a su vez en los capítulos que se presentan a continuación.

El capítulo 2 recoge el análisis de las diferentes tecnologías, tanto hardware como software disponibles para desarrollar las soluciones a los problemas planteados en el primer capítulo en cada uno de los sistemas contemplados en el alcance del trabajo. El resultado obtenido es un estudio comparativo de las mismas, el cual ha sido empleado posteriormente para seleccionar aquellas que mejor se adaptan a nuestros requisitos y especificaciones.

Este capítulo segundo incluye también el estudio de aquellas soluciones anteriores ya desarrolladas e implementadas en laboratorios remotos que son referencia a nivel mundial en el área de investigación. Esto nos ha permitido analizar y comparar estas soluciones, y determinar sus carencias y aportaciones, para a continuación justificar la necesidad de aportar una nueva metodología para el control de los equipos y experimentos de un laboratorio remoto.

El capítulo 3 está dedicado íntegramente a la descripción de las soluciones planteadas a los problemas descritos previamente en el capítulo primero. A continuación, en una segunda parte de este capítulo procedemos a describir en profundidad los detalles técnicos de cada una de estas soluciones, los cuales son completados con la información contenida en los apéndices anexados al final de este documento.

Las pruebas realizadas durante la evaluación de los desarrollos llevados a cabo para obtener el método de control propuesto aparecen recogidas en el capítulo 4. De este modo los resultados, datos e información recogidos durante esta fase de la investigación han servido para demostrar la validez del conjunto de soluciones aportadas.

La memoria se completa en el capítulo 5, con las conclusiones a las que llegamos una vez analizado y validado el trabajo realizado. Para ello, la hipótesis inicial y los objetivos son revisados y se analizan las contribuciones científicas de la solución propuesta junto con las publicaciones realizadas a lo largo del proceso de investigación.

*Capítulo***2****Recursos tecnológicos y
antecedentes**

Un laboratorio remoto puede ser desarrollado mediante una gran variedad de tecnologías y metodologías, tanto desde el punto de vista software y hardware, como de las comunicaciones necesarias para conectar al usuario con el experimento. Pero aunque intervienen numerosos actores en la construcción de un laboratorio remoto debido a su implicación directa en el proceso educativo e incluso en el modelo organizativo y de gestión de los recursos de la universidad (infraestructuras, redes de comunicaciones, equipos, etc.), el conjunto de tecnologías utilizadas para la implementación de un laboratorio remoto juega un papel decisivo en su viabilidad, su estabilidad en el tiempo y su aprovechamiento como herramienta educativa [HAE⁺07], y en definitiva en el método de desarrollo y despliegue del laboratorio remoto resultante.

Este capítulo está dividido en dos partes bien diferenciadas pero estrechamente relacionadas entre sí. En primer lugar, analizaremos las diferentes tecnologías que pueden ser empleadas para el desarrollo y despliegue de cada uno de los tres niveles contemplados en el contexto y alcance de nuestro trabajo de investigación. Para ello analizaremos las distintas tecnologías de desarrollo software utilizadas para codificar los algoritmos que controlaran los experimentos mediante el servidor del laboratorio. A continuación, presentaremos los interfaces de control

comúnmente más utilizados y que hacen posible que los comandos recibidos y gestionados por el software de control sean ejecutados sobre los instrumentos del laboratorio. Finalmente, estudiaremos las alternativas disponibles para controlar el experimento, es decir, aquellas que hacen posible que el circuito electrónico diseñado por el alumno a través del interfaz web del cliente sea construido de manera real y medido y validado mediante los instrumentos disponibles en el experimento.

En la segunda parte de este capítulo analizaremos las diferentes combinaciones de las tecnologías descritas anteriormente y que dan como resultado los distintos métodos de control empleados en laboratorios remotos actuales y que nos han servido de referencia durante la realización de este trabajo.

Los plataformas que se van a analizar son aquellas que son más relevantes en el área de investigación de los laboratorios remotos y que han servido por tanto como punto de partida para el desarrollo del trabajo de investigación que aquí se presenta. Estas plataformas han sido seleccionadas porque a) tienen un área de aplicación común: la electrónica analógica, b) todos presentan un método de control de experimentos diferente y específico para su laboratorio, c) sus equipos de desarrollo son referencia a nivel mundial en el ámbito de la ingeniería remota, tal y como se podrá comprobar en sus referencias bibliográficas y d) se ha podido recabar suficiente información de cada uno de ellos como para poder realizar un análisis exhaustivo de sus metodologías de desarrollo.

Por estas razones, los laboratorios remotos que se ha decidido analizar son los siguientes:

- a) *NetLab*, University of South Australia.
- b) *RemotElectLab*, Politécnico do porto.
- c) *ISILab*, Università degli Studi di Genova.
- d) *VISIR*, Blekinge Tekniska Högskola.
- e) *iLab*, Massachusetts Institute of Technology

Tras el análisis de esta última plataforma se han descrito brevemente otros trabajos de investigación relacionados con los laboratorios remotos. Estos desarrollos son el *LEDFE* de la Universidad Nacional de Rosario, el *eVALIDATE* de la Dayalbagh Educational Institute (India) y el *RwmLAB* de la

Western Michigan University (USA). El primero de estos trabajos se ha incluido debido a la sencillez con la cual se ha implementado una solución para la caracterización de dispositivos, mientras que el segundo de ellos ha sido analizado ya que es de reciente aparición y utiliza soluciones similares a los presentados en este trabajo de investigación. Por su parte, el laboratorio RwmLAB es un trabajo de investigación ampliamente referenciado ya que es uno de los primeros laboratorios remotos que hizo posible la construcción remota de un circuito electrónico previamente diseñado por el alumno a través de un navegador web, aunque posteriormente a su publicación no prosiguió su desarrollo.

Este capítulo finaliza con las conclusiones alcanzadas tras realizar los estudios presentados, tanto en lo referente a las tecnologías disponibles, como a la combinación de las mismas en los diferentes métodos de control ya implementados en otros laboratorios remotos.

2.1 Análisis de tecnologías de control

El capítulo anterior describía cómo un laboratorio remoto está diseñado en base a una arquitectura básica cliente-servidor, por lo que a la hora de la implementación deberemos seleccionar las tecnologías adecuadas para desarrollar a) la aplicación a través de la que el usuario accede e interacciona con el experimento, b) las aplicaciones del servidor que se encargan de gestionar tanto usuarios como experimentos y hacer que el laboratorio sea accesible remotamente, y c) las tecnologías que se encargan de comunicar el servidor con los instrumentos y los experimentos, así como de ejercer el control necesario sobre ellos para llevar a cabo el control remoto de los mismos desde el cliente web.

Tal y como matizábamos, en este trabajo únicamente nos vamos a centrar en el conjunto de procesos que hacen posible que el servidor pueda controlar y comunicarse con los experimentos del laboratorio, dejando fuera del alcance de esta tesis todos aquellos sistemas relacionados con el cliente y ciertas funciones del servidor de un laboratorio remoto, por lo que en esta sección nos centraremos únicamente en analizar las tecnologías disponibles y empleadas para llevar a cabo el control de los experimentos de un laboratorio.

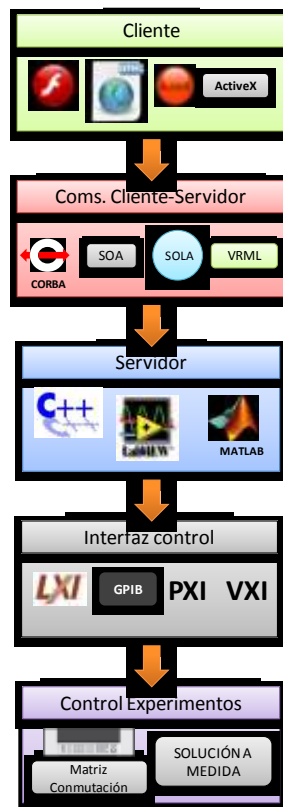


Figura 2.1: Tecnologías para el desarrollo de laboratorios remotos

Aún así y para tener una visión global de las tecnologías comúnmente utilizadas para el desarrollo de un laboratorio remoto, la Figura 2.1 muestra las diferentes alternativas tecnológicas que pueden ser empleadas en su implementación.

En el caso del cliente, las características que pueden primar en el diseño son su universalidad [GOI+09a] (capacidad que dispone el laboratorio remoto para ser accesible por cualquier usuario independientemente de su localización y de las tecnologías desplegadas en su terminal de usuario), seguridad en el acceso [GLH+07b] (tanto desde el punto de vista del terminal de usuario como de los equipos de la universidad), potencia del cliente desarrollado así como las herramientas y servicios proporcionados por la propia tecnología de desarrollo [GOL+09]. De este modo, las tecnologías disponibles van desde el Adobe Flash [GOL+07], pasando por AJAX [Pau05] y tecnologías basadas en HTML en sus diferentes versiones.

Las herramientas más usadas para el establecimiento del enlace de comunicaciones entre el cliente y el servidor son HTML y Java. Así mismo VRML es usada en laboratorio remotos orientados al control de robots [JCT08] [SPC⁺01], así como las tecnologías enmarcadas en los denominados Servicios Web [GB09]. CORBA por su parte fue ampliamente utilizada en el ámbito de la experimentación remota [GMP⁺03] [FGW⁺01] aunque se encuentra en desuso

Como veíamos en el capítulo anterior, son varias las funciones que debe desempeñar el servidor de un laboratorio remoto, por lo que puede que sea una combinación de tecnologías la opción elegida para su implementación, como por ejemplo LabVIEW junto a C++ y MySQL [GOA⁺05]. Por consiguiente, la principal característica a tener en cuenta para la selección de la tecnología a emplear en el servidor es su potencia, pues la universalidad concierne principalmente al cliente mientras que la seguridad en el servidor puede ser implementada con cualquier tecnología. Así pues, sin considerar las funciones de control del experimento concernientes al servidor, las tecnologías más utilizadas para el desarrollo de servidores usados en laboratorios remotos son Java [IGR07], .NET [Ala⁺03] y Python [GLO⁺06b].

Si nos referimos a las tecnologías empleadas para el desarrollo del control de instrumentos y que entran dentro del alcance de estudio de nuestro trabajo, las más utilizadas en este caso son MatLab, C++ y LabVIEW [GM05] [GFB⁺08], tal y como aparecen reflejadas en la Figura 2.1, las cuales son combinadas con otras tecnologías software denominadas drivers que permiten controlar remotamente los instrumentos. Entre los tipos de drivers que analizaremos en detalle más adelante, encontramos los drivers VXIPlug&play y drivers IVI. En la sección 2.1.1 de este capítulo estudiaremos en profundidad estas tecnologías, así como las utilizadas como interfaz de control entre el servidor y los instrumentos, además de las técnicas empleadas para el control de los experimentos.

Así pues las alternativas para la construcción de cada uno de los niveles de la arquitectura de un laboratorio remoto son muchas y muy diversas, cada una con unas características que confieren a su vez al laboratorio de unas cualidades u otras [GFB⁺08] [GM05], y que determinarán en gran manera el funcionamiento del mismo.

Un enfoque erróneo llevado a cabo en algunos desarrollos es comenzar a construir el laboratorio remoto partiendo del experimento que se desea controlar, al que se le agregan posteriormente una serie de aplicaciones software que permiten su disponibilidad online. Puede que en un primer momento esta metodología fuera válida, pero cuando se desea añadir nuevos equipos o funcionalidades al laboratorio es necesario rediseñar por completo la aplicación y modificar la metodología seguida. Es por tanto necesario realizar un análisis previo de estos requisitos, cuyo resultado ayudará a los desarrolladores del laboratorio a determinar cuáles serán las tecnologías más idóneas para utilizar en la construcción del laboratorio remoto.

Por tanto si queremos seguir este planteamiento, los requisitos que deberemos imponer a las tecnologías utilizadas tanto para desarrollar el software de control como para seleccionar el interfaz de comunicación y los mecanismos de control del experimento serán aquellos que hagan posible que cumplamos con los objetivos impuestos al comienzo de este trabajo de investigación. Para ello analizaremos a continuación cada uno de los tres sistemas contemplados en el alcance de nuestra investigación, estudiando cuáles son las opciones tecnológicas disponibles para el desarrollo de cada uno de ellos.

2.1.1 Tecnologías para el desarrollo del software de control

Como ya hacíamos referencia anteriormente, para desarrollar la aplicación software de control que se encargue de gestionar los comandos enviados al experimento y las respuestas generadas y capturadas por los instrumentos dispuestos a tal efecto, se necesita la combinación de dos elementos software:

- a) El software de control propiamente dicho, es decir, los algoritmos que son desarrollados mediante una tecnología determinada y que contienen toda la lógica de control desde el procesado de las peticiones enviadas por el cliente, la validación de los comandos y sus atributos, el envío de estos comandos al experimento y por último la formación de la respuesta devuelta al cliente con los datos obtenidos directamente de los instrumentos del experimento.
- b) Para poder enviar comandos o instrucciones a los instrumentos y leer e interpretar los datos capturados durante el proceso de validación del sistema bajo prueba, el software de control necesita disponer de los

drivers de los instrumentos. El driver es el conjunto de rutinas software que controlan un instrumento programable. Cada una de esas rutinas se corresponde con una operación de configuración, lectura o escritura de datos o toma de medidas en el instrumento. De esta forma se simplifica el control de los instrumentos y se elimina tener que aprender el protocolo de programación de cada instrumento.

Para la codificación del software o aplicación de control que se ejecutará en el servidor en combinación con el resto de sistemas que desempeñan otras funciones del servidor, disponemos de una gran cantidad de tecnologías de escritorio como MatLab, LabVIEW, C/C++, AgilentVEE, .NET, Java, etc.

No existen a priori unos requisitos esenciales que deba satisfacer la tecnología seleccionada para la implementación de esta aplicación de control del servidor, ya que todas las referenciadas anteriormente son lenguajes de programación ampliamente utilizados, siendo quizás las siguientes características las que pueden primar a la hora de seleccionar una u otra:

- Herramientas de desarrollo: esta característica hace referencia a la capacidad de que disponen las tecnologías empleadas en la codificación de los algoritmos de control para llevar a buen término el funcionamiento del laboratorio. En primer lugar es necesario que las técnicas empleadas sean fácilmente accesibles y que no tengan unos requisitos muy específicos de funcionamiento y desarrollo que sean difíciles de mantener y que hagan que el despliegue y uso de la aplicación de control sean complicados.
- Velocidad de desarrollo: tal y como decíamos, prácticamente todas las tecnologías anteriores disponen de librerías o funciones que permiten agilizar el desarrollo de aplicaciones de control complejas, aunque quizás sean LabVIEW y AgilentVEE las tecnologías más idóneas para desarrollar una aplicación de control de instrumentación, ya que ambas han sido creadas específicamente para trabajar con instrumentos y disponen también de entornos de desarrollo y aplicaciones concretas que agilizan la configuración y gestión de los instrumentos del laboratorio.
- Red de desarrolladores: disponer de una amplia red de desarrolladores que conozcan la tecnología y el ámbito de aplicación

de la misma, facilitará el acceso a recursos así como a soluciones ya resueltas previamente por esta comunidad de personas que trabajan en un mismo ámbito aunque cada uno en diferentes aplicaciones.

- **Robustez:** esta característica hace referencia a que la tecnología de desarrollo proporcione herramientas y facilidades que permitan obtener aplicaciones que puedan gestionar de manera autónoma fallos y errores del sistema, sin que en nuestro caso el método de control, falle por completo. Es decir, que la aplicación de control codificada pueda por ejemplo, gestionar errores en los comandos o atributos enviados a los instrumentos o que detecte el mal funcionamiento de alguno de los experimentos, y que no por ello el laboratorio remoto deje de funcionar.

Todas las tecnologías indicadas anteriormente incluyen funciones y aplicaciones orientadas a la interacción con instrumentación, ya sea mediante librerías específicas o subrutinas predefinidas que agilizan la tarea de codificación de la lógica de control, por lo que la selección de una u otra dependerá principalmente de la familiarización que tenga el desarrollador con cada una de ellas.

Aunque si tenemos que decantarnos por alguna de ellas nuestra recomendación es AgilentVEE o LabVIEW, ya que como indicábamos, ambos lenguajes junto con sus correspondientes entornos de desarrollo, fueron creados por Agilent Technologies y National Instruments respectivamente para su funcionamiento en laboratorios de test y medida, por lo que han sido ampliamente probadas y validadas. En nuestro caso, para el desarrollo del software de control incluido en el método propuesto, hemos seleccionado LabVIEW frente a AgilentVEE principalmente por el conocimiento y experiencia previa que disponíamos en el uso de esta herramienta.

Una vez identificada la tecnología a emplear en el desarrollo de la aplicación de control y gestión, hay que seleccionar los tipos de drivers que van a ser empleados para acceder a las funciones de control de cada uno de los instrumentos del laboratorio. Existen diferentes tipos de drivers, desde los más individualizados que únicamente son válidos para un instrumento concreto en función de su marca y modelo, hasta los más generales que permiten con un mismo driver acceder a instrumentos de un mismo tipo o

familia (osciloscopios, multímetros, etc.), pero de diferente fabricante. A continuación hacemos un análisis de cada uno de ellos.

A comienzo de los años 90, un grupo de fabricantes de equipos desarrollaron el Standard Commands for Programmable Instruments (SCPI), el cual define un conjunto de comandos para controlar instrumentos mediante caracteres ASCII, proporcionando una base común para cada conjunto de instrumentos [SCP99]. Aún así, actualmente cada instrumento y proveedor sigue manteniendo un conjunto de comandos propios, lo que complica que un mismo algoritmo de control pueda ser usado sobre instrumentos iguales pero de diferentes fabricantes.

En 1993, la Alianza VXI Plug&play Systems creó un conjunto de especificaciones para los drivers de los instrumentos denominadas VXIPlug&play drivers. A diferencia de SCPI, estos drivers no establecen normas para controlar instrumentos concretos, sino que especifican aspectos comunes en los drivers de un instrumento. Es decir, haciendo uso de este tipo de drivers no es necesario enviar una cadena de texto ASCII al instrumento como se hace en SCPI, sino que se dispone de una serie de rutinas que pueden ser llamadas desde la aplicación de control y que abstraen al programador de los comandos propios del equipo [VXI10].

Junto a este tipo de drivers, una de las grandes aportaciones de la alianza fue el desarrollo del Virtual Instrument System Architecture (VISA). Se trata de una arquitectura software que facilita enormemente la comunicación del PC con el instrumento, poniendo a disposición del programador los servicios y funcionalidades propias de los drivers. Este estándar incluye adicionalmente una serie de especificaciones para la comunicación con interfaces de entrada y salida de datos como pueden ser GPIB, PXI, VXI o TCP (mediante VXI-11), de forma que las aplicaciones de control se abstraen de las diferentes tecnologías de comunicación entre el servidor de control y el instrumento [VIS10].

Sin embargo, los VXIPlug&play drivers no proporcionan un interfaz común de programación. Esto quiere decir por ejemplo, que el programa que controla un multímetro de Agilent Technologies es diferente al que se utilizaría para controlar uno fabricado por National Instruments. Por consiguiente, debido a la inconsistencia existente a la hora de realizar los algoritmos que componen la aplicación encargada de controlar los

instrumentos del banco de test y medida, los programadores deben pasar tiempo aprendiendo el funcionamiento específico de cada driver para poder adaptar el software de control a los instrumentos específicos utilizados en cada situación.

Para superar este gran inconveniente y proporcionar una base común a todos los instrumentos que pudiera ser usada por los programadores, en 1998 se creó el consorcio Interchangeable Virtual Instruments (IVI) Foundation³, del cual son miembros los más importantes fabricantes de instrumentos de medida, competencia unos de otros, pero con el interés común de promover una serie de especificaciones para la programación de instrumentos que aumenten el rendimiento de los mismos, reduzca el coste de su programación y mantenimiento y simplifique el intercambio y reemplazo de instrumentos. A modo de ejemplo, mediante el uso de drivers IVI un mismo algoritmo de control sirve para controlar un multímetro, independientemente de su fabricante. Esta es la mayor motivación detrás del desarrollo de los drivers IVI, ya que la especificación de un driver define una arquitectura abierta para un conjunto de clases de instrumentos y un conjunto de componentes software comunes denominados IVI Shared Components, los cuales proporcionan una serie de servicios comunes como puede ser la administración de los instrumentos desde el sistema de control, tanto a los drivers como a los clientes de los drivers [IVI10a].

La arquitectura definida por el estándar VISA también es implementada por los drivers IVI. Igualmente, las aplicaciones software utilizan los drivers IVI para controlar los instrumentos a través de las funcionalidades de entrada y salida proporcionadas por VISA. Como veíamos, VISA ofrece una capa de abstracción permitiendo que la aplicación software desarrollada sea independiente del interfaz empleado para comunicarse con los instrumentos. Además, en el caso de emplear un interfaz de control basado en Ethernet, VISA permite acceder a funciones del protocolo VXI-11 orientadas a facilitar la identificación de los equipos conectados a la red.

³ <http://www.ivifoundation.org/>. Actualmente esta organización se encarga de gestionar tanto las especificaciones VISA, SCPI como VXIPlug&Play

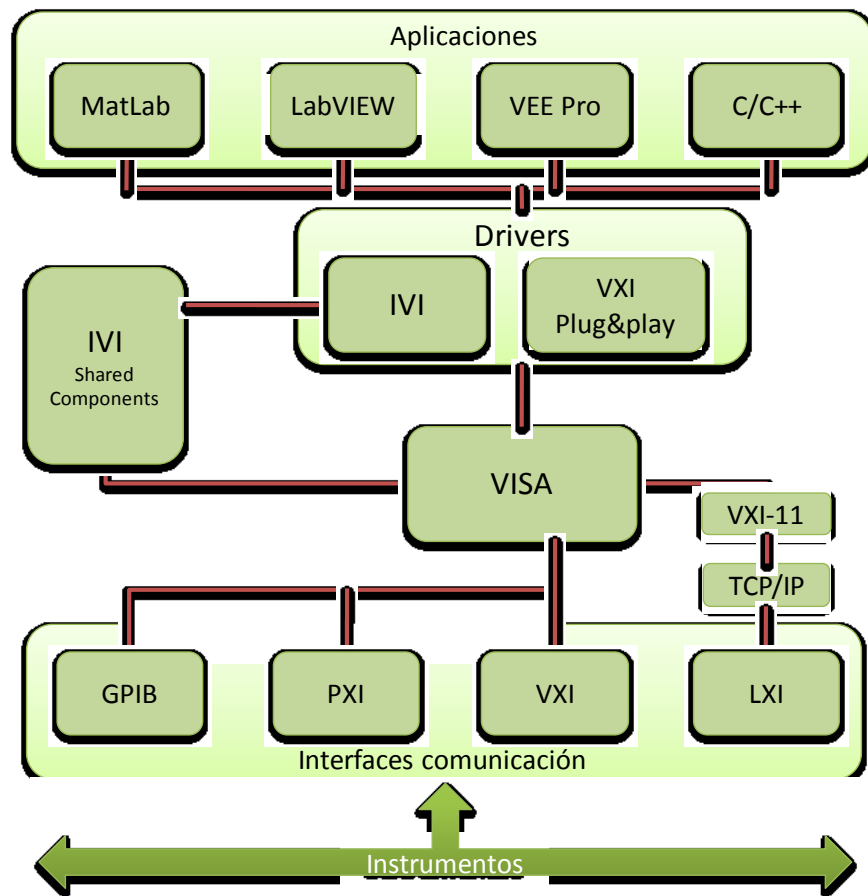


Figura 2.2: Diagrama de componentes utilizados para el control de instrumentación

De manera gráfica, la Figura 2.2 representa agrupadas las diferentes tecnologías analizadas y que son comúnmente utilizadas para el control de instrumentos y/o experimentos de laboratorio. Es importante remarcar que también un experimento diseñado para un propósito concreto puede cumplir con estos estándares de intercambio de información entre el software de control y el experimento, pues en todos los casos los estándares son abiertos y pueden ser implementados para controlar, por ejemplo, una matriz de conmutación no comercial desarrollada para un laboratorio remoto, como sucede en el caso del laboratorio remoto de electrónica analógica VISIR [GZH⁺07].

Entre los dos tipos de drivers descritos y que pueden ser empleados por las aplicaciones de control desarrolladas mediante diferentes tecnologías de programación, actualmente IVI es el que mayores ventajas proporciona frente a VXIPlug&play, ya que nos permitirá cumplir con los objetivos expuesto al comienzo de este trabajo de investigación:

- Favorecer el intercambio: los drivers IVI permiten que un instrumento sea reemplazo por otro de la misma clase sin necesidad de recodificar el algoritmo de control, por lo que se reduce considerablemente el tiempo de integración en nuevos sistemas o ya existentes.
- Consistencia: todos los drivers IVI siguen una estructura software común para el control de los instrumentos, lo que permite ahorrar tiempo a la hora de desarrollar el software de control de un nuevo instrumento.
- Comprobación de valores: este tipo de drivers realizan un chequeo de los valores que se envían al dispositivo para comprobar que se encuentra entre los valores soportados por el equipo.
- Comprobación de estado: los drivers IVI permiten hacer un seguimiento del estado de los instrumentos durante su operación, lo que permite por ejemplo, eliminar comandos redundantes que harían que el funcionamiento del equipo fuera más lento.
- Facilidad de uso: este tipo de drivers proporcionan un acceso fácil, intuitivo y rápido a las funciones de cada instrumento, y que pueden ser accedidas desde diferentes tecnologías software, independientemente del interfaz de control empleado.
- Calidad: al agrupar los instrumentos en diferentes categorías (multímetros, osciloscopios, generadores de funciones, fuentes de alimentación, matrices, analizadores de espectros, generadores RF y medidores de potencia), se asegura que el conjunto de instrucciones de una categoría es válido para todos los instrumentos, independientemente del proveedor.
- Simulación: los drivers IVI permiten que el algoritmo de control realizado pueda ser testeado sin necesidad de disponer del instrumento real, ya que permiten simular el comportamiento del dispositivo.

2.1.2 Interfases de control de instrumentación

Desde el punto de vista del interfaz de comunicaciones, el sistema de control y medida (aplicación software de control + interfaz de control) puede ser configurado de diferente manera en función del bus de comunicaciones seleccionado. Este interfaz de control o bus de comunicaciones es el enlace que permite el intercambio de información entre el servidor que contiene el

software de control y el procesador que se encuentra embebido en los instrumentos o en el experimento. Este procesador se encarga de ejecutar el firmware, es decir, el algoritmo que es capaz de interpretar las acciones realizadas por el usuario a través del interfaz del instrumento (un panel de control, por ejemplo), o las instrucciones enviadas por el alumno desde el cliente web y que son procesadas por la aplicación de control del servidor. Para implementar este enlace, disponemos para ello de dos alternativas, las cuales aparecen reflejadas en la Figura 2.3:

- **Instrumentos modulares:** en este tipo de configuración los componentes del sistema de control y gestión de los instrumentos residen en un único procesador y son compartidos por todos los instrumentos. De este modo, los instrumentos se encuentran embebidos en un chasis o plataforma integral de instrumentación y conectados todos al bus de comunicaciones que les pone en contacto con el procesador. Los chasis más extendidos son PXI y VXI, los cuales utilizan como bus de comunicaciones PCI y VXIBus respectivamente. Estas plataformas de instrumentación, a su vez disponen de interfaces de comunicaciones como GPIB o LAN, lo que hace que también puedan ser controladas y monitorizadas de manera remota.

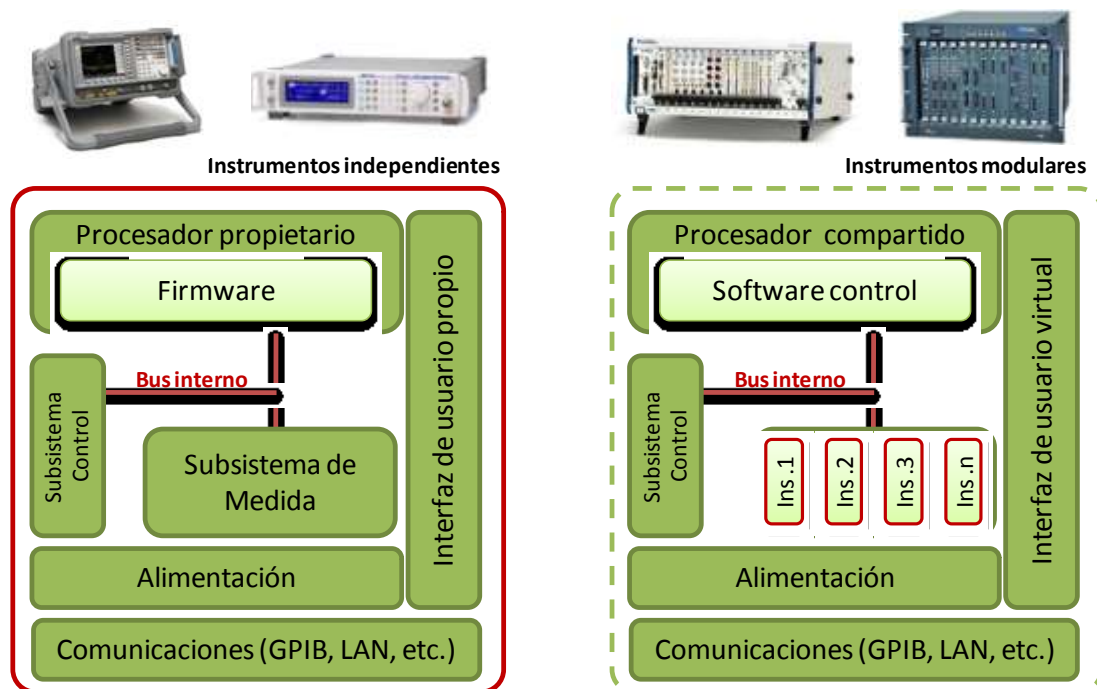


Figura 2.3: Alternativas para la configuración de la comunicación con los instrumentos

- Instrumentos independientes: en esta configuración los instrumentos están conectados al servidor como si se trataran de periféricos. Son instrumentos que disponen de un firmware que interpreta las acciones que el usuario realiza bien sobre su panel frontal de manera local o bien de manera remota a través del bus de comunicaciones y un software de control. Los resultados de las medidas se pueden mostrar bien en el interfaz de usuario que disponen, o bien a través del software de control de manera remota. Para el control remoto de estos equipos, los buses comúnmente más utilizados son GPIB (General Purpose Interface Bus), LAN/ Ethernet y en menor medida RS-232 y USB

A continuación, describiremos de manera somera cada una de estas tecnologías, finalizando con la realización de un análisis comparativo de las mismas con el objetivo de justificar la selección de una tecnología u otra para su uso como interfaz de control en un laboratorio.

A) PCI eXtensions for Instrumentation - PXI

PXI es una plataforma de instrumentación basada en PC desarrollada en sus comienzos por National Instruments con el objetivo de proporcionar nuevos sistemas de automatización y medida, combinando las características del bus PCI (Peripheral Component Interconnect) con la modularidad definida por el formato de placas electrónicas CompactPCI, definidas por los estándares IEEE 1101.1, IEEE 1101.10 y IEEE 1101.11 [Sch08].

Los sistemas PXI constan principalmente de tres componentes [PXI05]:

- Chasis: es el encargado de proporcionar la modularidad al sistema y contener los instrumentos, el controlador y el sistema de comunicación entre ellos. El chasis o bastidor puede albergar de 4 a 18 módulos, entre los cuales, aparte de la instrumentación, se pueden incluir periféricos convencionales, como un lector de CD. Contiene un plano trasero en donde se incluye el bus PCI, así como otros buses de temporización y disparo destinados a aplicaciones que requieran sincronización de eventos.
- Controlador: todos los chasis PXI contienen una ranura para incorporar un controlador que gestione el resto de los módulos incluidos en el bastidor. Los controladores pueden ser de dos tipos:

- Controladores remotos: mediante un enlace MXI (Multisystem eXtension Interface) el PXI puede ser controlado por un PC externo de manera transparente y a alta velocidad. Este tipo de conexión puede incluso habilitar diferentes topologías y permitir el control de varios PXI mediante un único PC.
- Controladores integrados: esta opción elimina la necesidad de disponer de un PC externo al disponer el propio PXI de un ordenador embebido en uno de los módulos, con las mismas características y posibilidades en cuanto a periféricos y conectividad que un ordenador convencional.
- Módulos PXI: PXI es compatible directamente con CompactPCI, así que cualquier instrumento o equipo que siga este formato, puede ser utilizado en un sistema PXI. Actualmente, la alianza de fabricantes PXI cuenta con más de 65 miembros, con lo que la gama de módulos disponible supera el millar.

Gracias a que el hardware PXI está basado en tecnologías utilizadas comúnmente en los ordenadores convencionales, cualquier sistema operativo puede utilizarse para la gestión y control de los recursos incluidos en el PXI, aunque Microsoft Windows es el más popular entre los desarrolladores de aplicaciones. Al ser PXI un estándar promovido inicialmente por National Instruments, las herramientas de desarrollo de este fabricante, como por ejemplo LabVIEW, cuentan con numerosas utilidades que agilizan y facilitan el desarrollo de aplicaciones basadas en PXI. Este estándar proporciona una alta velocidad de transmisión, capaz de alcanzar los 120Mb/s.

B) VME eXtensions for Instrumentation - VXI

El bus VME de 32 bits (Versa Module Europa, ANSI/IEEE 1014-1987) apareció a principios de los años 80 como un sistema de medida automatizado de prestaciones superiores a las del bus IEEE 488. Gracias a contar con la base establecida por GPIB, se convirtió rápidamente en una herramienta muy extendida para el control de instrumentación. La búsqueda de una mayor optimización de este bus de datos provocó en 1987 la formación de un consorcio de fabricantes de equipos con el objetivo de desarrollar instrumentos virtuales conectables al bus de un ordenador convencional. El resultado de estos trabajos fue la norma VXI, que como su nombre indica amplía la norma VME. Así, el consorcio adoptó por un lado

la estructura de mensajes definidas en la norma IEEE 488.2 y los tamaños europeos normalizados para las tarjetas de instrumentos modulares definidos en el estándar VME [VX10].

Un sistema de este tipo está generalmente formado por un bastidor (denominado también mainframe chassis) que contiene un número máximo de 13 ranuras en cada una de las cuales se puede colocar una tarjeta. La primera ranura (módulo 0) está destinada al módulo de control del sistema y las 12 restantes se dedican a los instrumentos modulares [SB07].

El uso del bus VXI se realiza mediante tres configuraciones básicas. La primera consiste en uno o más bastidores VXI unidos a un ordenador externo, a través de una comunicación GPIB mediante el cable y conectores definidos por el estándar. Mediante esta arquitectura, el programador no tiene conciencia de disponer de un equipo VXI, pues toda la programación se realiza como si se tratara de un instrumento GPIB. La segunda configuración hace uso de un ordenador insertado como una tarjeta más en el bastidor VXI, consiguiendo altas prestaciones respecto al rendimiento, ya que se comunica directamente con las tarjetas que componen el sistema. La tercera opción combina los beneficios de las dos anteriores: usa las altas prestaciones obtenidas al insertar un PC en el bastidor, y la flexibilidad de los ordenadores de sobremesa. En esta configuración se utiliza un bus de enlace de alta velocidad, el MXIbus (Multisystem eXtension Interface bus), para conectar directamente el ordenador externo al bus VXI del bastidor. Mediante esta configuración, se pueden llegar a los 20Mbits/seg [VX10].

C) General Purpose Interface Bus – GPIB

El bus GPIB tuvo sus orígenes en el Hewlett-Packard Instrument Bus (HP-IB), una norma para la comunicación y control de experimentos de laboratorio desarrollado por Hewlett-Packard a final de los años 1970, y fue adoptado al mismo tiempo por diversos fabricantes. Esta norma fue posteriormente estandarizada con la referencia como ASE/IEEE Std.488-1978 por el Institute of Electrical and Electronics Engineers (IEEE). Esta norma definía las especificaciones eléctricas, mecánicas y funcionales de este bus que permite tener un ordenador personal que contenga el interfaz específico GPIB controlar hasta un máximo de 15 instrumentos simultáneamente [IEE03].

A partir de esta normativa se continuó trabajando con el objetivo de mejorarla, lo que propició la aparición de la norma ASE/IEEE Std.488.2-1987, la cual definía de una manera más precisa la comunicación mediante el establecimiento de los protocolos de intercambio de mensajes, el formato de datos, y las órdenes generales de los tipos de instrumentos. Finalmente, la formación de un consorcio de empresas proveedoras de instrumentos para mejorar esta actualización de la norma, dio como resultado en 1990 la norma SCPI, a la que ya se hacía referencia anteriormente y que define un conjunto de instrucciones para programar y controlar remotamente los instrumentos de un banco de test y medida.

GPIB se trata de un protocolo de comunicación asíncrono, con ciertas limitaciones respecto al hardware y la tecnología utilizada para su implementación, que hacen que la velocidad máxima de transmisión sea 2Mbits/seg hasta una distancia máxima de 20 metros. Los dispositivos se conectan al bus mediante un cable apantallado de veinticuatro conductores, específicamente diseñado para su uso en el estándar GPIB. Todas estas líneas trabajan con lógica TTL negativa, pues utilizan circuitos lógicos con salidas en colector abierto, característica que hace que el cable se comporte como una línea de transmisión, por lo que se debe limitar el número de equipos conectados (15) y la distancia física entre dispositivos conectados al bus (20 metros). Estas especificaciones hacen que la implementación de una red de control basada en el estándar GPIB requiera que el servidor disponga del interfaz propio de GPIB y de los cables propios del estándar. Esto en sí no es una limitación, pues hoy en día la mayoría de los fabricantes dotan a sus equipos de un interfaz GPIB, que han hecho de GPIB una tecnología madura y muy extendida.

D) LAN eXtensions for Instrumentation - LXI

El estándar LAN eXtensions for Instrumentation (LXI) es un estándar promovido y desarrollado por el LXI Consortium, el cual fue creado en 2004 y que en septiembre de 2005 sacó a la luz la especificación de LXI. Este estándar está basado en tres tecnologías principalmente: Ethernet, interfaces web y el Precision Time Protocol (PTP, IEEE 1588). Así, el objetivo del consorcio es incrementar la interoperabilidad y la funcionalidad de los instrumentos que disponen de un interfaz Ethernet, mediante la estandarización de interfaces y sistemas de control que sean comunes a todos

los instrumentos de una misma clase, ya sean osciloscopios, generadores de funciones, multímetros o el resto de instrumentos recogidos en la normativa [LXI08]. LXI es una evolución de los interfaces de control a los que hacíamos referencia anteriormente y que gráficamente se aprecia en la Figura 2.4 [Wol06].

LXI combina las ventajas de los instrumentos denominados *rack&stack* y de los interfaces GPIB así como los beneficios de los sistemas modulares y compactos definidos por VXI y PXI superando los inconvenientes derivados de estos sistemas, como pueden ser la necesidad de disponer de un controlador específico y un formato de instrumentación basado en tarjetas como sucede en el caso de VXI y PXI o las limitaciones en cuanto a la tasa de transmisión de GPIB (2Mbits/seg) y el número máximo de dispositivos a controlar a través de un mismo bus.

La principal característica de los sistemas de test y medida que utilizan LXI como interfaz de control es que al estar basados en Ethernet, la velocidad de transmisión de información entre los dispositivos y los controladores puede ser elevada. Además, protocolos basados en Ethernet como el Dynamic Host Configuration Protocol (DHCP) también son implementados por los equipos LXI, lo que facilita y agiliza enormemente las tareas de configuración y direccionamiento de la información hacia los instrumentos que forman la red de test y medida del laboratorio.

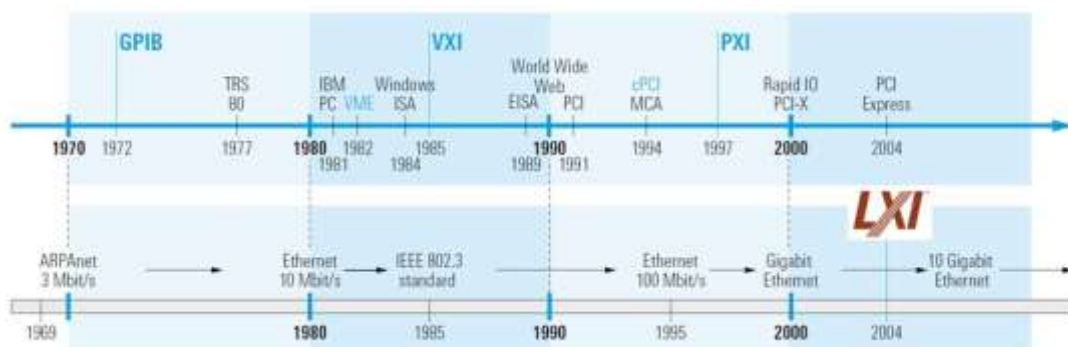


Figura 2.4: Evolución de los interfaces de control

Junto con estas ventajas, el servidor web integrado en los equipos LXI permite que de una manera sencilla, rápida e intuitiva se pueda mediante un sencillo navegador web acceder a los controles de los instrumentos de manera remota, sin requerir ninguna programación especial, aunque dependiendo del equipo sí puede ser necesaria la instalación de algún plugin en el navegador. La Figura 2.5 muestra el interfaz web proporcionado por un generador de funciones LXI, el cual coincide con el panel frontal del equipo real.

El estándar LXI define tres clases de instrumentos, A, B y C cada una de ellas con una serie de funcionalidades las cuales son heredadas jerárquicamente, siendo los instrumentos de clase C son los más simples, y los pertenecientes a las clases B y A ofrecen opciones más avanzadas, tal y como mostramos gráficamente en la Figura 2.6.

Los instrumentos LXI de Clase C disponen de las funciones básicas de LXI, como son: a) la capacidad de ser reconocidos directamente en una red LAN gracias al protocolo de descubrimiento VXI-11, b) ofrecer un servidor web para acceder a todas las opciones de control y remoto y configuración del equipo a través de un navegador web, y c) permiten su programación mediante drivers IVI.



Figura 2.5: Interfaz web de un generador de funciones LXI

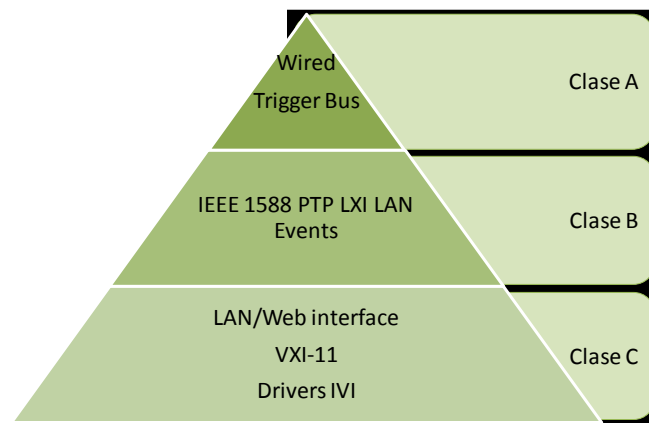


Figura 2.6: Clases de instrumentos LXI

Los instrumentos LXI Clase B, presentan las funciones de la clase C y también ofrecen un mecanismo de sincronización definido en el estándar IEEE 1588 conocido como Precision Time Protocol o PTP, el cual fue definido específicamente para la sincronización de los sistemas de test y medida configurados entorno a una red local de datos. De este modo, sobre una red LAN 100BaseT se pueden conseguir tiempos de sincronización entre los equipos LXI conectados de hasta 10ns. Esto elimina los problemas de latencia de la tecnología LAN y asegura una sincronización precisa en el sistema de medida.

Junto con las funcionalidades de las dos clases anteriores, los equipos LXI Clase A están equipados con un interfaz de disparo de 8 canales definido en el estándar, mediante el cual los instrumentos se pueden conectar en una topología de estrella o bus. Esta funcionalidad permite que los instrumentos puedan coordinarse para generar y recibir solicitudes de disparo de medidas mediante mensajes LAN (TCP o UDP), lo que permite a los equipos comunicarse entre sí sin necesidad de un controlador.

Gracias a las ventajas relacionadas con la flexibilidad, interoperabilidad, robustez en las comunicaciones y las facilidades proporcionadas por el estándar, LXI se está convirtiendo en el interfaz de comunicaciones más utilizado en los actuales sistemas de test y medida.

Desde el punto de vista de la programación de los instrumentos que dispongan del interfaz LXI, al implementar éstos el estándar IVI desde su clase más sencilla, es posible desarrollar aplicaciones de control de manera fácil y rápida y que así puedan ser utilizadas para controlar instrumentos de

una misma familia sin tener que modificar los algoritmos de control. Únicamente son necesarias ciertas modificaciones en la configuración del direccionamiento de la información desde la aplicación de control, pero como el estándar LXI implementa entre otros los protocolos VXI-11 y DHCP, detectar nuevos instrumentos conectados a la red y asignarles una dirección IP es sencillo y rápido.

2.1.2.1 Selección del interfaz de control de instrumentos

GPIB ha sido desde principios de los años 70, cuando se comenzaron a integrar las medidas en instrumentos con ordenadores, el estándar más probado y estable empleado en sistemas de test y medida. En el área de la experimentación remota, muchos de los laboratorios que utilizan instrumentación para realizar pruebas y medidas sobre los experimentos han utilizado este interfaz satisfactoriamente [GLH⁺07a] [ATF⁺05]. Los instrumentos modulares basados en PXI y VXI también han sido y son actualmente muy utilizados en laboratorios remotos, principalmente por tratarse de sistemas compactos y porque integran en un único equipo todos los instrumentos del laboratorio [NM07] [Gus03a].

Desde el año 2007, en el equipo de investigación WebLab-Deusto llevamos realizando pruebas y desarrollos orientados a estudiar las oportunidades de algunas de estas tecnologías, de modo que fruto de esta experiencia hemos obtenido y desplegado los siguientes laboratorios remotos:

- WebLab-GPIB: es el laboratorio remoto creado en torno al estándar GPIB, en el cual el objetivo es que los alumnos sean capaces de aprender a controlar equipos que implementen este interfaz de comunicaciones [GLH⁺07a]. Fruto de este trabajo hemos conocido las limitaciones del mismo, principalmente debido a la falta de una normativa común a los equipos utilizados en lo referente a los comandos de control a utilizar en cada una de las familias de instrumentos usados, lo que nos obliga a adaptar el laboratorio ante el cambio de unos instrumentos por otros. Además, la especificidad de las tarjetas de control necesarias a incluir en el PC que funciona a modo de servidor, así como las restricciones impuestas por los cables físicos utilizados en el laboratorio, hacen complicado posibles

modificaciones del hardware del laboratorio, así como su propio emplazamiento.

- VISIR-Deusto: tal y como también comentaremos en el capítulo cuarto, desde el año 2008 el equipo de investigación WebLab-Deusto ha desplegado en la Universidad de Deusto el laboratorio remoto VISIR orientado a la electrónica analógica [GGH+10] [GZH+07]. Este laboratorio está compuesto de una serie de instrumentos incluidos en un PXI así como de una matriz de conmutación diseñada a medida para este laboratorio. Gracias a esta colaboración, hemos estudiado y analizado el funcionamiento de los equipos PXI, así como de los drivers VXIPlug&play utilizados para el control de los instrumentos incluidos en el laboratorio.

Como resultado de esta experiencia práctica adquirida durante el trabajo de despliegue y desarrollo de estos dos laboratorios remotos, así como del estudio realizado del resto de interfaces de control analizados previamente en este capítulo, hemos valorado cada uno de estos estándares en base a los siguientes indicadores y requisitos:

- Coste: este indicador refleja los costes que deben ser considerados a la hora de realizar el despliegue del sistema de control: cables especiales, tarjetas, bastidores, controladores específicos, etc. En este caso, un mayor número de asteriscos en su valoración indica un bajo precio, lo que supone una ventaja a la hora de realizar cualquier despliegue tecnológico. Por el contrario, un único asterisco, indica un alto precio, lo que supone una desventaja desde el punto de vista del despliegue.
- Velocidad de transmisión: este indicador hace referencia a la tasa de transmisión máxima que permite la tecnología de control elevada. Quizás este indicador no sea crítico en ciertos laboratorios remotos, pero debe ser considerado en aquellos en los que se realicen aplicaciones de control en tiempo real y requieran cierto ancho de banda.
- Compatibilidad: este indicador hace referencia a la capacidad que tiene la tecnología para interoperar con otras tecnologías existentes y a poder utilizar un mismo algoritmo de control sobre equipos de distintos fabricantes.

	PXI	VXI	GPIB	LXI
Coste	*	*	***	****
Velocidad Transmisión	****	****	**	***
Compatibilidad	**	**	***	****
Flexibilidad	*	*	**	****
Longevidad	***	***	**	****
Facilidad de uso	**	**	***	****
Tamaño	****	****	***	***

Tabla 2.1: Análisis comparativo entre las tecnologías de control para laboratorios remotos

- **Flexibilidad:** relativa al número de instrumentos que pueden ser controlados, la distancia entre el sistema de control y los equipos.
- **Longevidad:** hace referencia a la capacidad de la tecnología de subsistir en el tiempo en base a la disponibilidad de equipos que la implementen de manera que puedan seguir siendo usados.
- **Facilidad de uso:** hace referencia a la sencillez de uso de la tecnología por parte del usuario y por parte del desarrollador.
- **Tamaño:** hace referencia al tamaño de los equipos y a las limitaciones en cuanto a requisitos de acondicionamiento especial: temperatura continua, alimentación, etc.

El resultado de este estudio, aparece reflejado en la Tabla 2.1, en la que se recoge el análisis comparativo que hemos llevado a cabo entre los diferentes interfaces de control que pueden ser utilizados para comunicar la aplicación de control del servidor con los instrumentos del laboratorio [GHA⁺¹⁰].

En lo relativo al coste de los equipos, los estándares VXI y PXI están basados en chasis o bastidores que presentan un elevado coste debido principalmente a que requieren una fuente de alimentación común para todos los módulos y de un potente sistema de refrigeración que mantenga la temperatura del bastidor en unos niveles determinados. Si se quiere también seguir la filosofía de la modularidad representada por estos sistemas, los controladores que requieren ser instalados para controlar todos los módulos presentan un elevado coste ya que son sistemas hechos a medida para estos bastidores y que presentan pocos beneficios frente a los ordenadores o controladores convencionales. Por su parte GPIB necesita que el servidor de

control tenga instalada una tarjeta específica GPIB, además de necesitar los cables definidos en el propio estándar para poder comunicarse con los equipos bajo control. En cambio LXI, al estar basado en el estándar Ethernet, el cual se encuentra ampliamente extendido, el incremento de coste de los equipos por disponer de este interfaz es mínimo, así como los cables y dispositivos necesarios para crear la red de control de área local necesaria para comunicar el servidor con los instrumentos.

Respecto a la comparación realizada relativa a la velocidad de transmisión de datos en estos estándares, PXI y VXI debido a su propia naturaleza modular presentan el mejor rendimiento respecto a este indicador alcanzando velocidades de hasta 120Mb/s. Y decimos rendimiento ya que si bien LXI al estar basado en Ethernet puede alcanzar velocidades de hasta 10Gb/s, el valor típico alcanzable con instrumentos LXI-C en una red local es del 100Mb/s, y además se debe considerar que pueden surgir retardos en las respuestas de los equipos debido a la latencia inherente en este tipo de comunicaciones, si bien puede solventarse este problema empleando la sincronización proporcionada por los equipos LXI de Clase B o Clase A. En nuestro caso, únicamente hemos podido experimentar con instrumentos de Clase C, los cuales no implementan ningún mecanismo de sincronización que puedan ayudar a paliar estos problemas, por lo que por esta razón le hemos otorgado una valoración inferior. En cuanto a GPIB, este estándar presenta limitaciones en cuanto a la distancia a la que se pueden situar los equipos, la velocidad de transmisión y el número máximo de equipos a controlar mediante un único interfaz, por lo que por ello ha obtenido la valoración inferior de las cuatro.

En cuanto a la compatibilidad, los equipos que únicamente disponen del interfaz GPIB solo pueden ser controlados a través de él, o bien utilizando conversores adicionales, generalmente al estándar USB o RS-232, lo cual implica la instalación de algún driver adicional o una codificación especial en el servidor de control. También tal y como veíamos previamente, GPIB hace uso del estándar SCPI, por lo que los comandos utilizados para controlar un instrumento son propios del fabricante y del propio equipo, lo que implica una recodificación del algoritmo de control en caso de sustituir un equipo por otro. Por su parte los equipos VXI y PXI pueden disponer de algún otro interfaz de comunicaciones como GPIB, lo que puede ayudar a aumentar la distancia entre los equipos y el servidor de control. En lo

referente a los algoritmos de control los equipos incluidos en estos chasis son mayoritariamente controlados mediante drivers VXIPlug&play lo que hace que también sean específicos de cada instrumento. En cambio, todos los equipos LXI independientemente de la clase a la que pertenezcan, implementan el estándar de control IVI, lo que permite, tal y como veremos más adelante, que un mismo algoritmo de control pueda ser utilizado para gestionar remotamente equipos de diferentes fabricantes.

Obviamente LXI es el estándar más flexible de los cuatro analizados, principalmente debido a la naturaleza Ethernet, la cual permite un número casi ilimitado de equipos conectados a la misma red de test y medida, así como una mayor distancia entre los equipos y el servidor de control. En cambio, los chasis PXI y VXI pueden contener un número determinado de equipos y en la gran mayoría de situaciones, para aumentar el rendimiento del sistema, el módulo de control se encuentra embebido en el propio chasis.

De igual modo, respecto a la longevidad del estándar es LXI el que más probabilidades posee de permanecer en el mercado y extenderse más fácilmente, aunque con ello no indiquemos que VXI y PXI tiendan a desaparecer. Al contrario de lo que sucede con GPIB, ya que se advierte cómo cada vez menos equipos incluyen este interfaz de control entre los disponibles de serie proporcionados por los fabricantes.

En lo referente a la facilidad de uso, un equipo LXI configurado para obtener una dirección IP de manera automática puede ser controlado al instante desde un navegador web, tecnología que ninguno de los otros interfaces implementan. Es decir, tanto en VXI como PXI y GPIB es necesario realizar algoritmos de control para poder gestionar los equipos del laboratorio. En cambio, si se desea realizar un algoritmo para controlar un instrumento LXI, las facilidades proporcionadas por el estándar IVI implementando por todas las clases hace que sea rápido y sencillo obtener una solución de control que pueda adaptarse a todos los instrumentos de una misma familia, tal y como veremos con más detalle en el capítulo cuarto de este trabajo.

Finalmente, gracias a la modularidad de PXI y VXI el tamaño de un sistema de test y medida que haga uso de los mismos es inferior al mismo sistema si lo realizamos mediante equipos individuales ya implemente

interfaces GPIB o LXI, si bien el tamaño de los instrumentos no debería ser un inconveniente a la hora de diseñar y desplegar un laboratorio remoto.

En base a las conclusiones derivadas de nuestra experiencia y del análisis realizado de los interfaces de control indicados previamente, se puede determinar que el estándar LXI se perfila como una de las mejores alternativas a la hora de comunicar y controlar los instrumentos implicados en un sistema de test y medida desplegado en torno a un laboratorio remoto [HG11]. Al tratarse LXI de un estándar basado en Ethernet, cualquier equipo que soporte este interfaz está predispuesto a ser controlado remotamente con unas altas capacidades y prestaciones. Si a esto le sumamos que los equipos LXI pueden ser programados de manera rápida y sencilla gracias al uso de drivers IVI, podemos realizar algoritmos de control que sean fácilmente reutilizables en diferentes escenarios con diferentes equipos [GHA⁺08].

2.1.3 Tecnologías para el control de experimentos

Tal y como se ha explicado anteriormente, el objetivo de un laboratorio remoto es hacer posible que un usuario pueda realizar sobre los experimentos las mismas acciones que si se encontrara en un laboratorio manual pero de manera remota.

En ambos tipos de laboratorios, manual y remoto, el experimento está formado por el sistema bajo prueba objeto de la experimentación y por el conjunto de instrumentos utilizados para realizar las pruebas de test y medidas necesarias para comprobar el funcionamiento del propio sistema bajo prueba que, en el caso del contexto en el que focalizamos nuestro trabajo de investigación, se trata de un circuito electrónico como ya exponíamos previamente.

Veámos en las secciones anteriores que la combinación de una aplicación de control que utilice los drivers de los instrumentos, junto con el empleo de un interfaz de comunicaciones específico entre el software de control y los instrumentos, hace posible que el usuario acceda remotamente a los equipos del laboratorio a través de un cliente web desarrollado a tal efecto.

Queda por tanto definir en el alcance de nuestro trabajo las técnicas y tecnologías disponibles para poder controlar remotamente el sistema o

circuito bajo prueba sobre el cual el alumno realizará los ejercicios contemplados en su práctica. Para ello, vamos a partir de los requisitos y necesidades que queremos imponer al sistema de control del experimento que integraremos en nuestro método de control:

- El conjunto de técnicas y tecnologías utilizadas deberán hacer posible que el usuario pueda construir de manera remota un circuito electrónico analógico, pudiendo seleccionar los componentes discretos a utilizar y las conexiones existentes entre ellos.
- Una vez construido el circuito, el alumno podrá conectar los instrumentos que considere oportuno para realizar los test y las medidas necesarias para estudiar y analizar su comportamiento.
- El alumno podrá modificar los componentes del circuito y las conexiones en función de sus necesidades y las especificaciones del mismo dictadas por el profesor, tal y como sucede en un laboratorio manual.
- El sistema de control diseñado deberá poder crear circuitos complejos, es decir, aquellos que tengan un número elevado de nodos y mallas.
- Como parte de su proceso de aprendizaje, el alumno podrá cometer fallos en la construcción del circuito, pero siempre manteniendo unos límites de seguridad que impidan la destrucción o mal funcionamiento del sistema de control.
- Una vez diseñado y desarrollado el sistema de control y medida, éste no debe introducir fallos o errores significativos en el funcionamiento del circuito creado o en las medidas adquiridas. Estos errores deben ser mínimos y mantenerse dentro de unos márgenes de tolerancia.
- El sistema resultante debe proporcionar repetitividad en las medidas realizadas sobre el circuito bajo prueba, proporcionando por tanto fiabilidad y robustez al sistema.
- La velocidad de respuesta del sistema debe ser óptima para proporcionar al alumno sensación de control en tiempo real y que el funcionamiento del laboratorio remoto que integre el sistema de control no tenga retardos elevados en las respuestas, haciendo que el usuario tenga que esperar para visualizar los resultados de su experimentación en el interfaz web del cliente.

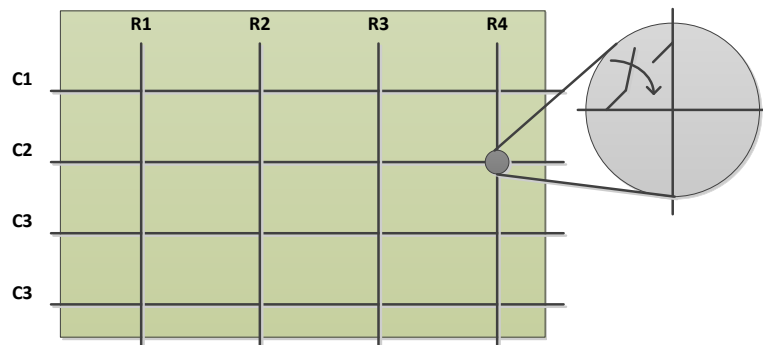


Figura 2.7: Ejemplo de matriz de conmutación

Para realizar un sistema de control del experimento que satisfaga en mayor o menor medida los requisitos que acabamos de describir, existen básicamente dos posibilidades, las cuales se diferencian principalmente en utilizar instrumentos comerciales o soluciones totalmente propietarias, siendo en ambas alternativas la matriz de conmutación el elemento principal.

Una matriz de conmutación es un circuito hardware que hace posible la interconexión entre sus 'n' filas disponibles y sus 'm' columnas. Es decir, mediante el control de los relés situados en los cruces existentes entre los canales de la matriz, los cuales están distribuidos en filas y columnas, es posible dirigir una señal procedente de un canal 'n' a un canal 'm', pudiendo ser cada canal entrada o salida en función del sentido de flujo de cada señal [Joh02]. Gráficamente, podemos observar una matriz de conmutación de 4x4 (cuatro filas y 4 columnas) en la Figura 2.7. A la derecha de la figura, vemos una ampliación del cruce existente entre la columna C2 y la fila R4. Accionando este relé, los elementos conectados en cada uno de dichos canales estarían interconectados. De este modo, cada uno de estos cruces representaría un nodo de un circuito electrónico.

A) Sistema de control mediante instrumentos comerciales

La configuración hardware de la matriz de conmutación comercial está formada por:

- Un mainframe encargado de contener las matrices de conmutación, gestionar las comunicaciones y ejecutar los comandos enviados desde una aplicación de control o introducidos a través del propio panel de control.
- Las propias matrices de conmutación que se conectan en los slots o ranuras dispuestas a tal efecto en el mainframe.



Figura 2.8: Ejemplo de sistema comercial de matriz de conmutación comercial

Estos dos sistemas pueden ser independientes o no, pero se necesitan el uno del otro para poder desempeñar sus funciones. Es decir, el mainframe sin matrices no tiene ninguna utilidad y las matrices por sí solas no disponen de ningún sistema que permita su control. Así, el mainframe es el encargado de la lógica, es decir, de interpretar las órdenes que llegan al firmware del instrumento, mientras que las matrices de conmutación se encargan de la ejecución física de las conexiones mediante los relés de que disponen a tal efecto.

La Figura 2.8 muestra un ejemplo de una matriz de conmutación configurada con este sistema, en donde en primer plano vemos los módulos que contienen las matrices y en segundo plano, el mainframe en cuyos slots se conectan dichas matrices.

Como esta solución está basada en un instrumento comercial, su control se puede integrar directamente en la aplicación dedicada a gestionar el resto de instrumentos del laboratorio, utilizando para ello su propio sistema de drivers. Además, como cualquier otro instrumento dispone también de los interfaces de control descritos anteriormente.

Empleando una matriz de conmutación comercial como elemento principal del sistema de control del experimento, únicamente se debe incluir algún mecanismo hardware que permita conectar los componentes e instrumentos a los canales de las matrices de conmutación contenidas en el mainframe de la matriz de conmutación.

Si también se quiere ampliar la matriz de conmutación, aumentando así el número de nodos de los posibles circuitos resultantes, se deberá proporcionar un mecanismo para interconectar las matrices de conmutación

contenidas en el mainframe, sin que ello implique una recodificación del software de control del instrumento.

B) Sistema de control mediante soluciones propietarias

Como decíamos anteriormente, muchos diseños de laboratorios remotos comienzan desarrollando en primer lugar el experimento al que posteriormente se le añaden capas de software. Esto suele suceder porque partiendo del experimento se adoptan soluciones propietarias diseñadas específicamente para el experimento o laboratorio en cuestión.

Sucede lo mismo en el caso de emplear una matriz de conmutación propietaria como eje del sistema de control del experimento destinado a hacer posible que el usuario construya y compruebe su propio circuito electrónico de manera remota. Generalmente se trata de soluciones únicas como las desarrolladas en los laboratorios RemotElecLab [SAG10], ISILab [SBP+05] o VISIR [GOA+05] entre otros.

En estos sistemas se han desarrollado unas matrices de conmutación específicas acorde a los requisitos del sistema de cada institución y laboratorio remoto en el cual se han desarrollado. Básicamente consisten en sistemas controlados mediante microcontroladores que se encargan de abrir o cerrar unos relés determinados que permiten seleccionar unos circuitos u otros de entre un conjunto de circuitos disponibles, o bien seleccionar los componentes a utilizar en el circuito bajo prueba de entre una serie de componentes determinados. En algunas de estas matrices es posible también seleccionar las conexiones entre los componentes, aunque dependiendo de la configuración del sistema, el número de conexiones puede ser limitado o necesita redundar los componentes. Posteriormente analizaremos con más detalle cada una de estas soluciones.

Así pues, al ser soluciones propietarias controladas en su mayoría por sistemas basados en microcontroladores, aparte de desarrollar el software de control que traducirá las órdenes enviadas por el usuario en comandos entendibles por los microcontroladores, es necesario codificar el firmware específico que se ejecutará en estos microcontroladores y que determinará qué relés se abren o cierran en cada momento.

El interfaz de comunicaciones entre la aplicación software y el sistema de control del experimento estará basado en sistemas de comunicaciones

como RS-232 o USB. Aún siendo RS-232 un estándar que fue usado en instrumentación hoy en día ya no se utiliza como interfaz de control de equipos de laboratorio, siendo sustituido por GPIB o LXI. USB es un interfaz de comunicación para periféricos de ordenadores que en algunos instrumentos se incluye con el objetivo de facilitar la toma de datos del equipo o imprimir resultados.

2.1.3.1 Selección del sistema de control de experimentos

Un sistema para el control de los experimentos diseñado mediante una matriz de conmutación comercial tiene la ventaja de que al tratarse de un instrumento estándar, puede ser tratado como un equipo más de los disponibles en el laboratorio, estando su control integrado en la aplicación de software y pudiendo usarse un interfaz de comunicaciones como los expuestos anteriormente. En cambio, si se utiliza una matriz de conmutación propietaria, el software de control será único para dicha solución, no siendo posible utilizar otra matriz de conmutación para controlar los experimentos del laboratorio.

En cuanto al interfaz de comunicaciones utilizado entre la aplicación de control y la matriz de conmutación, si empleamos un instrumento comercial podremos beneficiarnos de las características proporcionadas por los interfaces incluidos por el fabricante destinados específicamente al control de instrumentación, siendo sus prestaciones mejores que aquellos interfaces de propósito general (USB, RS-232, etc.) que se usarán en el caso de utilizar una matriz de conmutación propietaria.

En el caso de utilizar una matriz de conmutación propietaria, la capacidad del sistema respecto al número de nodos del circuito o número de circuitos a contener en el experimento depende exclusivamente del diseño que se realice de la misma. Pero sí podemos predecir que a mayor número de nodos, la capacidad del sistema aumenta al igual que la dificultad del software de control, ya que también se incrementan el número de relés a gestionar. Además, en una matriz propietaria se deben implementar también las conexiones físicas entre los nodos y los componentes, mientras que en una matriz comercial, las conexiones existen previamente y únicamente se deben habilitar mecanismos para emplazar los componentes y los terminales de prueba de los instrumentos.

En una matriz de comercial su capacidad está limitada por el número de slots del mainframe disponible y el número de columnas y filas de cada una de las matrices incluidas en dichos slots. Por tanto, si se quiere aumentar la capacidad de la matriz de conmutación resultante se deberán habilitar mecanismos para interconectar las matrices conectadas en los slots del mainframe principal.

En términos generales, una matriz de conmutación basada en un instrumento comercial puede presentar ventajas frente a una desarrollada expresamente para el sistema de control de los experimentos de un laboratorio, principalmente gracias a su fácil integración como un instrumento más en el método de control del laboratorio.

En cambio, al tratarse de una solución cerrada proporcionada por el fabricante, se deben emplear mecanismos externos para ampliar su capacidad. Además, los relés utilizados en estas matrices son los incorporados por el fabricante, por lo que su velocidad de conmutación, así como la resistencia introducida en el circuito debida a sus contactos, son parámetros no seleccionables durante el proceso de diseño y que pueden afectar a la velocidad de respuesta del sistema y a la exactitud en las medidas tomadas sobre el circuito bajo prueba.

En cualquier caso, la elección de un sistema de control de experimentos u otro recae en el diseñador, el cual empleará el método que mejor se adapte a los requisitos planteados por el escenario en el cual se desplegará el laboratorio remoto. En el capítulo cuarto se compara la solución propuesta basada en una matriz comercial con otra solución que emplea una matriz propietaria, lo que permite obtener más conclusiones a la hora de seleccionar una tecnología y otra.

2.1.4 Conclusiones

En las secciones anteriores se han analizado las alternativas tecnológicas existentes para desarrollar un método de control que pueda ser aplicado en laboratorios remotos, compuesto por una aplicación software, un interfaz de comunicaciones entre la aplicación software y el experimento, y un sistema de control del experimento del laboratorio compuesto del sistema bajo

prueba y la instrumentación necesaria para comprobar y validar su funcionamiento.

La elección de las tecnologías en cada uno de los subsistemas que forman el método de control, es una elección propia del equipo de diseño y desarrollo, entrando en juego factores como el conocimiento y experiencia en el uso de las tecnologías, su disponibilidad, su precio, el rendimiento y servicio que se desea ofrecer al usuario del laboratorio y el propio tipo de laboratorio remoto en el cual se va a incluir el método de control.

En base a nuestro estudio, el método de control que presentamos está diseñado y desarrollado utilizando las siguientes tecnologías:

- Aplicación de control: utilizaremos LabVIEW como tecnología de desarrollo, principalmente por su diseño específico para trabajar con instrumentación de laboratorio al incluir su entorno de desarrollo funciones, servicios y facilidades que agilizan la codificación de la lógica de control de los instrumentos y experimentos. Para el control de los equipos utilizaremos drivers IVI con el objetivo de abstraer la aplicación software de los instrumentos, sin vincular de esta manera la lógica de control a unos equipos específicos, consiguiendo a su vez independizarla del interfaz de comunicaciones empleado.
- Interfaz de control: utilizaremos LXI como tecnología de comunicación entre el software de control y el experimento, debido a la facilidad de configuración de la red de comunicaciones con los instrumentos, y a las características propias del estándar: velocidad adecuada, número de equipos controlados y sin la necesidad de emplear un equipamiento especial.
- Sistema de control del experimento: utilizaremos como instrumento clave una matriz de conmutación comercial la cual hará posible que el usuario cree circuitos electrónicos seleccionando los componentes y las conexiones existentes entre ellos. Añadiremos a esta matriz un hardware mínimo con el objetivo de conectar a la matriz los componentes y los instrumentos del experimento. Durante la fase de pruebas analizada en el capítulo cuarto, examinaremos el rendimiento respecto al tiempo de respuesta y fiabilidad en las medidas adquiridas proporcionadas por el sistema de control elegido.

Como citábamos previamente, hemos llegado a estas conclusiones después de realizar el análisis de las tecnologías disponibles y del estudio de los métodos de control desarrollados por otras instituciones que cuentan con laboratorios remotos en el área de la electrónica analógica. La siguiente sección de este capítulo recoge este estudio, finalizando el capítulo con el análisis comparativo de estas soluciones y las conclusiones así extraídas.

2.2 Antecedentes de metodologías de control en laboratorio remotos

El objetivo de esta sección es analizar los métodos de control de los experimentos incluidos en aquellos laboratorios remotos que son referentes a nivel internacional en el ámbito de la experimentación remota orientada a la docencia de la electrónica analógica.

En primer lugar hemos descrito la arquitectura de cada uno de los laboratorios para posteriormente pasar a centrarnos únicamente en el estudio de las técnicas utilizadas en cada laboratorio para controlar los experimentos, dejando fuera del análisis el estudio de otros sistemas, como pueden ser el cliente u otros subsistemas desarrollados en el servidor (gestión de conexiones, gestión de usuarios, sistema de reservas, etc.). Igualmente se han analizado cada una de estas soluciones de control en profundidad para determinar cuáles son las ventajas e inconvenientes de las metodologías empleadas en cada caso, para posteriormente poder compararlas en función de unos criterios determinados y proponer a partir de este estudio y del análisis de las tecnologías realizado en la sección 2.1 un nuevo método de control que supere las limitaciones de los sistemas actuales.

En el análisis relativo al sistema de control de experimentos de cada una de estas metodologías no se ha podido hacer una caracterización cuantitativa que nos otorgue datos relativos a su velocidad de respuesta, repetitividad en las medidas o errores inherentes a los propios sistemas de medida ya que no hemos dispuesto de las herramientas ni accesos necesarios para ello, por lo que hemos realizado únicamente un examen cualitativo de los mismos. Sin embargo, gracias a nuestra participación en el proyecto VISIR, sí hemos podido tomar su sistema de medida como referencia y compararlo con el que en este documento se propone. Los resultados y conclusiones de estas pruebas las examinamos en el capítulo cuarto.



Figura 2.9: Página web del NetLab

Una vez analizadas todas las metodologías, las hemos cotejado entre sí para determinar las oportunidades de mejora y las aportaciones que se quieren defender con este trabajo de investigación, partiendo de las necesidades y limitaciones extraídas de este estudio del estado del arte. Este análisis sirve de nexos con el capítulo tercero en el cual se describirá la solución propuesta y se profundizará en las aportaciones con respecto a las necesidades de mejora identificadas.

2.2.1 NetLab. University of South Australia

El NetLab es el laboratorio remoto desarrollado por la University of South Australia (UniSA). Este laboratorio se encuentra hoy en día disponible en la siguiente dirección: <http://netlab.unisa.edu.au/>, sitio web de la Figura 2.9. Actualmente, el NetLab es usado por los estudiantes y el personal académico de la School of Electrical and Information Engineering (EIE), situado en el edificio Sir Charles Todd del Campus Mawson Lakes, con el propósito de dotar a las lecciones teóricas y prácticas de una nueva herramienta de aprendizaje [NMS+05]. Al ser una herramienta online, el laboratorio remoto NetLab está disponible para los estudiantes fuera del horario prefijado para las clases presenciales en el laboratorio por lo que también puede ser usado por usuarios distintos a los propios estudiantes.

La plataforma NetLab comenzó a desarrollarse como un proyecto de investigación en el año 2002 [MN02]. Desde 2003, los experimentos incluidos en NetLab son usados por los profesores y estudiantes de 3 grados universitarios y desde el año 2007 también está siendo utilizado en aquellos

grados impartidos fuera de la UniSA, como en el Asia Pacific Management Institute (APMI) en Singapur [NM07].

2.2.1.1 Arquitectura del NetLab

El Netlab está desarrollado sobre una arquitectura cliente servidor, la cual puede ser descrita gráficamente tal y como se muestra en la Figura 2.10, en la que aparece el servidor del laboratorio remoto en el centro, rodeado de la representación del hardware real utilizado durante la experimentación, en donde también se incluye una cámara web y los potenciales usuarios situados en cualquier lugar desde el que dispongan de un acceso a Internet [MMN06] [MN08]. Se puede apreciar en la figura los interfaces de control utilizados en el laboratorio, así como parte del sistema de control de los experimentos empleados. En las siguientes secciones los analizamos más en detalle.

2.2.1.2 Aplicación de control en NetLab

El control de los experimentos del NetLab se realiza mediante una aplicación desarrollada en C, la cual utiliza drivers *VXIPlug&play* que reciben las entradas y salidas de los instrumentos a través de librerías VISA. Mediante el código desarrollado en C y compilado en una .dll, los desarrolladores del NetLab han conseguido simplificar las llamadas a las API VISA a través de C, de tal manera que una aplicación desarrollada en Java puede hacer uso fácilmente de estas librerías y controlar un instrumento [MN03].

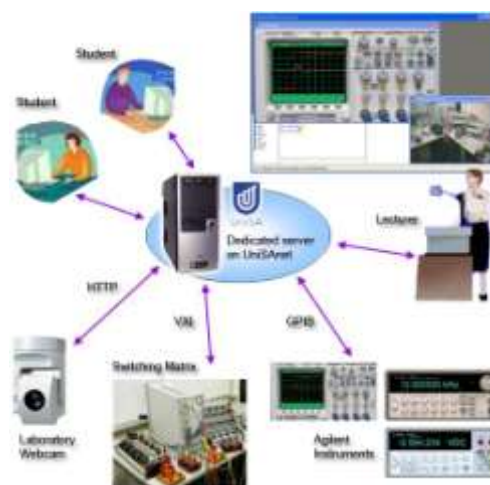


Figura 2.10: Arquitectura del laboratorio remoto NetLab

Por medio del interfaz desarrollado en Java en el cliente, el alumno puede construir virtualmente el circuito que posteriormente se implementará en el laboratorio [MNY04]. Desde la pantalla principal del interface de NetLab, el usuario puede comenzar una sesión del *Circuit Builder*, la cual le permite al usuario crear sus propios circuitos y conexiones entre el experimento y los instrumentos, tal y como se puede advertir en el ejemplo de la Figura 2.11, donde el usuario está comprobando el funcionamiento de un circuito formado por varias resistencias variables conectadas en serie.

Los componentes disponibles para ser utilizados por el usuario, aparecen en la parte derecha del interfaz (Figura 2.11). Estos son los componentes que físicamente se encuentran conectados en la matriz de conmutación, y que también aparecen recogidos en un fichero de configuración que define el administrador del sistema, el cual también recoge cómo están interconectados los componentes entre sí.

Gracias a este interfaz, los componentes pueden ser dispuestos y conectados entre sí de manera sencilla. Las líneas de color representan las conexiones entre componentes o bien entre componentes e instrumentos. Estos circuitos pueden ser guardados por el usuario para un posterior estudio y análisis. La descripción del circuito y sus conexiones se realiza mediante un fichero en formato *Extensible Markup Language* (XML), el cual proporciona una estructura fácil y bien definida para poder usar este fichero en otras posibles aplicaciones.

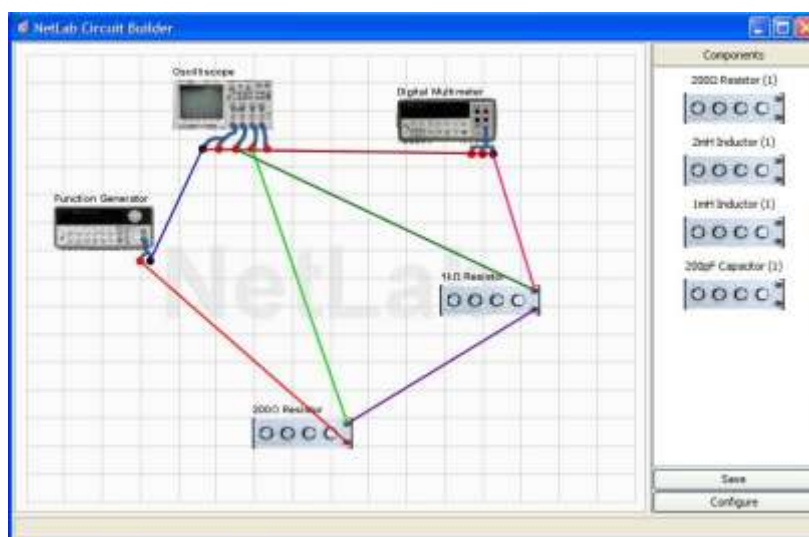


Figura 2.11: Interfaz para el diseño de circuitos

2.2.1.3 Control de experimentos e interfaz de control en NetLab

El principal elemento hardware del sistema de control de los experimentos en el laboratorio NetLab es su matriz de conmutación, la cual hace posible la interconexión de los componentes y de éstos con los instrumentos tal y como se puede apreciar en la Figura 2.12. El equipamiento que compone esta matriz de conmutación está formado por los siguientes instrumentos de laboratorio:

- Agilent E1465A: es una matriz de conmutación formada por relés de dos contactos, en formato 16x16.
- Agilent E8408A: es un mainframe VXI con capacidad de 4 slots. En uno de estos slots se encuentra conectada la matriz E1465A
- Agilent E1406A⁴: módulo de comandos VXI GPIB. Este módulo de comunicaciones permite a este conjunto de instrumentos que forman el sistema de la matriz de conmutación, comunicarse externamente con el servidor de NetLab, a través del bus de comunicaciones GPIB.

En el caso del NetLab, se utilizan dos interfaces de control con los instrumentos: VXI, como estándar de comunicación interno con el módulo de comandos y GPIB para la comunicación con el servidor de NetLab, mediante los comandos SCPI propios de cada instrumento.



Figura 2.12: Configuración de la matriz de conmutación NetLab

⁴ Más información sobre estos equipos en <http://www.home.agilent.com>

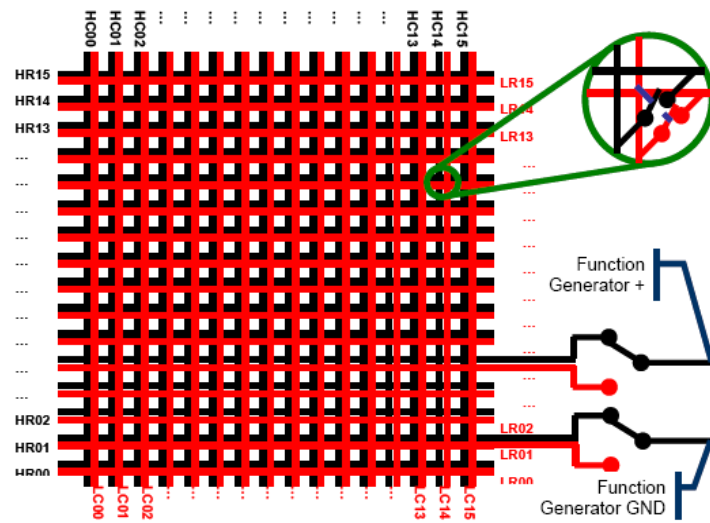


Figura 2.13: Estructura de la matriz de relés y capa de conmutación

El módulo de la matriz de conmutación está compuesto por 256 posibles nodos de interconexión, formados por 16 filas y 16 columnas. La conmutación de estos nodos se realiza mediante relés dobles⁵ (DPDT), los cuales pueden mantener su estado (abierto o cerrado) después del encendido y apagado del equipo.

En la ampliación de la Figura 2.13 se muestra un nodo de conexión y cómo se realiza la conmutación entre las filas y las columnas de la matriz. Esta estructura de relés dobles permite tener dos capas de conmutación separadas: una capa Superior o Higher (denotada como 'H' en la Figura 2.13) y otra capa Inferior o Lower (denominada 'L'). Cada capa puede tener su propio conjunto de componentes conectados en sus terminales, lo que posibilita doblar el número de componentes que pueden ser utilizados remotamente por el usuario.

De no utilizar esta configuración de doble capa ('H' y 'L'), solamente se podrían utilizar como máximo 16 componentes de dos terminales cada uno. Haciendo uso de la configuración de doble capa, el número de componentes se duplica, es decir, el usuario dispone de 32 componentes de dos terminales, pero con la restricción de que no se puede utilizar en un mismo circuito componentes de diferentes capas. Es decir, los componentes

⁵ Se trata de relés *Double Pole Double Throw*, conocidos en español como relés de dos contactos o dobles.

de la capa 'H' no pueden interconectarse con componentes de la capa 'L', ni viceversa.

Pero esta versión de doble capa únicamente estuvo disponible en una de las primeras versiones del NetLab, ya que para su correcto funcionamiento requería el uso de una circuitería externa extra, lo que dificultaba el interconexionado de todos los elementos hardware del laboratorio, haciéndolo más complejo [MN04]. Por esta razón, la versión actual de NetLab cuenta con un osciloscopio de 4 canales, un generador de señales, un multímetro digital, 4 resistencias variables, 2 condensadores variables, 1 inductor variable, 1 transformador y 3 terminales aún sin asignar. El único mecanismo para ampliar el número de componentes disponibles para configurar nuevos experimentos es añadir un nuevo módulo E1465A (o compatible) al mainframe que alberga las matrices de conmutación.

2.2.1.4 Resumen método de control en NetLab

Mediante el método de control desplegado en el laboratorio NetLab, el alumno puede realizar experimentos interactivos en los que puede monitorizar el comportamiento de los mismos a través de los instrumentos dispuestos a tal efecto. Adicionalmente puede modificar el funcionamiento del circuito, ya que ciertos componentes como resistencias o bobinas disponibles en el laboratorio son variables, por lo que el alumno puede alterar la lógica del circuito bien cambiando las conexiones entre dispositivos o bien cambiando sus valores. Como se ha visto anteriormente la configuración de la matriz de conmutación utilizada en este laboratorio solo dispone de 8 componentes y 3 instrumentos por lo que los circuitos que se pueden construir son básicos. A modo de resumen, la Tabla 2.2 muestra las tecnologías utilizadas en el desarrollo del método de control empleado en el laboratorio NetLab.

Aplicación de control	Interfaz de control	Control experimentos	Tipo interacción posible
Desarrollada en C Drivers VXIPlug&play	VXI y GPIB	Matriz de conmutación comercial	Modificación lógica del experimento

Tabla 2.2: Resumen tecnologías empleadas por en el método de control NetLab

2.2.2 RemotElectLab. Polytechnic Institute of Porto

El RemotElectLab es el laboratorio remoto desarrollado en el Instituto Superior de Ingeniería de Oporto (ISEP) del Politécnico de Oporto. Este laboratorio se encuentra integrado dentro de un Sistema de Gestión de Contenidos Moodle, herramienta que hace posible a los estudiantes interactuar con los recursos electrónicos del curso en el cual se encuentren matriculados [MOO09].

Previa petición de usuario y contraseña, se puede acceder al curso en la siguiente URL: <https://moodleproj.isep.ipp.pt/>. Actualmente, el RemotElectLab está siendo usado como complemento en la asignatura Electrónica II en el primer semestre del Grado en Ingeniería Electrónica e Informática.

El principal reto al que se enfrentaron los desarrolladores del RemotElectLab era diseñar un laboratorio que, a pesar de ser accedido de manera remota replicase las funciones y acciones que el usuario puede realizar en un laboratorio real.

Conjuntamente, la plataforma a diseñar debía ser flexible en cuanto a los componentes a utilizar, pudiendo ser exactamente los mismos que el alumno se encontrara en el laboratorio de las instalaciones de la universidad. En términos de funcionalidad, sobre un circuito en la plataforma remota debía ser posible realizar las mismas modificaciones del hardware que el experimento real y llevar a cabo las mismas medidas que el profesor determina durante una clase en un laboratorio real. De esta forma se asegura que los ejercicios que se pueden desarrollar en un laboratorio real puedan ser reproducidos fielmente en la plataforma remota. La única restricción impuesta es que los alumnos o usuarios partirán de una configuración ya realizada previamente, por lo que no podrán implementar el circuito desde cero, sino que únicamente será posible modificar ciertos parámetros del mismo, no teniendo por tanto el alumno el control completo del laboratorio.

2.2.2.1 Arquitectura del RemotElectLab

El laboratorio RemotElectLab también ha sido implementado mediante una arquitectura cliente-servidor, la cual aparece representada en la Figura 2.14.

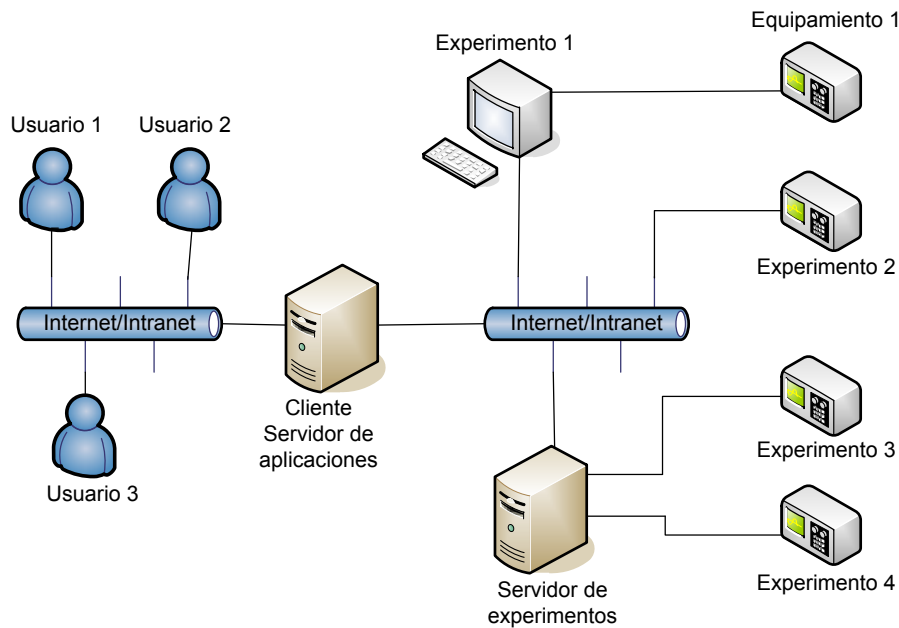


Figura 2.14: Arquitectura software del RemotElectLab

Se puede observar que en este planteamiento existe un cliente que alberga las aplicaciones proporcionadas a los usuarios para acceder e interactuar en el laboratorio, y un servidor de experimentos que se encarga de proporcionar el control sobre los experimentos.

Esta configuración permite proporcionar a los usuarios un número ilimitado de experimentos a través de un único ente público, en este caso el denominado servidor de aplicaciones. No obstante, los investigadores del ISEP están trabajando en las aplicaciones desarrolladas en Moodle y en LabVIEW con el objetivo de que ambas puedan ser accesibles desde el exterior, con el objetivo de que se pueda utilizar el propio servidor Moodle del ISEP como gestor de clientes y controlador de accesos, dejando únicamente al servidor de aplicaciones como encargado de gestionar los experimentos.

2.2.2.2 Aplicación de control en RemotElectLab

Como veremos más adelante, la instrumentación y experimentos del RemotElectLab está basada en el empleo del equipo NI-Elvis, de National Instruments, el cual es únicamente controlable mediante LabVIEW, por lo que la aplicación de control ha sido desarrollada íntegramente utilizando esta tecnología combinada con los drivers específicos de los instrumentos

incluidos en la plataforma NI-Elvis. El entorno de programación de LabVIEW proporciona de manera paralela a la lógica que se está codificando un panel frontal que sirve de interfaz de usuario para la entrada y salida de datos hacia y desde el experimento en este caso. El interfaz obtenido como resultado de la codificación del software de control del RemotElectLab es el que se puede ver en la Figura 2.15.

Recorriendo el interfaz desde la parte superior a la inferior, se puede advertir que dispone de:

- Displays para la visualización de tensiones y corrientes en diferentes puntos del circuito (de 1 a 8 puntos de test posibles)
- Dos módulos de salida gráficos para la visualización de los dos canales del osciloscopio disponible en la plataforma.
- Controles independientes para la configuración de ambos canales del osciloscopio.
- Controles para la configuración del generador de funciones. Se permite la configuración del tipo de onda, amplitud, frecuencia y tensión de offset.

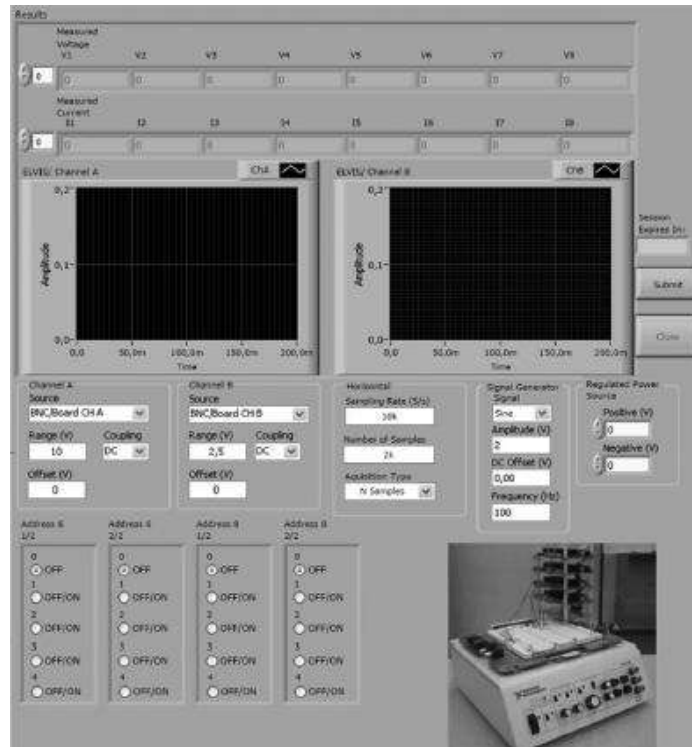


Figura 2.15: Interfaz de usuario del RemotElectLab

- Controles para la fuente de alimentación. Se permite al usuario configurar la tensión positiva o negativa de alimentación del equipo.
- En la parte inferior del interfaz, dispone del panel de control que permite seleccionar la activación o desactivación de los relés contenidos en el módulo de conmutación.
- En la parte derecha del interfaz, el usuario puede visualizar el contador que le indica el tiempo del que dispone para acabar su ejercicio, así como los botones *Submit* para ejecutar su configuración y recibir los resultados, y el botón *Clear* para borrar la configuración y realizar una nueva.

Una vez configurado el experimento y pulsado el botón *Submit*, los datos son enviados al servidor de experimentos, representado en la Figura 2.16, y se espera a que las medidas se realicen y se visualicen en el interfaz. El interfaz de cada estudiante permanece con estos valores hasta que no se envía una nueva configuración a la plataforma NI-Elvis.

En la versión actual del RemotElectLab el usuario remoto no puede guardar la configuración de los equipos ni las lecturas realizadas, siendo el *imprimir pantalla* la única opción disponible para almacenar información sobre su sesión en el laboratorio remoto.

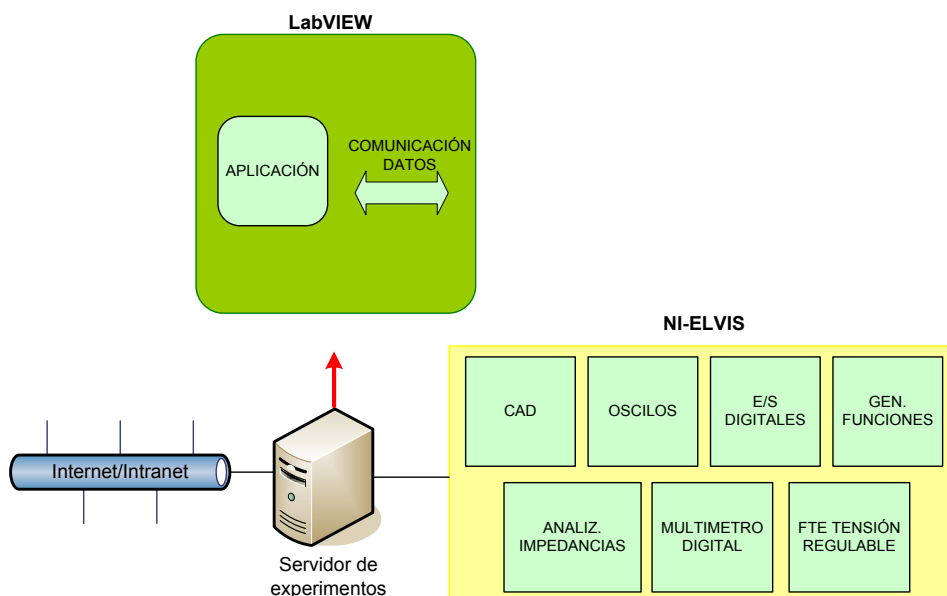


Figura 2.16: Servidor de experimentos del RemotElectLab

Mediante esta configuración las medidas de tensión y corriente sobre el circuito son realizadas de manera automática cada vez que el usuario ejecuta una configuración. Los módulos de conmutación escanean secuencialmente cada uno de los puntos de medida definidos y obtienen los valores de tensión y corriente de cada uno de dichos puntos. Una vez que todas las medidas han sido adquiridas, el software devuelve de una sola vez estos valores al interfaz remoto de usuario.

2.2.2.3 Control de experimentos e interfaz de control en RemotElectLab

Como ya indicábamos anteriormente, la selección de un método de control u otro depende exclusivamente del equipo desarrollador y de las necesidades y requisitos que ellos impongan a su método y al laboratorio remoto en su totalidad. En el caso del RemotElectLab, el equipo de trabajo asumió desde un principio que los alumnos no tienen por qué repetir una y otra vez los pasos desde cero para construir el circuito, por lo que eliminaron esta necesidad de la plataforma remota. Es decir, asumieron que la implementación del circuito se realiza en las clases prácticas del laboratorio real, por lo que no es necesario repetir este proceso en la práctica remota.

Como resultado de estas premisas se desarrolló el RemotElectLab teniendo en consideración las siguientes especificaciones [SAG10]:

- La implementación de un circuito en la plataforma RemotElectLab no debe requerir ninguno recurso extra, aparte de aquellos que se encuentran comúnmente en el laboratorio real.
- El RemotElectLab debe ser lo suficientemente flexible para permitir a los alumnos implementar todos los circuitos propuestos durante las clases prácticas, por lo que es obligatorio que el laboratorio sea reutilizable desde el punto de vista de los circuitos y componentes.
- Las medidas de tensión y corriente y la visualización de las formas de onda que se pueden implementar sobre los nodos del circuito a testear deben producirse sin introducir ningún error significativo en estas medidas, por lo que éstas deben ser prácticamente las mismas que las adquiridas en las clases que tienen lugar en el laboratorio real.
- La posibilidad de sustitución remota de componentes y de parámetros del circuito debe ser la misma que la que existe en el laboratorio real,

donde pequeñas modificaciones del circuito son posibles para comprobar su comportamiento.

- El tiempo que un profesor o un técnico necesitan para configurar un circuito debe ser similar al necesario para crear el mismo circuito en una placa de prototipos real.
- No es necesario que el profesor o técnico encargado de disponer nuevos circuitos en la plataforma disponga de conocimientos especiales, aunque sí es recomendable que tengan ciertas nociones de programación en LabVIEW.
- El interfaz utilizado por los usuarios remotos debe ser genérico y no se deberían requerir modificaciones cuando un nuevo circuito sea añadido a la plataforma.
- El laboratorio remoto debe estar integrado en el mismo LMS usado en la universidad, por lo que de esta forma los alumnos tienen todos los recursos de aprendizaje integrados en una misma plataforma.
- El coste del laboratorio remoto no debe exceder el coste del laboratorio real.
- El laboratorio remoto obtenido debe ser fácilmente replicable en cualquier lugar.

Como se puede leer en las especificaciones anteriores, uno de los objetivos del RemotElectLab es reducir el coste de desarrollo del laboratorio remoto. Algunas veces, la necesidad de incorporar a la arquitectura hardware del laboratorio equipos que dispongan interfaces que posibiliten su control remoto (Ethernet, GPIB, etc.), incrementan el coste del laboratorio. Por otro lado, para interactuar con estos equipos muchas veces es necesario instalar drivers específicos en el servidor y desarrollar algoritmos de comunicación dedicados a la comunicación con los mismos. Además, el número de equipos necesarios generalmente es alto, componiéndose como mínimo de una fuente de alimentación con varias salidas regulables, un multímetro, osciloscopio con al menos dos canales y un generador de funciones.

La solución adoptada por el equipo del RemotElectLab para solventar estas restricciones ha sido el uso de una solución integrada, el equipo NI-Elvis (National InstrumentsTM Educational Laboratory Virtual Instrumentation Suite), un entorno de diseño y prototipado basado en instrumentación virtual [NIE09], que puede ser usado tanto en las clases prácticas de un laboratorio real, como servir de plataforma para el desarrollo

de un laboratorio remoto. Desde el punto de vista docente, esta solución está en consonancia con la idea de que un ordenador con un experimento online es un complemento a la actividad experimental junto con un ordenador local con un experimento integrado [Soy00].

El uso de la estación de trabajo NI-Elvis está muy extendido y es utilizada tanto en el ámbito de la docencia como a nivel industrial debido a su versatilidad, facilidad de funcionamiento y entorno de trabajo y desarrollado integrado [BI08] [SN05] [SHH+06]. Además, al incorporar de manera integral en el mismo paquete, el hardware y el software de control LabVIEW, proporciona numerosas librerías y aplicaciones ya prediseñadas, que facilitan un rápido desarrollo de nuevas aplicaciones. LabVIEW cuenta con una amplia comunidad de desarrolladores en donde se pueden encontrar soluciones a muchos problemas y funciones ya desplegadas, lo que hace que el desarrollador se encuentre arropado a la hora de realizar una aplicación de control.

NI-Elvis utiliza el concepto de instrumentación virtual para proveer de los equipos de laboratorio al usuario. Un instrumento virtual es la combinación de un software de control ejecutado en un PC, a través del que se interactúa con un hardware, de tal manera que se permite a los usuarios realizar las mismas acciones que si estuvieran manejando un instrumento convencional [MVM+99]. NI-Elvis emplea el entorno LabVIEW como software de control y el hardware de adquisición de datos de National Instruments para crear los instrumentos virtuales de la plataforma. En este caso, el hardware de adquisición de datos es el encargado de capturar, medir y analizar fenómenos físicos del mundo real. De este modo, los paneles de control virtuales de los instrumentos desarrollados en LabVIEW [NIL10], junto con los sistemas de adquisición de datos posibilitan que el usuario del RemotElectLab tenga a su disposición hasta doce instrumentos diferentes [NAT06]: generador de ondas aleatorio, analizador Bode, lectura/escritura en bus digital, multímetro digital, analizador dinámico de señal, generador de funciones, analizador de impedancias, osciloscopio, analizador voltaje/corriente de dos hilos, analizador voltaje/corriente de tres hilos y fuente de alimentación variable.

El control de la plataforma NI-Elvis reside en el ordenador al cual se encuentra conectado de manera local mediante una conexión USB de alta

velocidad, de modo que es relativamente sencillo disponer de todas las funcionalidades de la plataforma de manera remota a través de Internet, con la ventaja de seguir manteniendo intactas todas las funcionalidades en modo local.

Además de los instrumentos enumerados anteriormente, el laboratorio RemotElectLab dispone de una placa de desarrollos para el ensamblaje de circuitos electrónicos en la parte superior, tal y como se puede apreciar en la Figura 2.17. Esta placa de prototipos, combinada con la matriz de conmutación basada en relés electro-mecánicos y desarrollada expresamente para este laboratorio, hace posible que el alumno realice medidas, intercambie componentes y reconfigure el circuito bajo prueba, según las especificaciones de la práctica que esté realizando en cada momento. Así pues, los módulos que forman esta matriz de conmutación son los encargados de proporcionar flexibilidad y escalabilidad al sistema.

Como en el laboratorio real del ISEP, en el RemotElectLab únicamente se dispone de un voltímetro y un amperímetro para realizar medidas sobre los experimentos de electrónica. Por tanto, si mediante esta plataforma se quiere reproducir las mismas acciones que en el laboratorio real, los módulos de la matriz de conmutación que acompañan al NI-Elvis deben ser capaces de proporcionar un método para medir tensiones y corrientes con un único instrumento, y en diferentes nodos del circuito bajo prueba disponible en el laboratorio, el cual aparece representado en la Figura 2.18.

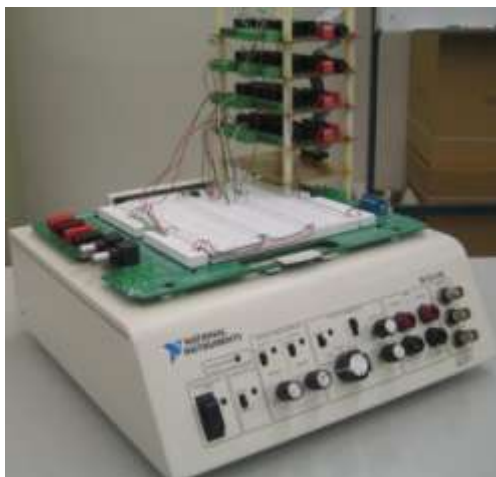


Figura 2.17: Equipos y componentes del RemotElectLab

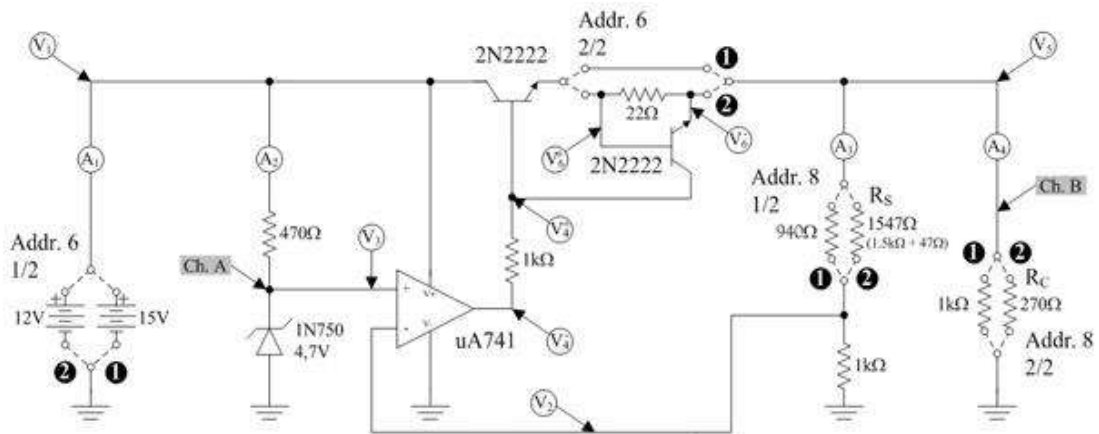


Figura 2.18: Circuito bajo prueba del RemotElectLab

Vemos en este ejemplo de la Figura 2.18 como los relés dispuestos en los módulos de la matriz de conmutación pueden ser usados para intercambiar componentes de diferentes valores, en las distintas ramas del circuito bajo prueba. Además, estos módulos deben cumplir con dos especificaciones importantes:

- La solución diseñada no debe estar limitada en cuanto al número de medidas de tensiones y corrientes a realizar sobre el circuito, ya que si no, habría acciones que no se podrían ejecutar de manera remota.
- No se deben introducir no-linealidades (errores) en el circuito, ya que se interferiría el normal funcionamiento del esquema.

Ante estas especificaciones, la posibilidad de utilizar interruptores basados en transistores no es viable, debido a su resistencia y capacidad intrínseca, por lo que la solución adoptada es realizar el diseño en el que se utilizan relés electromecánicos del tipo *Double Pole Double Throw* (DPDT). Estos relés constan de dos contactos de entrada independientes, cada uno de ellos con una salida abierta y la otra cerrada. De este modo se obtiene una configuración similar a un multiplexor doble, de manera que se pueden usar estos módulos de conmutación para medir voltajes en modo diferencial y corrientes en las diferentes ramas del circuito.

Cada uno de los módulos que forman esta matriz de conmutación propia del RemotElectLab es accedida a través de un bus de direcciones de 4 bits y un bus de datos de 8 bits. Estos buses proporcionados por el equipo NI-Elvis son usados para seleccionar el módulo a utilizar y enviarle la

configuración oportuna, en función de las acciones a implementar sobre el circuito bajo prueba. Mediante este sistema, el número máximo de módulos a utilizar son 12. El pequeño microcontrolador instalado en cada uno de los módulos de la matriz de conmutación es el encargado de recibir la configuración seleccionada por el usuario, interpretarla y abrir o cerrar los relés correspondientes.

Cuando se quiere añadir un nuevo circuito a la plataforma RemotElectLab, el profesor selecciona los 8 nodos y las 8 ramas en las cuales quiere obtener medidas de tensión y corriente, y conecta estos puntos al voltímetro y amperímetro del NI-Elvis a través de los módulos de conmutación. Al mismo tiempo también selecciona los componentes o partes del circuito que el estudiante podrá reconfigurar durante el ejercicio práctico del laboratorio.

En el caso del ejemplo de la Figura 2.18, el objetivo de la práctica es que el alumno comprenda el funcionamiento de un regulador de tensión lineal. Sobre el esquema, el estudiante puede cambiar la tensión de alimentación, la resistencia de carga, el divisor de tensión o incluso el limitador de corriente. Al mismo tiempo, puede comprobar las implicaciones de estos cambios sobre el circuito, tomando las pertinentes medidas de tensión y corriente en los nodos oportunos.

Los cambios en la configuración y los resultados de las medidas se hacen a través del interfaz que se puede observar en la Figura 2.15. Por ejemplo, para cambiar la tensión de alimentación de 15V a 12V, y atendiendo a las etiquetas de la Figura 2.18, el estudiante tiene que activar la opción 2 del módulo de conmutación referenciado por la dirección 6-1/2. Los cambios en las medidas de tensión y corriente ante este cambio, se muestran en la parte superior del interfaz.

El error introducido en las medidas se ha reducido al mínimo debido a la reducida resistencia y capacitancia intrínseca de los relés electromecánicos escogidos. De este modo se consigue que las medidas realizadas a través del laboratorio remoto coincidan con aquellas realizadas en el laboratorio real, permitiendo a los estudiantes verificar o completar los ejercicios realizados en los laboratorios de la universidad. Estos módulos permiten también aplicar diferentes tensiones y señales al circuito, intercambiar componentes, o incluso reconfigurar ciertas partes del circuito bajo test.

Aplicación de control	Interfaz de control	Control experimentos	Tipo interacción posible
Desarrollada en LabVIEW Drivers propios NI-Elvis	USB alta velocidad	Matriz de conmutación propia	Modificación ciertos parámetros del experimento

Tabla 2.3: Resumen tecnologías empleadas por en el método de control RemotElectLab

2.2.2.4 Resumen método de control en RemotElectLab

Actualmente el único experimento disponible en la plataforma RemotElectLab es el representado en la Figura 2.18, el cual requiere que el alumno tenga conocimientos avanzados de electrónica analógica, ya que conjuga diversos componentes como transistores o amplificadores operacionales que no son estudiados por el alumno hasta haber adquirido los conocimientos básicos de electrónica analógica. Por tanto el experimento contemplado en este laboratorio es un experimento de cierta dificultad y que también posibilita que el alumno modifique los parámetros del circuito y obtenga sus propias conclusiones en base a los resultados obtenidos.

La instrumentación disponible en el laboratorio es sencilla y limitada, permitiendo únicamente al alumno medir tensiones y corrientes en las ramas del circuito, sin poder modificar la configuración de los instrumentos disponibles a tal efecto.

A modo de resumen, la Tabla 2.3 muestra las tecnologías utilizadas en el desarrollo del método de control empleado en el laboratorio RemotElectLab.

2.2.3 ISILab. Universidad de Génova

El ISILab (Internet Shared Instrumentation Laboratory) es el laboratorio web desarrollado por la Universidad de Génova, disponible en el en sitio web <http://isilab.dibe.unige.it/> y que es usado por los alumnos de ingeniería como soporte a asignaturas relacionadas con la electrónica, tanto analógica como digital [BCP+00]. El laboratorio está diseñado para permitir a estos estudiantes practicar con los instrumentos del laboratorio y las metodologías de medida aprendidas en las clases teóricas, ejecutando de manera remota experimentos reales.



Figura 2.19: Website del laboratorio remoto ISILab

Estos experimentos están relacionados con la obtención de medidas básicas en circuitos analógicos, como puede ser el retraso de los circuitos digitales o la ganancia y distorsión en amplificadores.

El acceso al laboratorio remoto ISILab se realiza a través del portal web de la Figura 2.19. Éste es el punto de acceso a través del cual se puede actuar sobre los diferentes experimentos disponibles, los cuales pueden estar distribuidos geográficamente y ser accedidos directamente por los usuarios, sin ningún impedimento respecto a su localización geográfica y la del experimento, tal y como ocurre en otras plataformas de educación remota, como en el caso del diseñado por Arpaia [ABC⁺00]. El portal ofrece además un índice con todos los experimentos que se encuentran activos y se encarga tanto de las políticas de seguridad de los accesos como de iniciar la comunicación entre los usuarios y los servidores de experimentos.

2.2.3.1 Arquitectura del ISILab

La arquitectura del ISILab también ha sido desarrollada mediante el paradigma cliente-servidor, pero en este caso se ha creado lo que el equipo de desarrollo del ISILab denomina Laboratorio Virtual Compartido [BCS⁺03], el cual representa un conjunto de laboratorios reales, cada uno de ellos con unos experimentos definidos, que pueden ser accedidos desde un mismo sitio web.

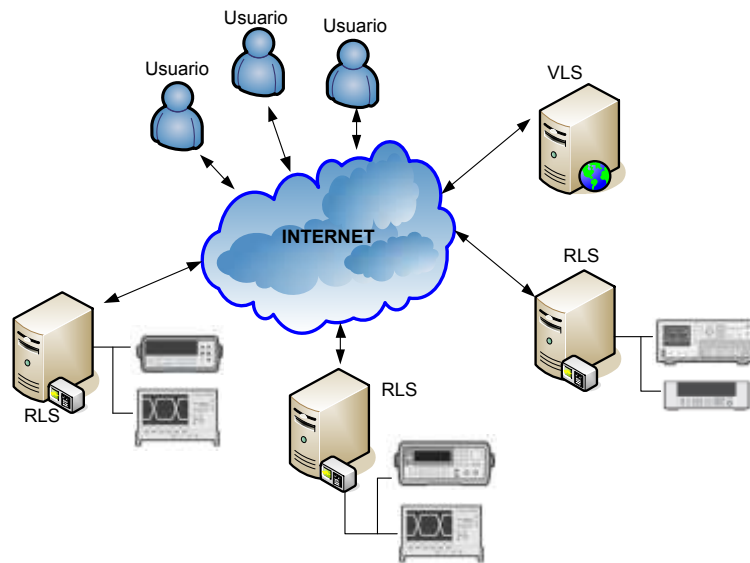


Figura 2.20: Arquitectura del Laboratorio Virtual Compartido

El Laboratorio Virtual Compartido desarrollado en el ISILab está basado en un entorno software distribuido (Figura 2.20), consistente en un servidor de Laboratorio Virtual Principal (VLS), uno o varios Servidores de Laboratorio Reales (RLS) y los clientes a través de los cuales se conectan los usuarios. Los servidores de laboratorio reales se encuentran distribuidos geográficamente y controlan los experimentos.

Cada uno de las entidades que forman parte de esta arquitectura han sido desarrollados mediante diferentes tecnologías, las cuales aparecen reflejadas en la Figura 2.21, junto con los protocolos de comunicación utilizados para conectar los clientes, los RLS y los VLS [BCS05a].

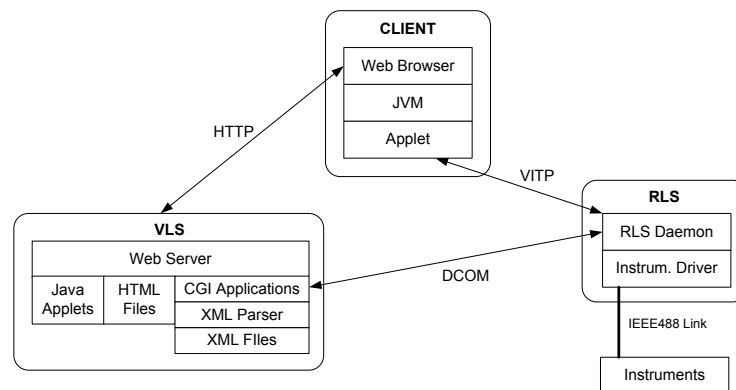


Figura 2.21: Arquitectura Software del ISILab

2.2.3.2 Aplicación de control e interfaz de control en ISILab

Como se puede apreciar en la arquitectura descrita anteriormente, es el servidor denominado RLS (Servidor de Laboratorio Real) el que está en contacto directo con los experimentos, y por tanto en donde reside la aplicación software encargada de su control.

El diseño tanto de este servidor como del servidor VLS se ha realizado siguiendo un método basado en la descripción de los experimentos mediante ficheros XML. Estos ficheros XML contienen toda la información referente a cada uno de los experimentos que contiene el laboratorio, de modo que cada vez que se quiera añadir un experimento al laboratorio se debe describir el equipamiento hardware que utiliza sus instrumentos, así como los interfaces de control empleados entre el servidor y el experimento. Así se consigue catalogar los experimentos como recursos de aprendizaje y se facilita la gestión del laboratorio remoto [BCS05b].

Estos ficheros contienen también la dirección IP y el puerto del RLS en el cual se ejecuta el software que gestiona el experimento, así como los parámetros de control específicos de cada uno de los elementos (circuito bajo prueba e instrumentos) del experimento [BSV+03]. La Figura 2.22 muestra el esquema de la estructura de datos que describe un experimento contenido en la plataforma ISILab.

En esta arquitectura de servidores virtuales definida en el ISILab, las tareas de control de acceso al laboratorio y las de gestión de recursos del mismo recaen exclusivamente en el VLS, mientras que el Servidor de Laboratorio Real es el encargado de ejecutar la aplicación desarrollada íntegramente en Java y que controla el experimento a través de las acciones realizadas por el usuario en el interfaz web, haciendo llegar estas órdenes al experimento mediante el bus GPIB o RS-232, dependiendo de la configuración del experimento.

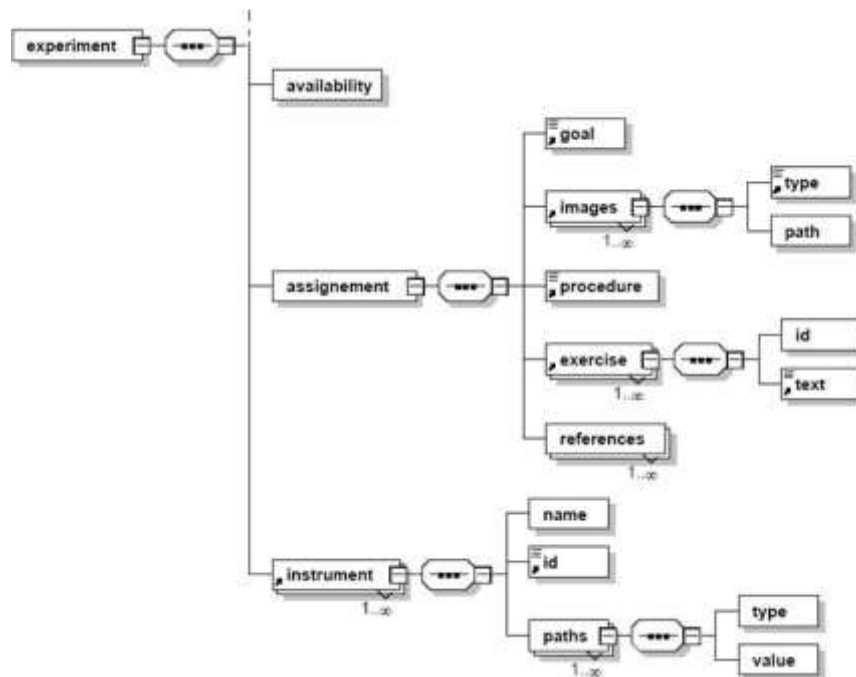


Figura 2.22: Esquema XML de un fichero de configuración del ISILab

Esta aplicación de control se ha desarrollado de modo que los instrumentos son recursos abstractos que ofrecen APIs (Application Program Interface) homogéneas al motor del RLS, es decir, una capa software independiente del instrumento encargada de la comunicación con el cliente remoto. Para añadir un nuevo instrumento únicamente es necesario a) encapsular el driver del instrumento en el adaptador adecuado, el cual se encarga de mostrar las funcionalidades del instrumento, y b) modificar algunos ficheros de configuración [BS09]. De este modo se consigue alta capacidad de reconfiguración, ya que ISILab permite introducir nuevos instrumentos o experimentos sin necesidad de realizar ninguna ampliación en el código del código fuente

El adaptador del driver es una pequeña capa de software encargada de interceptar las llamadas al instrumento desde el motor del RLS, traducirlas y reenviarlas en el formato adecuado al driver nativo del instrumento. Es decir, proporciona el método de comunicación y entendimiento entre el interfaz de usuario y el instrumento físico, además de poder implementar funciones extras, como por ejemplo, realizar la FFT (Fast Fourier Transform) de los datos proporcionados por el driver de un osciloscopio, y poder usarlo por tanto como un analizador de espectros.



Figura 2.23: Placa base ISIBoard del laboratorio ISILab

Los encapsulados de drivers que se han realizado son aquellos que abarcan los instrumentos más comunes de los laboratorios y con los que actualmente cuenta el ISILab: osciloscopio, generador de funciones y multímetro digital. Están basados en la tecnología IVI y pueden ser por tanto usados por instrumentos de distintos vendedores, pertenecientes a la misma clase de dispositivos.

2.2.3.3 Control de experimentos en ISILab

En ISILab se ha seguido una metodología diferente a las vistas anteriormente para poder proporcionar a los usuarios los experimentos disponibles en el laboratorio. Se ha creado un sistema modular denominado ISIBoard consistente en una placa base con 17 slots, en los cuales se pueden insertar las tarjetas que contienen los experimentos a testear (Figura 2.23).

Cada una de las tarjetas que se pueden insertar en estos slots, comprende un área de unos 45cm² y posee 18 líneas de conexiones, disponibles tanto para las líneas de alimentación de los circuitos bajo prueba, como la conexión de instrumentos al circuito. En concreto estas líneas son:

- Líneas dedicadas a la alimentación (+12V, -12V, GND de alimentación, +5V y señal de GND)
- Una línea como señal de entrada
- Tres líneas como señales de salida
- Ocho líneas para identificar el circuito.

Una vez que la tarjeta que contiene el experimento es insertada en uno de los slots de la placa base, las conexiones con los instrumentos son gestionadas

dinámicamente por un conjunto de interruptores controlados por el RLS, que a su vez se encarga de controlar los instrumentos a través de interfaces GPIB, y/o RS-232.

De acuerdo con esta aproximación los profesores pueden crear circuitos en función de sus objetivos educativos sin un número límite de componentes por circuito. Una vez diseñado el circuito, se implementa en una de las tarjetas y se inserta en un slot de la placa base. Por consiguiente se elimina la necesidad de construir complejas matrices de conmutación para la interconexión de componentes. Además, un número reducido de relés hace que la calidad de las señales tomadas sea mejor que si éstas tuvieran que cruzar por múltiples interruptores.

Es decir, en ISILab no se dispone para el control de cada uno de los experimentos de una matriz de conmutación, sino que se ha desarrollado un sistema de control de los experimentos del laboratorio, que permite al usuario seleccionar uno de entre un conjunto de experimentos ya implementados. Mediante la configuración y definición del fichero XML de cada uno de estos experimentos, cuando el usuario accede al laboratorio, se le muestran todos los recursos relativos a dicho experimento, sin poder realizar ninguna modificación adicional.

2.2.3.4 Resumen método de control en ISILab

Como se ha descrito en la sección anterior, los experimentos a los cuales se pueden acceder a través de la plataforma del ISILab son circuitos previamente configurados por el profesor de la asignatura. Es decir, no se permite al usuario remoto configurar un circuito partiendo de los componentes discretos tal y como se puede hacer en otros laboratorios [GZH⁺07] [MMN06] [ATF⁺05].

Según la clasificación realizada en el capítulo primero, los experimentos incluidos en ISILab son de lectura o de instrumentación, ya que no se permite cambiar la lógica del circuito bajo prueba incluido en el experimento. Cada una de las tarjetas que contiene el experimento también dispone de un número fijo de líneas de control, por lo que los puntos de medida en cada circuito o las modificaciones habilitadas por el profesor están limitadas. Por lo tanto, la complejidad de los circuitos incluidos en los experimentos dependerá del profesor que los diseñe, de modo que el

laboratorio ISILab puede ser usado con estudiantes principiantes o avanzados.

A modo de resumen, la Tabla 2.4 muestra las tecnologías utilizadas en el desarrollo del método de control empleado en el laboratorio ISILab

2.2.4 Proyecto VISIR. Blekinge Institute of Technology

El sitio web OpenLabs (<http://openlabs.bth.se/>) es la plataforma educativa destinada a albergar los laboratorios remotos desarrollados por el departamento de Procesado de señal del Blekinge Institute of Technology (BTH) en colaboración con National Instruments-USA y la empresa Axiom EduTech de Suecia. Desde esta web (Figura 2.24) se puede acceder a varios laboratorios remotos, cada uno de ellos orientado a una temática diferente y usado en su propia asignatura [Gus06]:

- Laboratorio de antenas: los estudiantes no necesitan hacer ninguna modificación en el experimento, ya que en este laboratorio, únicamente observan el comportamiento de las antenas. El objetivo de este laboratorio es comprender el funcionamiento de las antenas y aprender a analizar el diagrama de radiación de la antena bajo prueba [NZP08].
- Laboratorio de seguridad de la información: el laboratorio remoto de seguridad, ofrece un entorno de experimentación a distancia en materia de seguridad. Con la creación de una red de control de máquinas y aislándolo de otras redes, permite experimentar con software potencialmente dañino, sin exponer a ninguna otra máquina a los problemas que este software malicioso pudiera crear, fuera de la red experimento [ZS08].

Aplicación de control	Interfaz de control	Control experimentos	Tipo interacción posible
Desarrollada en Java Drivers IVI	GPIB – RS-232	Matriz de conmutación propia	Observación solo E/S experimento

Tabla 2.4: Resumen tecnologías empleadas por en el método de control ISILab

OpenLabs

START
 + Antenna theory
 + Electronics
 + Security
 + Signal processing

RESULTS
 + 2008
 + 2009
 + 2010

OpenLabs Electronics Laboratory

Go to the electronics laboratory web page

Open your electronics laboratories for remote experimentation

The Internet provides new possibilities for universities and other teaching organizations to share laboratories equipped with expensive instruments. Blekinge Institute of Technology (BTH) in Sweden has created a lab workbench equipped with a unique remote control interface, enabling students to perform physical experiments at home or elsewhere. For example, they can practise before regular lab sessions. Remote experimentation supplements face-to-face sessions. Off campus students can participate in supervised sessions using teleconferencing tools.

Invitation

A project, VISIR (Virtual Instrument Systems in Reality), has been started to disseminate methods to open laboratories for remote experimentation. This is an invitation to:

- join the VISIR Community
- set up a VISIR workbench for electrical experiments
- participate in the further development of this VISIR Lab Platform

It is an opportunity of providing students with free experimentation resources 24/7 without substantial increase in cost per student. Spending more time on experiments your students will become true experimenters being able to design goods and services complying with the requirements of a sustainable society. FH Campus Vienna in Austria and University of Deusto in Spain have set up VISIR workbenches and use them in their own lab courses. The VISIR software is released under a GNU GPL license. The instruments and LabVIEW, which is also required, can be bought from National Instruments. The telemanipulator needed for circuit wiring is commercially available as well. Please contact [ingvar.gustavsson](mailto:ingvar.gustavsson@bth.se) for more information.

Figura 2.24: Página Web principal del proyecto VISIR

- Laboratorio de procesamiento de señal: mediante este laboratorio remoto, los estudiantes pueden desde un PC conectado a Internet, acceder a experimentos basados en el análisis de vibraciones. Para ello pueden controlar remotamente un analizador dinámico de señal, un vibrador electro-dinámico y un amplificador de vibraciones con el objetivo de monitorizar cómo una estructura mecánica se comporta cuando se le somete a una serie de vibraciones [AHG+06].
- Laboratorio de electrónica analógica: es el laboratorio resultado del proyecto VISIR (Virtual Instrument Systems in Reality) y que permite a los usuarios construir físicamente un circuito analógico y testarlo con los equipos propios de un laboratorio de electrónica [Gus03b].

De todos ellos, en el análisis de la metodología de control implementada en el laboratorio VISIR únicamente se estudiará en profundidad el laboratorio remoto de electrónica analógica, ya que es la plataforma tecnológica que ha servido como punto de partida para desarrollar el método defendido en este trabajo de investigación.

El proyecto VISIR comenzó a finales del año 2006 con el objetivo de crear una red de laboratorios remotos, todos ellos basados en esta

plataforma, denominada VISIR Open Laboratory Platform [GZH⁺07]. Este proyecto ofrece a aquellas universidades o centros educativos que se quieran sumar a la iniciativa, una distribución software lanzada bajo licencia GNU GPL, que contiene todas las herramientas y documentación necesarias para poner en marcha un laboratorio remoto [ZS08] [ZGH07].

Por tanto el objetivo del proyecto VISIR es establecer una Comunidad VISIR de organizaciones/universidades colaboradoras que contribuyan al desarrollo y promoción de esta plataforma y manera que se posibilite el intercambio de recursos y experimentos en las instituciones involucradas.

El grupo de investigación WebLab-Deusto (www.weblab.deusto.es), lleva desde el año 2007 participando en el desarrollo y difusión del proyecto VISIR <https://weblab-visir.deusto.es/electronics/> [GNZ⁺09]. Fruto de esta colaboración y trabajo desarrollado en el proyecto VISIR surgieron las motivaciones que dieron lugar al método de control que proponemos en este trabajo de investigación y que detallaremos en el capítulo siguiente.

2.2.4.1 Arquitectura del VISIR

La plataforma VISIR también ha sido desarrollada mediante el paradigma cliente-servidor, el cual aparece representado en la Figura 2.25. En esta arquitectura se pueden ver los tres servidores utilizados para proporcionar al usuario conectividad y control sobre los experimentos incluidos en los laboratorios del BTH.

El servidor web (que desempeña el papel de cliente) se encarga a través de un interfaz de proporcionar las herramientas administrativas para el registro y acceso de usuario, así como de la reserva de horas del laboratorio. A través del cliente web los usuarios construyen y testean sus experimentos utilizando paneles frontales de los instrumentos del laboratorio. Por otro lado, el Servidor de Medidas (*Measurement Server*) es el encargado de gestionar las peticiones realizadas por el cliente de experimentos. Finalmente, el Servidor de Equipos (*Equipment Server*) se encarga de ejecutar los comandos del usuario sobre los instrumentos del laboratorio y obtener sus respuestas [Gus04].

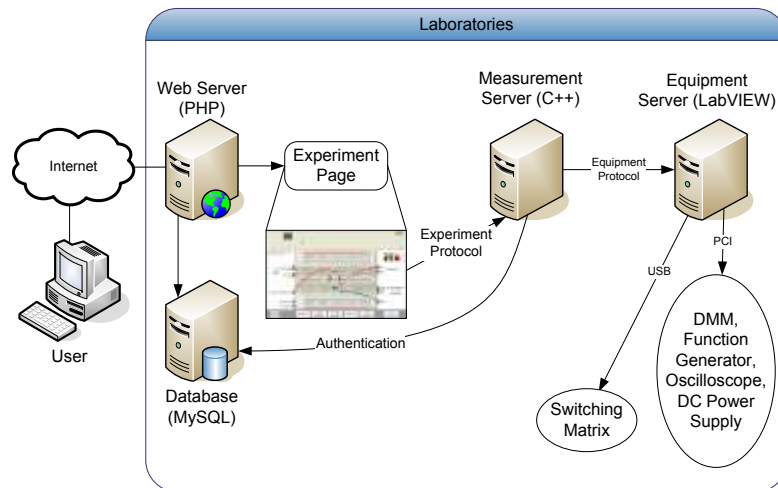


Figura 2.25: Solución software de la plataforma VISIR

2.2.4.2 Aplicación de control en VISIR

En esta arquitectura, las funciones desempeñadas por el método de control diseñado en el laboratorio VISIR recaen principalmente en el Servidor de Equipos, si bien el Servidor de Medidas también desempeña, entre otras (gestión de conexiones y usuarios), funciones que facilitarán el control de los experimentos. Entre otras tareas, el Servidor de Medidas se encarga de validar las peticiones enviadas desde el cliente antes de ejecutarse sobre el experimento. Así habrá solicitudes por parte del cliente que no llegarán a ejecutarse sobre el experimento, como por ejemplo la opción de crear un cortocircuito. El Servidor de Medidas también comprueba que se pueda llevar a cabo la implementación del circuito que se quiere construir sobre la matriz de conmutación. Este chequeo se realiza tanto respecto a los componentes que se quieren utilizar como a las conexiones entre dichos componentes. Para ello, el profesor de la asignatura describe en varios ficheros de configuración tanto los componentes que se encuentran disponibles en la matriz, como las conexiones físicas existentes, y por tanto posibles entre ellos.

Pero es el Servidor de Equipos quien ejecuta las aplicaciones principales de control contempladas en el método de control desarrollado para el laboratorio VISIR. Este servidor es el encargado de gestionar, controlar y albergar los experimentos (circuitos bajo prueba e instrumentos) disponibles en la plataforma VISIR, incluyendo una matriz de conmutación propietaria desarrollada para dar soporte al laboratorio de electrónica

analógica. Para ello se encarga de traducir las peticiones que recibe del servidor de medidas a través de una conexión TCP/IP y de ejecutarlas sobre los instrumentos mediante una serie de algoritmos desarrollados íntegramente en LabVIEW, los cuales hacen uso de los drivers VXIPlug&play propios de los instrumentos integrados en el PXI.

En el caso de la plataforma VISIR, los instrumentos que forman parte de los experimentos del laboratorio, a excepción de la matriz de conmutación, están integrados en un chasis PXI el cual se encuentra conectado al PC en el que reside y se ejecuta tanto Servidor de Equipos como el Servidor de Medidas, tal y como se puede observar en la Figura 2.26.

2.2.4.3 Control de experimentos e interfaz de control en VISIR

Junto a la matriz de conmutación que posteriormente analizaremos, los instrumentos que el usuario dispone para comprobar el funcionamiento del circuito bajo prueba y realizar medidas sobre él se encuentran integrados en un chasis PXI de National Instruments (Figura 2.26). Según el capítulo anterior, el estándar PXI permite disponer en uno de los módulos de un controlador integrado. En el caso de la plataforma VISIR desplegada en el BTH, el módulo PXI no dispone de este controlador embebido, por lo que se requiere de un ordenador anexo que realice las funciones del Servidor de Equipos [Gus02].



Figura 2.26: Instrumentos y servidor de equipos de la plataforma VISIR.

En cambio, en la plataforma VISIR desplegada en la Universidad de Deusto se dispone del chasis PXI-1042Q, el cual dispone de espacio para hasta 8 módulos de instrumentación y dispone además del controlador integrado. Este controlador se comporta como un computador convencional, solo que se encuentra adaptado a las especificaciones físicas impuestas por el estándar PXI. Es en este ordenador embebido en donde se ejecuta el algoritmo de control del Servidor de Equipos.

Además de controlador, el servidor de equipos de la plataforma VISIR está preparado para gestionar los siguientes instrumentos, los cuales se encuentran a disposición de los usuarios para su control remoto:

- Fuente de alimentación triple, NI PXI-4110
- Generador de funciones, NI PXI-5402
- Multímetro digital, NI PXI-4070
- Osciloscopio digital, NI PXI-5112

En función de la versión del chasis PXI seleccionado, el controlador puede estar dotado de un interfaz GPIB, por lo que el Servidor de Equipos también podría adaptarse para controlar equipos mediante este sistema de control.

Por otra parte, siendo la construcción íntegra del circuito bajo prueba y su posterior testeo uno de los requisitos de partida del equipo de trabajo del laboratorio VISIR, la matriz de conmutación encargada de crear ese circuito de manera remota y de hacer posible que el alumno tome medidas sobre él, toma especial relevancia en el diseño del sistema de control empleado en este laboratorio.

El método utilizado para diseñar e implementar la matriz de conmutación en la plataforma VISIR consiste en una matriz propietaria basada en una pila de tarjetas diseñadas bajo el estándar internacional de sistemas embebidos PC/104⁶. Los relés que hacen posible la conexión entre los componentes y entre éstos y los instrumentos se encuentran dispuestos en estas tarjetas, formando una plataforma tridimensional, junto con los conectores y zócalos destinados a albergar los componentes disponibles para crear los circuitos reales de manera remota. Esta disposición tridimensional tiene como objetivo limitar la longitud de los cables necesarios en el cableado, y aumentar por tanto el ancho de banda de los circuitos.

⁶ Referencias al estándar PC/104 pueden ser consultadas en www.pc104.org

En la estructura de la matriz de conmutación se han definido tres tipos de tarjetas: una para la conexión de fuentes, otra para la conexión de otro tipo de instrumentos y un tercer tipo para la conexión de los componentes que formarán los circuitos electrónicos diseñados por el usuario. La configuración básica de la matriz de conmutación está formada por una tarjeta para la conexión de fuentes, dos tarjetas para la conexión de instrumentos y un número máximo de veinticinco tarjetas para componentes. El número de tarjetas de componentes dependerá de la cantidad de componentes que se quieren poner a disposición del usuario: a más componentes, más tarjetas de este tipo se hacen necesarias [GZH07].

Las tarjetas para la conexión de fuentes hacen referencia a aquellas tarjetas que están diseñadas para que el usuario conecte en ellas bien una fuente de alimentación, o bien un generador de funciones. La tarjeta para el conexionado de instrumentos está preparada para la conexión de equipos de baja frecuencia, como puede ser un multímetro digital y bien de alta frecuencia como un osciloscopio digital de dos canales. Por su parte, las tarjetas de componentes contienen los zócalos en los que se pueden colocar los componentes que formarán parte de los circuitos. Actualmente, una nueva versión en el layout de esta tarjeta, permite disponer hasta 10 componentes de dos terminales o bien reconfigurar los relés de la tarjeta y disminuir el número de componentes de dos conectores y aumentar el número de componentes de más de dos terminales.

Cada una de las tarjetas que forman parte de la matriz de conmutación dispone de un microprocesador encargado de identificar cada tarjeta y controlar los relés de dicha tarjeta. Mediante un bus I²C, el algoritmo de control diseñado específicamente para el control de la matriz de conmutación del laboratorio VISIR y que se ejecuta como parte de la aplicación de control del Servidor de Equipos se comunica con la tarjeta correspondiente, indicando cuáles son los relés que se tienen que abrir o cerrar, en función de las conexiones a implementar. El microprocesador de la tarjeta recibe esas órdenes, y se encarga de ejecutar la apertura o cierre de los relés que gestiona.

En la plataforma VISIR únicamente se pueden crear circuitos con un máximo de 8 nodos, mientras que el número de ramas de un circuito depende del número de tarjetas de componentes de que se disponga. Los

relés disponibles son de dos tipos: de contacto simple o doble. Los relés de contacto simple, funcionan como un interruptor, es decir, unen dos puntos del circuito. En cambio los relés dobles funcionan como dos interruptores, poniendo en contacto dos puntos pertenecientes al circuito. En el caso de la conexión de los instrumentos al circuito se pueden utilizar relés simples, ya que únicamente se pone en contacto un nodo del circuito con una sonda del instrumento.

Un ejemplo de circuito electrónico que se puede disponer sobre la matriz de conmutación para la realización de prácticas de laboratorio desde la plataforma VISIR se puede ver en la Figura 2.27. Sobre el esquema se pueden apreciar todos los nodos disponibles así como las ramas formadas por los componentes dispuestos en la matriz de conmutación. En este esquema la referencia *jumper_lead* indica una resistencia de 0 ohmios, utilizada bien para interconectar varios nodos o bien para realizar medidas de intensidad en el circuito, ya que el proceso para medir corrientes es el mismo que sobre un circuito real: se debe abrir una rama del circuito e introducir entre esos dos puntos abiertos el multímetro.

En el método para el desarrollo seguido en la plataforma VISIR se permite a los alumnos cometer fallos a la hora de diseñar el circuito, pero siempre controlando estos errores para que no se produzca ninguna situación de peligro en el laboratorio, como por ejemplo un cortocircuito.

Para evitar estas situaciones se ha desarrollado una aplicación que se ejecuta en el Servidor de Medidas que se encarga de comparar la lista de conexiones enviada por el cliente de experimentos, con la lista de conexiones definida previamente por el profesor de la asignatura. Así pues, si alguna de las conexiones solicitada por el usuario no se encuentra incluida en la lista configurada por el profesor el circuito no es válido y se le mostrará un mensaje de error al usuario, sin indicarle cuál o dónde está el error, por lo que deberá ser él quien razone dónde puede encontrarse el fallo.

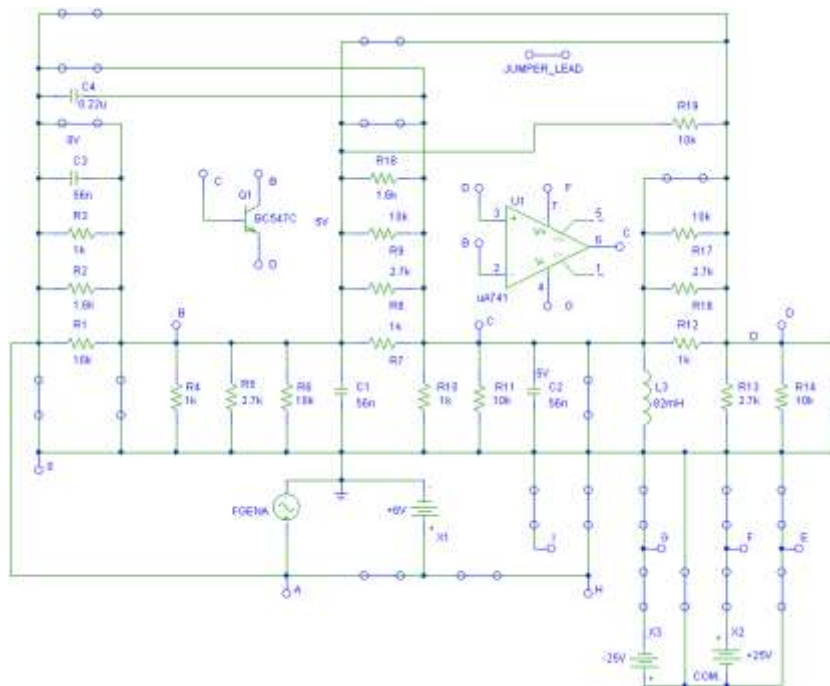


Figura 2.27: Configuración de los componentes disponibles en la plataforma VISIR.

La labor de crear este listado, tal y como se ha señalado no es una tarea sencilla, primero por el número máximo de nodos permitidos por la matriz y segundo porque a mayor número de componentes, mayor número de placas es necesario y la complejidad del cableado aumenta. Hay que tener en cuenta también que el espacio físico para disponer componentes en la matriz es limitado, por lo que habrá que evitar la redundancia de componentes en la medida de lo posible, comprobando cada vez que se añade un circuito a la matriz si alguno de los componentes existe previamente, y si se encuentra entre los mismos nodos del circuito.

2.2.4.4 Resumen método de control en VISIR

El método de control empleado en la plataforma hardware del VISIR ha sido desarrollado para reproducir de manera remota los mismos experimentos que el alumno puede realizar en el laboratorio tradicional. Para ello, una de las acciones más importantes es poder construir físicamente un circuito eléctrico, seleccionando los componentes necesarios y realizando el cableado correspondiente entre ellos y posteriormente conectar los instrumentos necesarios para comprobar su funcionamiento. Para poder ejecutar esta

acción de manera remota, la plataforma VISIR utiliza una matriz de conmutación equipada con relés electromecánicos [GZO04].

Gracias el interfaz gráfico proporcionado por el cliente y la matriz de conmutación, el alumno es capaz de llevar a cabo experimentos interactivos a través de su navegador web, y modificar las conexiones entre los componentes disponibles en la matriz de conmutación, según las configuraciones diseñadas por el profesor, con el valor añadido que el circuito se creará de manera real, y las medidas serán totalmente reales, iguales a las que se obtendrían en las instalaciones de la universidad.

Así, según disponga el profesor se podrán realizar circuitos básicos como puede ser la medida de resistencias o circuitos más complejos como el estudio de las distintas configuraciones de un amplificador operacional, pero siempre teniendo en cuenta las limitaciones de número de conexiones posibles impuestas por la matriz de conmutación.

A modo de resumen, la Tabla 2.5 muestra las tecnologías utilizadas en el desarrollo del método de control empleado en el laboratorio VISIR, teniendo en cuenta que la mayor parte de la aplicación de control reside en el Servidor de Equipos, por lo que no consideramos el Servidor de Medidas como parte del método de control.

2.2.5 iLab. Massachusetts Institute of Technology

En el año 1998 se comenzó a desarrollar el iLab (<http://iLab.mit.edu>) de la mano de Jesús del Álamo, profesor del MIT que por aquel entonces impartía clases de electrónica en las que mostraba a los alumnos los principios básicos del funcionamiento de semiconductores. A su entender, las clases teóricas necesitaban del apoyo de ejercicios prácticos de laboratorio, pero realizarlos con el gran número de alumnos que asistían a sus clases y disponiendo además de un único puesto de trabajo era muy complicado [HAL⁺08].

Aplicación de control	Interfaz de control	Control experimentos	Tipo interacción posible
Desarrollada en LabVIEW Drivers VXIPlug&play de NI	PXI - GPIB	Matriz de conmutación propia	Modificación lógica del experimento

Tabla 2.5: Resumen tecnologías empleadas por en el método de control VISIR

Partiendo de una subvención proporcionada por Microsoft y que definió la alianza iCampus entre esta empresa y el MIT, del Álamo comenzó a desarrollar lo que serían los inicios de una plataforma que hiciera posible el acceso remoto al puesto de trabajo que disponía. Esta primera versión, denominada Microelectronics WebLab, se basaba en un applet de Java que permitía enviar descripciones a un servidor el cual se encontraba conectado al equipo del laboratorio. En otoño de 1998 el laboratorio remoto comenzó a utilizarse en las prácticas de laboratorio, y en la primavera de 1999 cerca de 100 alumnos había realizado ya las prácticas de manera remota.

Poco a poco se fueron integrando más experimentos en el Microelectronics WebLab, así como más instrumentos, de modo que en el año 2002 el Microelectronics WebLab contaba ya con 6 experimentos distintos y diferente hardware [AHM⁺02]. El éxito del laboratorio hizo que varias universidades se interesasen por cómo poder utilizarlo, pues la ventaja del acceso remoto permite que alumnos de otras universidades accedan a los laboratorios proporcionados por otras instituciones. Así por ejemplo en el año 2000 se creó la alianza Singapore-MIT Alliance (SMA), entre el MIT y varias universidades de Singapur.

Actualmente, los equipos de trabajo del iLab han creado laboratorios remotos en el MIT en temáticas tan diversas como la microelectrónica, ingeniería química, estudio de la cristalización de polímero, ingeniería estructural y procesamiento de señales. El objetivo es mostrar como una misma arquitectura de laboratorio remoto, que analizaremos a continuación, puede satisfacer necesidades de funcionamiento complejas, de diferentes temáticas y escalar su uso a grandes grupos de estudiantes en el MIT y todo el mundo. Los laboratorios disponibles son [Exp11]:

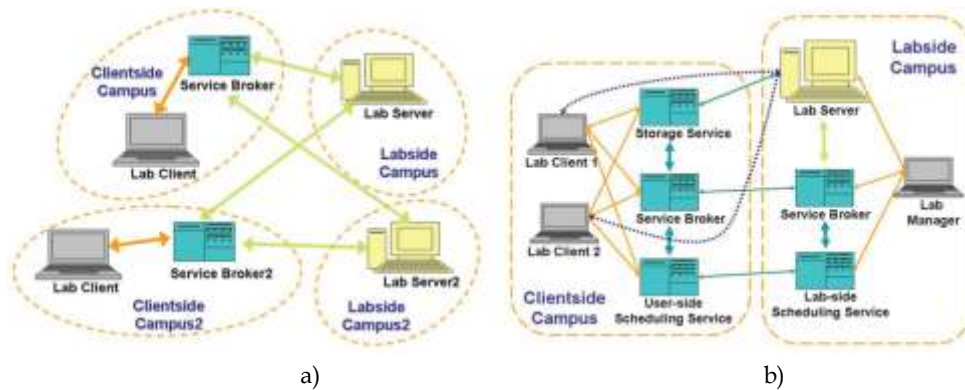
- Electrónica:
 - Microelectronics Device Characterization Lab
 - Dynamic Signal Analyzer (DSA) lab
 - ELVIS Lab
 - Digital Electronic experiment
 - Elvis Op-amp experiment
 - Digital Logic Gate
- Control:
 - Inverted pendulum

- Power Flow Control Generators
- Física:
 - Force On a Dipole Experiment
- Espectrómetros:
 - Neutron Beam
 - Maxwell Boltzmann Distribution
 - Bragg Diffraction and the DeBroglie Wavelength
- Telecomunicaciones
 - Nyquist Sampling Theorem
 - Pulse Code Modulation
 - Line Coding Schemes
 - Amplitude Modulation
 - Frequency Modulation

2.2.5.1 Arquitectura del iLab

En el año 2001 el *Center for Educational Computing Initiatives* del MIT comenzó a desarrollar una arquitectura basada en Servicios Web, que ofrecía una infraestructura común para todos los experimentos, permitiendo compartirlos entre diferentes universidades. De este modo, tomando como base esta arquitectura denominada ISA (iLab Shared Architecture) se desarrolló en el año 2004 una nueva versión del Microelectronics WebLab pero ya integrado como experimento dentro de la plataforma denominada iLab.

El iLab es por tanto un laboratorio remoto centrado en proveer un sistema sobre el que desarrollar experimentos remotos, teniendo como referencia la arquitectura ISA [HDB⁺08]. Esta arquitectura permite controlar experimentos de diferentes tipos, ya sean por lotes o interactivos, mediante diferentes sistemas operativos y plataformas de desarrollo. Además, esta arquitectura ISA proporciona herramientas de gestión y utilidades para los proveedores de laboratorios y tiene un diseño inminentemente escalable. Dependiendo del tipo de experimentos ISA ofrece dos tipos de arquitecturas, una para experimentos por lotes (Figura 2.28a) y otra para experimentos interactivos (Figura 2.28b).



a) b)
Figura 2.28: Arquitecturas disponibles en ISA

En la arquitectura diseñada para el uso con experimentos por lotes, cada campus, que podría representar a dos universidades diferentes, dispone de un Lab Server, el cual contiene la lógica del experimento y un Service Broker, encargado de la gestión de los experimentos. En la arquitectura propuesta para su uso con experimentos interactivos, se optimiza la comunicación entre el cliente y el Lab Server, eliminando la necesidad de una base de datos intermedia, permitiendo así la comunicación directa entre ambos. Ante este paradigma, si se quiere mantener un registro de las actividades del cliente en el laboratorio es necesario disponer de un servicio que proporcione esa información: el Experiment Storage Service, el cual es llamado por el Lab Server cada vez que el cliente solicita una acción sobre un experimento.

2.2.5.2 Aplicación de control en iLab

Tal y como se ha descrito previamente, la plataforma iLab contiene laboratorios de muchos tipos y temáticas. En concreto para nuestro estudio nos vamos a centrar únicamente en dos, aquellos que están más enfocados en el área de la electrónica analógica y los que más similitud tienen con el resto de laboratorios que hemos analizado y con el laboratorio en el cual validaremos el método de control que proponemos. Estos laboratorios son el Microelectronics Device Characterization Lab y el ELVIS Lab. En cada uno de estos dos laboratorios las tecnologías de desarrollo utilizadas para la codificación de la aplicación de control son diferentes, siendo .Net en el caso del Microelectronics Device Characterization Lab y LabVIEW en el caso del ELVIS Lab [Gik06] [ACH+03].

Uno de los logros de la arquitectura ISA desarrollada en el proyecto iLab es proporcionar mecanismos para convertir de manera rápida y sencilla

un laboratorio independiente y aislado, en una aplicación web accesible a través de Internet. La arquitectura ISA hace posible que los propios algoritmos diseñados mediante LabVIEW para el control local del experimento puedan ser usados como interfaces de usuario mediante la aplicación LabVIEW Web Server incluida en el propio entorno de desarrollo. Así, el panel frontal de control del experimento puede ser mostrado mediante el LabVIEW Remote Panel Server de una manera sencilla y rápida. iLab proporciona además una serie de paneles de control genéricos desarrollados en LabVIEW que facilitan el rápido desarrollo de aplicaciones de una manera controlada y segura.

En las arquitecturas mostradas anteriormente, es el Lab Server la entidad en la cual se ejecutan las aplicaciones de control de los experimentos, así como de enviar la respuesta de los experimentos al cliente. En esta arquitectura el Lab Server recibe las peticiones de los clientes a través del Service Broker, el cual se encarga de gestionar el intercambio de información entre el Lab Client y el Lab Server.

El Lab Server se encarga de ejecutar dos procesos complementarios: un interfaz Web y una aplicación de control encargada de la ejecución de los experimentos, ambos conectados a través de una base de datos persistente, tal y como puede apreciarse en la Figura 2.29. El interfaz Web implementa los servicios Web que el Service Broker ejecuta cada vez que se quiere realizar la ejecución de un experimento, proporcionando también los métodos para la validación del experimento y el interfaz de administración del laboratorio [VDL05].

Respecto al tipo de drivers empleados, en ambos laboratorios analizados las aplicaciones de control que se ejecutan en el LabServer utilizan drivers VXIPlug&play para el control de los instrumentos asociados a los experimentos, para lo cual emplean los comandos SCPI propios de cada equipo empleado en los experimentos.

2.2.5.3 Control de experimentos e interfaz de control en iLab

El laboratorio iLab no se trata de un laboratorio remoto como los estudiados anteriormente. Tal y como veíamos en al principio de su análisis, el iLab dispone de laboratorios con experimentos de temáticas muy diversas, en donde todos ellos comparten la misma arquitectura definida por ISA.

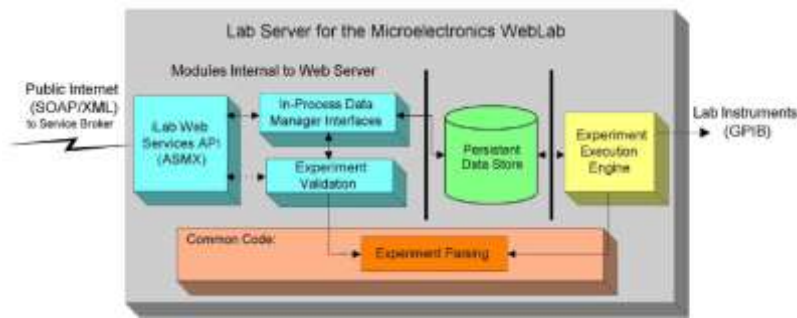


Figura 2.29: Estructura del Lab Server

En cada uno de estos laboratorios se emplean una serie de mecanismos de control y de instrumentos propios de cada disciplina, aunque de una manera generalizada es GPIB el estándar de control más utilizado por el Lab Server para comunicarse con los experimentos incluidos en los laboratorios de electrónica [AHM⁺02].

Respecto a los sistemas de control de experimentos empleados en los dos laboratorios analizados, en el caso del Microelectronics WebLab los equipos disponibles son instrumentos independientes controlados por el estándar GPIB: un equipo data logger Agilent 34970A encargado de recopilar la información del experimento, un analizador de parámetros de semiconductores Agilent 4155C utilizado en la parametrización de componentes electrónicos y una matriz de conmutación comercial Agilent E5250A encargada de seleccionar cada uno de los semiconductores disponibles en el laboratorio [ABM⁺02] y que el alumno podrá caracterizar a través del interfaz web.

En el caso del ELVIS iLab, el sistema de control del laboratorio está basado en la plataforma NI-Elvis, la misma empleada en el laboratorio RemotElectLab analizado anteriormente, por lo que en ambos laboratorios los instrumentos disponibles son los mismos. Junto con ellos, el laboratorio ELVIS Lab dispone de una matriz de conmutación de National Instruments SCXI-1169 que permite seleccionar el experimento sobre el que se quiere trabajar [Gik06].

2.2.5.4 Resumen método de control en iLab

En los dos laboratorios pertenecientes a la plataforma iLab analizados, el alumno puede desarrollar experimentos por lotes, destinados generalmente a

la caracterización de dispositivos y testeo de circuitos electrónicos sencillos, aunque en el caso del laboratorio ELVIS Lab, gracias a la placa de prototipado incluida en la plataforma NI-Elvis, se puede aumentar la capacidad y complejidad de los circuitos. En ambos laboratorios el alumno únicamente puede visualizar el desarrollo del experimento, permitiéndole el sistema modificar únicamente alguno de los parámetros de los instrumentos utilizados durante el proceso de test y medida. Se trata por tanto en ambos casos de laboratorios de instrumentación en los que no se puede modificar la estructura o la lógica del sistema bajo prueba.

A modo de resumen, la Tabla 2.6 recoge las tecnologías empleadas en el método de control utilizado en los dos laboratorios pertenecientes a la plataforma iLab que han sido analizados en esta sección.

2.2.6 Otras plataformas de experimentación remota

En esta sección hemos incluido una descripción breve de una serie de metodologías de control desarrolladas en otros laboratorios remotos, bien porque estuvieron activas y supusieron un avance en este tipo de metodologías hace unos años o bien porque son de reciente aparición y aún no disponemos de mucha información al respecto. En ambos casos, la información publicada donde se describen estos trabajos no es elevada, por lo que se ha decidido recogerlas en un mismo apartado, pero hemos considerado que sus aportaciones son importantes por lo que han sido incluidas en este análisis, describiendo de manera sucinta sus características más importantes.

	Aplicación de control	Interfaz de control	Control experimentos	Tipo interacción posible
Microelectronics Lab	Desarrollada en .NET Drivers VXIPlug&play y comandos SCPI	GPIB	Matriz de conmutación comercial	Observación únicamente de E/S
ELVIS Lab	Desarrollada en LabVIEW Drivers VXIPlug&play de NI	GPIB	Matriz de conmutación comercial	Observación únicamente de E/S

Tabla 2.6: Resumen tecnologías empleadas por en el método de control iLab

2.2.6.1 LEDFE. Universidad Nacional de Rosario

El Laboratorio Experimental a Distancia de Física Electrónica (LEDFE) ha sido desarrollado por el Departamento de Educación a Distancia perteneciente a la Facultad de Ciencias Exactas, Ingeniería y Agrimensura de la Universidad Nacional de Rosario (Argentina). Actualmente, este laboratorio se encuentra disponible en el sitio web <http://labremf4a.fceia.unr.edu.ar>

El primer desarrollo de este laboratorio [LP07] fue realizado por un grupo de estudiantes como trabajo de su proyecto final de carrera de Ingeniería Electrónica, por lo que el principal objetivo era obtener un prototipo hardware y software de bajo coste que permitiera a cualquier usuario conectado a Internet realizar experimentos y mediciones a distancia de parámetros y el análisis mediante curvas de dispositivos semiconductores reales. Este primer prototipo fue aceptado por los docentes, pero era necesario realizarle una serie de modificaciones para poder ser aceptado como una herramienta de aprendizaje al servicio de los alumnos matriculados en la asignatura Física IV, en la cual se introduce a los alumnos en el comportamiento de este tipo de componentes electrónicos y por tanto, en la que tiene perfecta cabida un laboratorio como este [LMP⁺08].

Tras sucesivas revisiones y mejoras del laboratorio, éste ha sido aceptado para su empleo docente, tanto por parte de los profesores como por parte de los alumnos, y actualmente es utilizado como complemento a las clases teóricas presenciales en la asignatura Física IV [LMV09], para mostrar a los alumnos cómo se realiza la caracterización de semiconductores. En estos momentos, mediante la plataforma LEDFE se pueden caracterizar los siguientes dispositivos: diodo Zener (1N4736), transistor BJT (BC548), transistor J-FET (2SK117) y un foto-transistor (4N28). Las hojas de características de cada dispositivo se encuentran a disposición de los usuarios en cada uno de los experimentos disponibles. La arquitectura sobre la que se basan los desarrollos de este laboratorio remoto está basada también en el paradigma cliente-servidor, tal y como se puede observar en la Figura 2.30.

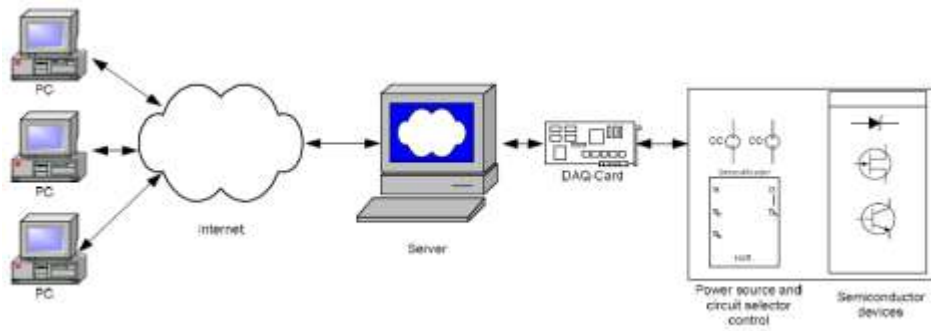


Figura 2.30: Arquitectura del laboratorio LEDFE

La aplicación de control del laboratorio LEDFE ha sido desarrollada íntegramente en .NET para su funcionamiento en un ordenador que contenga un sistema operativo Windows XP. Esta aplicación se encarga de recibir las órdenes enviadas desde el interfaz de usuario y gestionadas por un servidor TCP/IP desarrollado también mediante .NET. El uso de esta tecnología está impuesto por la propia tarjeta de adquisición de datos y su driver, ya que son dispositivos antiguos que requieren ser gestionados por un servidor TCP/IP desarrollado mediante estas tecnologías [CDK01].

El laboratorio LEDFE no dispone como tal de una serie de instrumentos para el testeo y medida del circuito bajo prueba, tal y como se ha visto en los laboratorios anteriores. A través de la tarjeta de adquisición de datos y de una pequeña matriz de conmutación desarrollada a medida en una placa de prototipos, se selecciona el componente a caracterizar y se realizan las medidas de tensión e intensidad requeridas.

La tarjeta de adquisición de datos de National Instruments DAQ-1200 dispone de ocho canales de entradas analógicas, dos salidas analógicas y ocho líneas digitales que pueden ser configuradas como entradas o salidas. Las entradas analógicas son utilizadas para realizar las medidas de tensión, mientras que las salidas analógicas son configuradas por el servidor para proporcionar la alimentación seleccionada al experimento. Al mismo tiempo se utilizan tres de las líneas digitales para la selección del experimento.

La matriz de conmutación desarrollada permite seleccionar un experimento entre los cinco actualmente disponibles así como proporcionar la alimentación al experimento. La tensión de la alimentación está proporcionada por la tarjeta de adquisición de datos, que está conectada al

servidor a través de un bus PCI. Adicionalmente, y para evitar problemas en la tarjeta de adquisición, esta tarjeta incluye una etapa de alimentación para proporcionar la corriente necesaria por cada uno de los experimentos.

Los experimentos a desarrollar en este laboratorio están basados en la caracterización de semiconductores mediante sencillos circuitos electrónicos que no pueden ser modificados por los usuarios, por lo que es un laboratorio de pura observación.

2.2.6.2 eVALIDATE. Dayalbagh Educational Institute

El Virtual Advanced Laboratory for Interactive Design, Analysis&Test in Electronics (eVALIDATE), es la plataforma educativa desarrollada en el Departamento de Física e Informática del Dayalbagh Educational Institute, en Agra (India).

A pesar de la escasa información que se ha podido obtener referente al desarrollo usado en la plataforma eVALIDATE debido a su reciente aparición [MBG10], se ha incluido en esta sección ya que junto con la metodología defendida en este trabajo de investigación, son las únicas que utilizan la tecnología LXI como interfaz de comunicaciones entre la aplicación de control y los experimentos del laboratorio, aunque ambas lo emplean de manera diferente, ya que en la solución propuesta no usaremos el servidor web integrado en los instrumentos.

Todos los equipos utilizados en eVALIDATE son equipos compatibles con la tecnología LXI, por lo que son controlables a través de un interface LAN y proporcionan de fábrica un servidor web que permite al usuario acceder a las funciones del instrumento en tiempo real.

La plataforma eVALIDATE ha sido desarrollada utilizando estos interfaces web sin realizar ninguna modificación sobre ellos más que adecuando sus dimensiones para poder mostrarlos todos en una misma pantalla de explorador, tal y como se puede distinguir en la Figura 2.31.

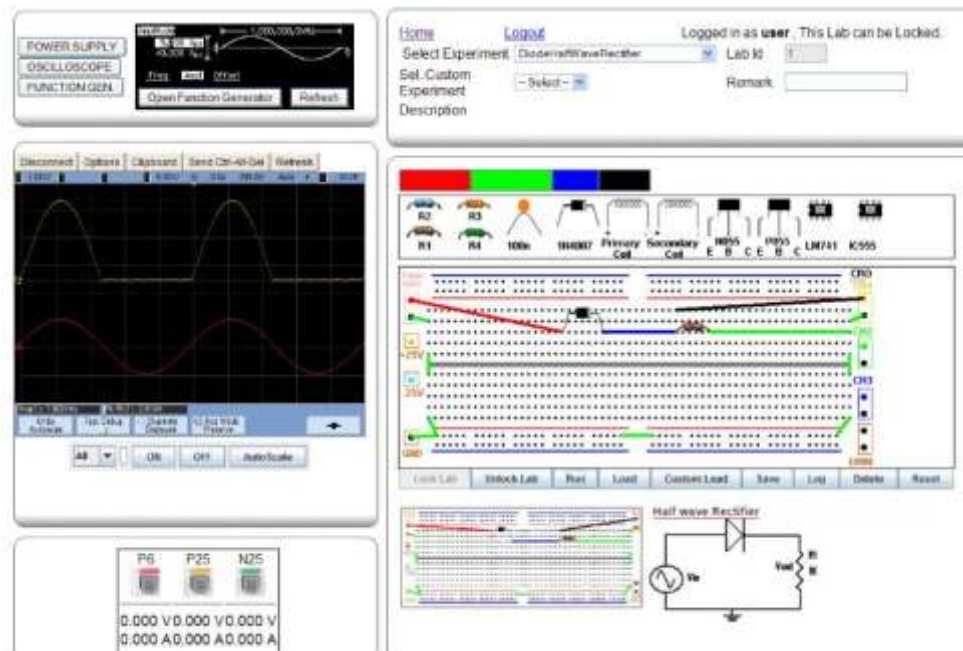


Figura 2.31: Interfaz de la plataforma desarrollada en eVALIDATE

El método desarrollado en la plataforma eVALIDATE es similar a los vistos anteriormente en esta sección: un entorno virtual que permite acceder a los recursos físicos del laboratorio y poder crear y medir circuitos con componentes reales.

Para ello, el usuario dispone de una placa de prototipos a la que puede arrastrar los componentes disponibles en la caja de componentes (sobre la placa de prototipos en la Figura 2.31) e interconectarlos entre sí y a los instrumentos de medida. Una vez configurado el experimento, el usuario pulsa el botón 'run' y se envía un fichero de configuración al servidor, el cual lo analiza para extraer el listado de componentes y conexiones para implementar el circuito de manera real mediante una matriz de conmutación y configurar los equipos según las acciones del usuario sobre los paneles frontales.

Toda la virtualización de los componentes y los equipos disponibles en la plataforma eVALIDATE están desarrollados mediante tecnología HTML, JavaScript, Java y lenguaje Pearl para el interface CGI. El software de la aplicación web, en el cual se integran la placa de prototipos virtual, los componentes y los instrumentos, está desarrollado en HTML, JavaScript y Java, por lo que el usuario deberá tener instalado el *run time* de Java

adecuado y permitir los pop-ups en su navegador para el correcto funcionamiento de la aplicación web.

Todos los instrumentos son compatibles con el estándar LXI, por lo que se ha creado con ellos una red LAN en la que todos están interconectados mediante un *hub* o concentrador, formando una topología en estrella, en donde el servidor es el encargado de enviar las peticiones del cliente a los instrumentos.

Los equipos disponibles son: osciloscopio digital, fuente de alimentación, generador de funciones, multímetro digital y la matriz de conmutación comercial, encargada de la realización física de las conexiones del experimento. Cada uno de los equipos dispone de una página web a través de la cual se puede acceder al panel frontal para su control remoto.

Estos paneles de control virtuales tratan de reproducir el panel frontal del equipo aunque no siempre lo consiguen, como sucede en este caso con el osciloscopio o la fuente de alimentación, por lo que el usuario no puede reproducir exactamente las mismas acciones que el laboratorio real.

Por otro lado, los comandos enviados desde el cliente para controlar los equipos siguen el estándar SCPI, por lo que son comandos específicos de dichos instrumentos. Es decir, esta metodología es únicamente factible si se usan los equipos para los que está configurada la plataforma eVALIDATE.

2.2.6.3 RwmLAB. Western Michigan University

La plataforma Remote Wiring and Measurement Laboratory (RwmLAB) fue desarrollada por el laboratorio de Ingeniería Electrónica e Informática de la Western Michigan University. Este es uno de los primeros laboratorios desarrollados que tienen como elemento principal del sistema de control de los experimentos, una matriz de conmutación y un entorno virtual web que permitía crear circuitos reales sobre dicha matriz de conmutación, y posteriormente realizar medidas sobre ellos [AT01].

En este laboratorio remoto, gracias a un sencillo interfaz web, los estudiantes pueden cablear un circuito electrónico en una placa de prototipos y posteriormente hacer medidas y capturar los resultados mediante los paneles virtuales de los instrumentos disponibles.

Su aplicación de control está desarrollada mediante la combinación de LabVIEW y lenguaje C. Como se puede ver en la Figura 2.32, el usuario dispone de una serie de resistencias, condensadores y diodos para realizar los experimentos, así como una serie de instrumentos para comprobar su funcionamiento. Cada vez que el alumno desea configurar un instrumento, se le abre una nueva pantalla en la cual aparece el panel de control desarrollado en LabVIEW, tal y como puede apreciarse en la parte inferior derecha de la Figura 2.32, donde se aprecia el panel frontal del osciloscopio. También dispone de una cámara web para seguir en tiempo real cómo se ejecutan sus órdenes sobre la plataforma real instalada en el laboratorio de la universidad.

Una vez que el usuario realiza el diseño del circuito en la placa de prototipos virtual, se genera un fichero en el cual quedan representados los componentes y las conexiones realizadas en cada circuito. Este fichero se envía al servidor, el cual lo decodifica y actúa sobre los instrumentos. Toda esta aplicación está desarrollada en lenguaje C. Para la creación de los circuitos, el sistema de control de los experimentos en el RwmLAB está basado en una matriz de conmutación de diseño propio formada por un array de relés en configuración 8x8, controlada por un microservidor web (Xecom's AWC86A) y un CPLD [AO04].

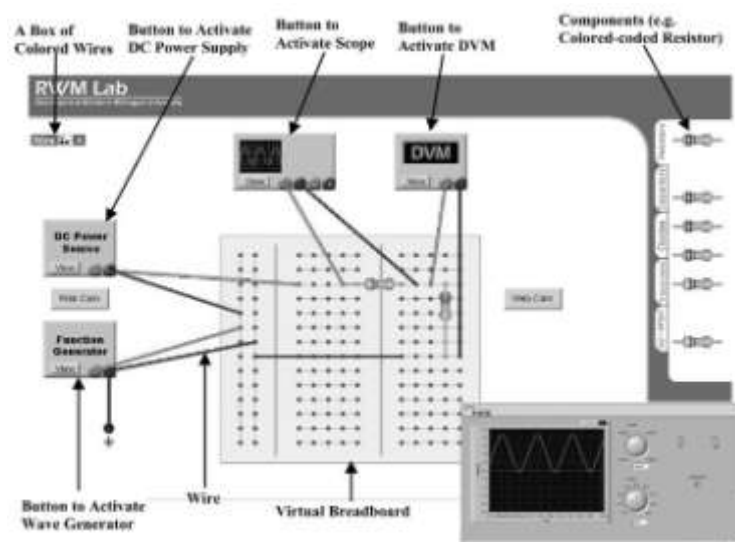


Figura 2.32: Interfaz de usuario del RwmLAB

	Aplicación de control	Interfaz de control	Control experimentos	Tipo interacción posible
LEDFFE	Desarrollada en .NET Drivers propios	Tarjeta adquisición de datos	Matriz de conmutación propia	Observación únicamente de E/S
eVALIDATE	Desarrollada en Java Drivers propios y comandos SCPI	LXI	Matriz de conmutación comercial	Modificación lógica del experimento
RwmLab	Desarrollada en LabVIEW y C Drivers propios y comandos SCPI	GPIB	Matriz de conmutación propia	Modificación lógica del experimento

Tabla 2.7: Resumen tecnologías empleadas en LEDFFE, eVALIDATE y RwmLab

Una vez que el usuario realiza el diseño del circuito en la placa de prototipos virtual, el servidor recibe la configuración de los componentes y el cableado, y por medio del CPLD se abren o cierran los relés correspondientes que ponen en contacto los componentes seleccionados. El número de componentes viene limitado por el número de conexiones disponible en la matriz de 8x8 [ATF+05]. Junto con la matriz de conmutación, el laboratorio dispone de un multímetro digital, un osciloscopio, una fuente de alimentación y un generador de funciones. Los comandos de configuración y lectura de datos de estos equipos siguen el estándar SCPI y se emplea como interfaz de control GPIB entre la aplicación software y los instrumentos. Todos los equipos disponen de una cámara web enfocada a sus paneles frontales, para que el usuario pueda ver en tiempo real la progresión de sus acciones sobre los instrumentos.

2.2.6.4 Resumen de tecnologías usadas en estas metodologías

Para finalizar el estudio de los diferentes métodos empleados en laboratorios remotos analizados, la Tabla 2.7 recoge las técnicas empleadas para implementar la configuración, control y monitorización de los experimentos en ellos incluidos.

2.3 Análisis comparativo de las metodologías estudiadas

En esta sección vamos a realizar un estudio comparativo de las diferentes metodologías utilizadas para el control de los experimentos empleadas en los

laboratorios remotos descritos anteriormente⁷. Este análisis nos ayudará a justificar cómo el método de control elegido a la hora de desplegar un laboratorio, determina una serie de características que definen el funcionamiento y el uso del mismo.

Para realizar este estudio, hemos seleccionado ciertas características a partir de las cuales se puede determinar la clase de experimentación posible en el laboratorio, el tipo de interacción realizada por el alumno en el mismo, así como la capacidad del laboratorio para adaptarse a cambios en la configuración del sistema de control de experimentos, añadiendo o actualizando circuitos bajo prueba e instrumentos. Estas características son las siguientes:

- En primer lugar se compararan las tecnologías empleadas en cada una de las plataformas analizadas, ya que esta selección determina en gran medida las propiedades del laboratorio respecto a su despliegue y mantenimiento, así como de la utilización que se pueda hacer del mismo.
- A continuación, se examinará el tipo de experimentación que se puede llevar a cabo en el laboratorio, analizando si el alumno únicamente puede monitorizar la ejecución del experimento sin alterar ninguno de sus parámetros, o bien si puede modificar la lógica del circuito bajo prueba y/o la configuración de los instrumentos. Estudiaremos también la complejidad de los ejercicios que el alumno puede llevar a cabo en el laboratorio. Finalizaremos comparando la naturaleza de los experimentos, si se trata de una actividad interactiva o bien el alumno debe esperar para observar los resultados del experimento.
- Estudiaremos las diferentes acciones que el alumno puede llevar a cabo sobre el experimento mediante el método de control empleado en el laboratorio. Esta característica nos permitirá determinar la capacidad del método de control para reproducir de manera remota las mismas acciones que si el alumno se encontrara en el laboratorio manual análogo al remoto.
- Finalmente, cotejaremos de manera cualitativa el sistema de control empleado en cada una de estas plataformas, ya que la selección de un

⁷ En este análisis no se incluyen los laboratorios eVALIDATE y RwmLAB debido a la falta de información disponible sobre los mismos. En el caso del primero, debido a su reciente aparición, y en el caso del RwmLab ya que se trata de un laboratorio en desuso y antiguo.

sistema u otro determina la capacidad del método de control para albergar nuevos experimentos o adaptarse a cambios en los ya existentes.

Para realizar el análisis de cada una de estas características hemos utilizado los indicadores más representativos de cada una de ellas, si bien es posible que no aparezcan todos los posibles y que alguno pueda ser utilizado para analizar varias características. Se trata por tanto de un estudio objetivo del que hemos obtenido nuestras propias conclusiones pero que puede ser adaptado en función de las características a analizar.

Una vez finalizado este análisis y para concluir esta sección, se mostrarán las conclusiones a las que llegamos y que han supuesto el punto de partida en el diseño y desarrollo del método de control propuesto en este trabajo de investigación.

2.3.1 Tecnologías empleadas en cada método de control

Debemos tener presente que la selección de las tecnologías empleadas en cada uno de los sistemas que forman el método de control desplegado en un laboratorio remoto determinan en gran medida sus características y propiedades. Por ello, en esta sección vamos a comparar cada una de las tecnologías empleadas para el desarrollo del software de control y el despliegue del interfaz de comunicaciones, ya que su combinación, junto con la selección del tipo de drivers empleados para el acceso a los instrumentos, determinará la facilidad para actualizar o añadir equipos al experimento. Las tecnologías empleadas en el sistema de control de experimentos las analizaremos en la sección 2.3.4 ya que es un sistema clave en el diseño del método de control.

Si se desea que la plataforma sea en un futuro actualizada con nuevos equipos, el interfaz empleado para comunicar la aplicación de control y los experimentos (circuito bajo prueba e instrumentos) debe ser lo suficientemente escalable y basado en estándares, para que modificar los circuitos bajo prueba o los instrumentos de los experimentos sea sencillo o bien no requiera una reconfiguración completa de dicha aplicación. Además, se debe utilizar un método de control y gestión de los experimentos que haga posible utilizar distintos tipos de instrumentos de distintos proveedores,

abriendo ampliamente el abanico de posibilidades sin estar restringido el laboratorio al uso de un fabricante determinado.

En el caso del laboratorio remoto NetLab, el software de control está desarrollado en C mientras que el interfaz de comunicaciones empleado está basado en los buses de datos VXI y GPIB, utilizando comandos de control SCPI sobre drivers VXIplug&play. Tanto VXI como GPIB son estándares ampliamente utilizados, por lo que la adaptación del laboratorio a nuevos instrumentos sería relativamente sencilla. El problema ante una actualización de instrumentos aparece cuando el control de los equipos se realiza a través de comandos SCPI. Como se indicaba en el capítulo anterior, cada instrumento y cada fabricante tiene sus propios comandos de control SCPI, por lo que en el caso del NetLab solamente cambiar un equipo por otro supondría tener que actualizar íntegramente el software de control de dicho experimento. Es decir, una solución basada en VXI, GPIB y comandos SCPI no es fácilmente escalable y adaptable a nuevos instrumentos y requerirían modificaciones en el software desarrollado.

En el caso del RemotElectLab, la metodología de control está basada en el uso del equipo NI-Elvis por lo que la aplicación de control está desarrollada íntegramente en LabVIEW, basando el control de los equipos en los drivers propios de la plataforma NI-Elvis. Ésta es una plataforma cerrada y no actualizable, que dispone de sus propias herramientas de desarrollo y métodos de control. Es decir, desde el punto de vista de la instrumentación, el laboratorio RemotElectLab no es ni escalable ni reconfigurable, a no ser que se añadan instrumentos controlados mediante otros sistemas, ya sean USB, GPIB, LXI, etc., lo que requerirá una programación nueva para el control del dispositivo.

En el desarrollo del ISILab, los instrumentos emplean GPIB como interfaz de comunicaciones con el servidor. La aplicación de control está basada en el estándar IVI y desarrollada en Java, por lo que la selección de estas tecnologías de programación hace posible que el laboratorio sea fácilmente adaptable bien a nuevos equipos o a instrumentos de diferentes proveedores.

La plataforma VISIR presenta un método de control cerrado ya que todo el software de control desarrollado en el Servidor de Equipos está realizado mediante los drivers específicos de los instrumentos incluidos en el

chasis PXI utilizado. Desde este punto de vista, añadir nuevos equipos o instrumentos de un nuevo fabricante requeriría la readaptación completa del software de este servidor e incorporar un chasis PXI con mayor capacidad.

El laboratorio LEDFE no dispone de ninguno de los instrumentos típicos de un laboratorio de electrónica analógica, sino que la adquisición de datos del experimento y su configuración se realizan a través de una tarjeta de adquisición de datos conectada al bus PCI del servidor, el cual se encarga de su gestión y control a través del driver propietario del fabricante, en este caso National Instruments, quien impone su funcionamiento exclusivo en el sistema operativo Windows. Es decir, se trata de un laboratorio de electrónica que no utiliza ningún sistema de control de los analizados anteriormente y que está desarrollado a medida para esta tarjeta de adquisición de datos, por lo que su adaptación a nuevos instrumentos es complicada.

En el caso de la plataforma iLab, la arquitectura ISA permite que los experimentos puedan ser controlador mediante diferentes interfaces, en función de las necesidades del laboratorio. En el caso de los dos laboratorios estudiados, el Microelectronics WebLab utiliza GPIB y SCPI para la comunicación de la aplicación de control desarrollada en .Net y los experimentos, mientras que el ELVIS iLab utiliza el NI-Elvis como plataforma hardware para la caracterización de los dispositivos a través de una aplicación de control desarrollada íntegramente en LabVIEW. En ambos casos, sustituir los equipos por otros nuevos, implicaría la necesidad de adaptar el software de control diseñado. Aún así, las facilidades ofrecidas por la arquitectura ISA harían más sencilla esta tarea.

A modo de resumen, la Tabla 2.8 recoge las tecnologías empleadas tanto para la codificación de la aplicación como en el despliegue del interfaz de control en cada una de las plataformas analizadas. La calificación que otorgada a cada uno de estos indicadores (un asterisco equivaldría a difícil y cinco a fácil), es fruto del estudio de las tecnologías realizado al comienzo de este capítulo y de la experiencia que disponemos al trabajar tanto con los diferentes tipos de drivers empleados en las aplicaciones de control, como de los interfaces de control utilizados para la comunicación entre las aplicaciones de control y los experimentos.

Plataforma	Aplicación Control y Drivers	Interfaz control	Adaptación a nuevos instrumentos
NetLab	Desarrollada en C Drivers VXIPlug&play	GPIB-VXI	***
RemotElectLab	Desarrollada en LabVIEW Drivers propios NI-Elvis	NI-Elvis USB alta velocidad	*
ISILab	Desarrollada en Java Drivers IVI	GPIB – RS-232	****
VISIR	Desarrollada en LabVIEW Drivers VXIPlug&play de NI	PXI-VXI	*
LEDFE	Desarrollada en .NET Drivers propios	Tarjeta adquisición de datos	*
iLab	.NET + SCPI / LabVIEW + Drivers VXIPlug&play de NI	GPIB-SCPI Ni-Elvis	**

Tabla 2.8: Tecnologías de control hardware y adaptación a nuevos instrumentos

2.3.2 Tipo de experimentación de cada método de control

En este apartado vamos a comparar los diferentes modos que el usuario dispone para realizar acciones en el laboratorio remoto. Idealmente, el alumno debe tener la posibilidad de llevar a cabo las mismas tareas que puede desempeñar en un laboratorio manual durante el transcurso de la asignatura. Generalmente, durante el proceso de aprendizaje de la electrónica analógica el alumno tiene que adquirir una serie de conocimientos teóricos y aplicarlos durante las clases prácticas, en donde construye una serie de circuitos y realiza una serie de medidas sobre ellos.

La naturaleza y el tipo de experimentos que el alumno tiene que implementar durante una sesión de laboratorio dependerá de las competencias que el profesor quiera trabajar [SCU07], pero generalmente en una asignatura relacionada con la electrónica analógica los experimentos con materiales reales (no simuladores) se pueden agrupar en dos tipos: a) el alumno puede construir un circuito en base a componentes discretos y realizar medidas sobre él, de tal modo que el alumno tiene la posibilidad de hacer cambios en el circuito o incluso crearlo partiendo de las especificaciones dadas por el profesor, o bien b) únicamente puede tomar medidas sobre un circuito previamente construido y observar los resultados para su posterior análisis.

Ambas opciones presentan sus ventajas e inconvenientes y cada una de ellas trabaja unas competencias distintas, por lo que no hay una solución mejor que otra. En el caso de trabajar en un laboratorio real, el modo de experimentación lo define el profesor, pues generalmente ambos tipos de experimentos son posibles. En cambio en un laboratorio remoto, la metodología de control del experimento puede determinar que solo se pueda trabajar un tipo de experimento o ambos.

En el caso de la plataforma NetLab, los alumnos disponen de una serie de componentes (bobinas, resistencias y condensadores) que pueden interconectar mediante la matriz de conmutación y crear así un circuito y testarlo. Este circuito se crea en un entorno virtual que posteriormente se implementa físicamente de manera remota en el laboratorio, por lo que las medidas obtenidas son siempre valores reales. Por el momento, este laboratorio solo dispone de estos componentes, por lo que el número de experimentos a desarrollar es limitado. Además, no se dispone de ninguna herramienta que facilite el uso de circuitos ya predefinidos, por lo que el alumno que quiera testar un circuito, siempre deberá construir previamente el experimento. NetLab es un laboratorio en el que el alumno puede modificar la estructura y la lógica de los circuitos, lo que denominábamos, como laboratorios de escritura y lectura.

La metodología de desarrollo llevada a cabo en el RemotElectLab permite disponer de una mezcla de ambos tipos de experimentos, ya que aunque se parte de un circuito previamente construido y definido por el profesor, el alumno puede modificar ciertos componentes del circuito (modificar resistencias, habilitar/deshabilitar partes del circuito), y examinar su comportamiento ante estos cambios y las diversas entradas al circuito. Se trata por tanto también de un laboratorio que permite la lectura y la escritura o modificación de los parámetros del experimento. El inconveniente es que el alumno siempre se debe ceñir al circuito preconstruido por el profesor y a los cambios y puntos de medida que el sistema le permite. Además y muy importante, el alumno nunca se puede equivocar, ya que las acciones que puede realizar sobre el circuito son únicas y están acotadas, por lo que hace que pierda un elemento muy importante del aprendizaje [VM10].

En el laboratorio ISILab los experimentos siempre están preconstruidos por lo que el alumno nunca tiene la posibilidad de hacer

modificaciones sobre el circuito y solamente puede analizar el comportamiento del circuito ante variaciones en las señales de entrada del mismo. Así, en este laboratorio el alumno solo puede realizar ciertas modificaciones en la instrumentación del laboratorio y observar las salidas ante los cambios en las señales de entrada del experimento en los puntos de test establecidos, pero sin cambiar su lógica interna. Es por tanto una configuración estática, en la que no tienen cabida los componentes discretos, y donde el alumno siempre dependerá de los circuitos dispuestos en la matriz por el profesor de la asignatura. Igual que en la plataforma RemotElectlab, el alumno tampoco tiene posibilidad de cometer ningún error a la hora de ejecutar el experimento, por lo que los resultados obtenidos siempre serán correctos. Únicamente queda en la mano del alumno, la interpretación de los mismos.

En el caso de la plataforma VISIR el alumno, gracias al interfaz que dispone, puede realizar exactamente las mismas acciones que si se encontrara en el laboratorio real. Puede crear un experimento partiendo de un amplio conjunto de componentes (resistencias, condensadores, transistores, circuitos integrados, etc.), o bien puede partir de un circuito ya predefinido por el profesor y cambiarle alguno de los parámetros. Gracias a los paneles de control de los instrumentos disponibles, las medidas y configuraciones de los equipos que puede llevar a cabo el alumno son iguales que las realizadas durante las sesiones en el laboratorio real. Así pues, en este laboratorio el alumno es capaz de modificar completamente la lógica del experimento, ya que puede crear el mismo los circuitos partiendo de los componentes, y puede comprobar su funcionamiento, modificando la configuración de los instrumentos según su propio criterio.

Por otra parte, la metodología de control desarrollada ha sido tal, que el alumno puede cometer fallos, y no se le informa de cuáles son. Únicamente se le indica que el circuito no es válido o bien que el circuito no es seguro. Esta última opción forma parte de una serie de reglas de seguridad para evitar que el hardware del laboratorio resulte dañado. De este modo tal y como le sucede al alumno cuando se enfrenta a un problema en el laboratorio debe discernir dónde está el problema y cómo solucionarlo.

Tanto en el laboratorio LEDFE como en los dos laboratorios analizados del iLab (Microelectronics WebLab y ELVIS iLab), el alumno no tiene la oportunidad de cablear su propio circuito, ya que los experimentos se encuentran ya previamente ensamblados. En estos laboratorios, el alumno es el encargado de interpretar y analizar los resultados obtenidos para posteriormente obtener sus propias conclusiones. También es cierto que como veíamos anteriormente, estos laboratorios están orientados a alumnos que se inician en la electrónica, por lo que los circuitos son sencillos y están orientados a mostrar los conceptos básicos de esta materia.

A modo de resumen, la Tabla 2.9 recoge el tipo de interacción que puede llevar a cabo el alumno, así como los tipos de circuitos bajo prueba con los cuales el alumno puede realizar experimentos en cada una de las plataformas analizadas. Como se indicaba anteriormente, el tipo de experimentación depende de las competencias que se quieran desarrollar, aunque sí sería recomendable que el laboratorio remoto usado en la asignatura pudiera albergar circuitos de ambos tipos, y dar al alumno la posibilidad de elegir el tipo de ejercicio a realizar. Además, la metodología de desarrollo empleada debe ser capaz de permitir que el alumno cometa fallos, como sucede en una sesión de laboratorio tradicional, y así tener que enfrentarse a ellos y analizarlos para encontrar una solución.

Plataforma	Circuitos bajo prueba			Tipo de interacción		Posibilidad cometer fallos
	Preconstruidos	Modificables	Lógica	Parámetros	E/S	
NetLab		X	X			Sí
RemotElectLab		X		X		No
ISILab	X				X	No
VISIR		X	X			Sí
iLab	X				X	No
LEDFE	X				X	No

Tabla 2.9: Tipo de experimentos e interacción en cada plataforma y tolerancia a fallos

Cada uno de estos tipos de experimentos requiere una metodología de control diferente, pues las acciones a realizar sobre el experimento, ya sea esperar un resultado, intervenir en el experimento o bien únicamente observarlo, necesitan que el laboratorio remoto disponga en cada caso de unos recursos tecnológicos diferentes. Así pues, la metodología seguida para el control de los experimentos del laboratorio determinará el tipo de experimentos que se quieren servir en cada laboratorio.

En el caso del laboratorio NetLab, los alumnos pueden realizar experimentos interactivos, en los que pueden alterar en tiempo real el circuito del que disponen y cotejar los resultados ante dichos cambios.

En cambio los alumnos en el RemotElectLab realizan experimentos por lotes, de manera que realizan en el cliente la configuración del experimento, lo ejecutan y observan los resultados en el interfaz desarrollado en LabVIEW.

Los alumnos del ISILab, aún utilizando experimentos predefinidos por el profesor, pueden interactuar en tiempo real sobre ellos, modificando según su criterio las señales de entrada del experimento, y analizando los resultados obtenidos en los instrumentos de salida.

Mediante la metodología empleada en el desarrollo y gestión del laboratorio VISIR, los alumnos tienen la impresión de estar interactuando en tiempo real sobre el experimento, aunque en realidad se trata de un laboratorio donde los experimentos son por lotes, ya que el usuario envía su configuración, se ejecuta en el laboratorio, y se devuelven los resultados al cliente. Pero estas acciones se realizan de manera tan rápida, que el usuario no se percata de ello pues apenas tiene que esperar la respuesta (en torno a milisegundos, en función de la conexión a internet utilizada), y se puede considerar que tiene el control del laboratorio en tiempo real.

En el caso de los laboratorios usados para la caracterización de semiconductores, el LEDFE, Microelectronics WebLab y ELVIS iLab, los experimentos son por lotes, en donde el usuario únicamente configura el circuito y espera la respuesta del laboratorio. Pero no debemos olvidar que la arquitectura ISA implementada en el iLab, permite la realización de experimentos interactivos, por lo que iLab sí contempla y provee de

laboratorios en los que el alumno puede controlar de manera interactiva el laboratorio.

Consideraremos circuitos difíciles aquellos que estén formados por un mayor número de componentes y por tanto que dispongan de un mayor número de nodos y mallas. De este modo, para analizar estos laboratorios desde la perspectiva de esta característica, vamos a tomar como referencia y ejemplo los experimentos mostrados en la Figura 2.33, en donde podemos observar diferentes circuitos, desde el más sencillo formado por un componente y una malla hasta uno más complejo con 6 componentes y varias mallas y nodos.

El primero de ellos se corresponde con un circuito básico y sencillo para la caracterización de un diodo. El segundo de ellos presenta un rectificador de media onda basado en el diodo anterior. Para este circuito de dificultad media, el alumno ya tiene que comprender el funcionamiento del diodo y saber aplicar lo aprendido en la caracterización, para analizar que le sucede en función de la polarización a la que está sometido. En el tercer esquema, el circuito se corresponde con un circuito más complejo que muestra un conversor AC-DC basado en un puente de diodos con un filtro a la salida. En este experimento el alumno ya tiene que ser capaz de sintetizar este esquema por sí solo en función de unos objetivos y evaluar posteriormente los resultados obtenidos.

En el laboratorio NetLab, tal y como veíamos en su análisis, los alumnos pueden realizar experimentos con pocos componentes, de manera que la complejidad de los circuitos que permite realizar la plataforma es limitada. Aún así, como es el alumno quien tiene que realizar el cableado del circuito, podemos considerar que los experimentos que se contemplan pueden ser de dificultad baja o media.

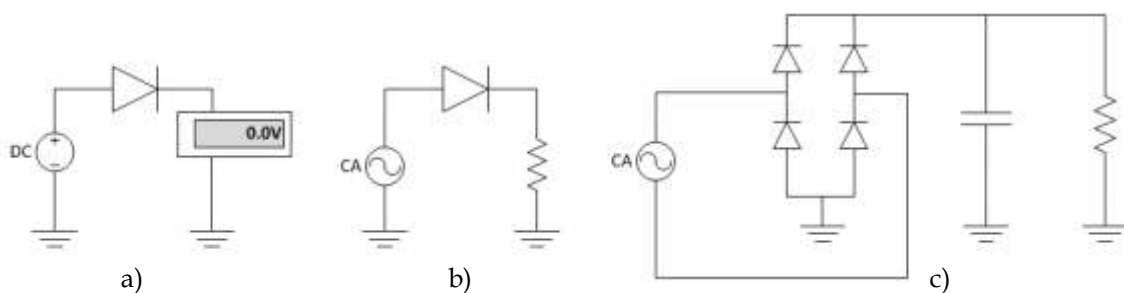


Figura 2.33: Tipos de experimentos en función de su complejidad

En el caso del laboratorio RemotElectLab, el circuito bajo prueba se encuentra pre-ensamblado por el profesor, así como los puntos de test y componentes alternativos. A este respecto, el experimento no presenta dificultad, pero sí la hay en la interpretación de los datos, ya que el circuito es una configuración electrónica avanzada.

Lo mismo sucede en el laboratorio ISILab, donde la configuración del circuito y los puntos de medida se encuentran preestablecidos. Por tanto la sencillez o dificultad de los circuitos dependerá de la selección de los experimentos que realice el profesor de la asignatura, pudiendo ser básicos o avanzados. Es decir, la plataforma ISILab se puede adaptar a experimentos con diferentes grados de complejidad y proporcionar al alumno una herramienta que puede ser usada durante todo el proceso de aprendizaje, desde el conocimiento de los componentes a la evaluación de resultados.

En el laboratorio VISIR se plantean igualmente varios escenarios posibles ya que la plataforma permite dejar total libertad a los alumnos para, partiendo de los componentes discretos, cablear sus propios circuitos, o bien el profesor puede predefinir una serie de circuitos que le aparecerán automáticamente construidos al alumno en el interfaz del cliente. De este modo, el alumno únicamente deberá configurar los instrumentos, realizar las medidas e interpretar los resultados. Así, la plataforma VISIR se presenta al igual que el ISILab como un laboratorio remoto que puede ser usado con alumnos que se inician en la electrónica y necesitan conocer los fundamentos de esta disciplina, hasta por alumnos más avanzados que requieren de una plataforma que les permita sintetizar y valorar los resultados obtenidos del experimento.

En el caso de los laboratorios LEDFE y Microelectronics WebLab, los experimentos están orientados a la caracterización de componentes, por lo que se tratan de experimentos sencillos en los que los alumnos únicamente deben interpretar los resultados. En el caso del ELVIS iLab, los circuitos también se encuentran predefinidos por el profesor, pero al igual que en el ISILab puede albergar circuitos más complejos como por el ejemplo el tercero de la Figura 2.33.

A modo de resumen, la Tabla 2.10 muestra los tipos de experimentos que se pueden llevar a cabo en cada una de las plataformas, tanto desde el punto de vista del control del experimento, como del nivel de complejidad de

los experimentos que son posibles realizar. La solución más cercana a la realidad es aquella en la que el alumno puede realizar las mismas acciones que el laboratorio real, es decir, realizar actividades interactivas, y disponer de experimentos de diferente complejidad. Aunque también es cierto que en ciertas ocasiones es preferible proporcionar a los alumnos los circuitos objetos de análisis predefinidos, y que se puedan centrar únicamente en el estudio de su funcionamiento, tal y como enunciábamos en el primer capítulo de este trabajo de investigación.

Por tanto, una solución completa debería permitir a los alumnos implementar sus propios circuitos partiendo de componentes discretos, cometiendo errores para evaluarlos y aprender de ellos, pero también debería proporcionar alternativas para que el profesor proporcione circuitos de diferente dificultad previamente definidos y centrar al alumno únicamente en el funcionamiento real del experimento.

Además, si el mismo laboratorio puede disponer desde circuitos destinados a usuarios que se inician en la electrónica hasta circuitos más complejos para usuarios avanzados, el alumno podrá usar de manera continuada durante el proceso de aprendizaje una misma herramienta que es capaz de adaptarse a las necesidades concretas, dependiendo del estadio de conocimiento en el que se encuentre.

Plataforma	Tipo de experimentos	Complejidad de los experimentos
NetLab	Interactivos	Básica –Media
RemotElectLab	Lotes	Media-Alta
ISILab	Interactivos	Todo tipo. Depende del profesor
VISIR	Interactivos por la rapidez en que se ejecutan por lotes	Todo tipo. Depende del profesor
LEDFE	Lotes	Básica
iLab	Lotes aunque ISA permite interactivos	Depende del laboratorio

Tabla 2.10: Tipo de experimentos de cada plataforma y complejidad de los mismos

2.3.3 Acciones posibles con cada método de control

Si bien un laboratorio remoto no tiene porqué sustituir al laboratorio presencial, aunque en algunas situaciones puede que sea la única alternativa para observar la ejecución de un experimento, sí debería proporcionar mecanismos para reproducir, en la medida de lo posible, las acciones que el alumno realizaría sobre el mismo experimento en las instalaciones de la universidad.

Si nos centramos en un experimento de electrónica analógica, una vez que el alumno ha adquirido ciertos conocimientos teóricos, puede comenzar las prácticas con los instrumentos y los componentes del laboratorio. En un sesión tipo, al alumno se le asigna un ejercicio que consiste en un circuito electrónico que debe implementar utilizando para ello los componentes discretos que el profesor le proporciona (resistencias, condensadores, transistores, etc.) y una vez construido, realizar medidas sobre él para comprobar su funcionamiento usando los instrumentos del laboratorio, y así contestar a las preguntas formuladas por el profesor en el guión de la práctica. Igualmente, el profesor puede pedir al alumno que realice modificaciones sobre el circuito, ya sea mediante cambios en los componentes o en las señales de excitación, y anotar también los resultados del experimento. Al finalizar, el alumno entrega un informe al profesor con los datos obtenidos y sus razonamientos ante las preguntas formuladas sobre el circuito bajo prueba.

Recordando la Figura 1.8, en donde se describían los pasos que el alumno debe seguir para llevar a cabo el experimento, podemos advertir como todos ellos implican bien realizar modificaciones sobre el experimento o bien obtener información del mismo, acciones que en un laboratorio remoto el alumno realizará a través del cliente y que dependerán directamente del método de control empleado en el diseño y desarrollo del laboratorio.

En el caso del laboratorio NetLab, tanto los componentes como las conexiones que el alumno puede realizar en el circuito bajo prueba están limitados. Es decir, el alumno no dispone de una gran variedad de componentes disponibles y no es posible realizar todas aquellas conexiones que sí serían posibles de realizar en un laboratorio manual. Respecto a los instrumentos, el alumno puede seleccionar varios instrumentos en el

laboratorio así como realizar ajustes sobre los mismos. En este laboratorio la adquisición de datos es manual, es decir, el alumno observa los resultados en la pantalla del ordenador y anota los resultados manualmente.

En la plataforma RemotElectLab al estar el circuito predefinido por el profesor, el alumno no tiene posibilidad de seleccionar los componentes empleados aunque en algunas ramas del circuito bajo prueba sí tiene la posibilidad de elegir entre un componente en otro. El alumno en este caso tampoco tiene la posibilidad de seleccionar diferentes instrumentos, sino que la plataforma únicamente ofrece medidas de tensión e intensidad en ciertas ramas del circuito, sin opción alguna a realizar ajustes en las medidas. Al igual que en el laboratorio anterior, la adquisición de datos es manual.

En el caso de los experimentos contenidos en el laboratorio ISILab, todos están preconstruidos por el profesor, por lo que el alumno no puede realizar ninguna selección o modificación respecto a los componentes o las conexiones ya implementadas. Los instrumentos también están predefinidos por el laboratorio, si bien sí pueden ser ajustados por el alumno. Al igual que en los laboratorios anteriores, la adquisición de datos es manual por parte del alumno.

En el laboratorio VISIR, los componentes a utilizar en el circuito bajo prueba no están limitados, aunque su uso depende de la capacidad de la matriz de conmutación empleada: a mayor número de tarjetas de componentes en la matriz, más componentes disponibles. En cambio, las conexiones están limitadas ya que para que un alumno sí pueda implementar una conexión, ésta debe estar realizada físicamente en la matriz. Respecto a los instrumentos, el alumno puede seleccionar libremente los equipos a utilizar, pudiendo elegir incluso el panel frontal de cada uno de ellos y realizar sobre ellos los ajustes que considere oportunos según las especificaciones del circuito o la información a visualizar. En cambio, la recogida de información obtenida en la experimentación, sigue siendo de forma manual.

Tanto en los laboratorios LEDFE como en aquellos analizados pertenecientes a la plataforma iLab, el alumno no puede realizar ninguna selección en cuanto a los componentes, conexiones o instrumentos a emplear, si bien sí puede ajustar los equipos empleados. Ambas plataformas disponen de herramientas para que el alumno exporte los datos obtenidos durante la

experimentación de manera electrónica, en formato Excel, que posteriormente puede ser tratada y modificada por el alumno para llevar a cabo sus explicaciones y responder a las preguntas formuladas en la práctica en cuestión.

A tenor del análisis realizado, se advierte cómo la capacidad de selección de componentes o conexiones en los circuitos bajo prueba, depende del sistema de control de experimentos empleado, en concreto de la configuración y capacidad de la matriz de conmutación. La selección de instrumentos dependerá de la configuración que se haga de los mismos en el sistema de control, al igual que la posibilidad de realizar ajustes sobre los mismos, acciones que dependerán tanto de la aplicación de control como del interfaz de comunicaciones empleado.

A modo de resumen, la Tabla 2.11 nos permite observar los resultados del análisis comparativo realizado en esta sección en base a las acciones que el alumno puede realizar sobre los laboratorios tomados como referencia en el estudio del estado del arte.

Plataforma	Selección de componentes	Selección de conexiones	Selección de instrumentos	Ajuste de instrumentos	Adquisición de datos
NetLab	Sí, limitados	Sí, limitados	Sí	Sí	Manual
RemotElectLab	No	Sí, limitados	No	No	Manual
ISILab	No	No	No	Sí	Manual
VISIR	Sí	Sí, limitados	Sí	Sí	Manual
LEDFE	No	No	No	Sí	Sí
iLab	No	No	No	Sí	Sí

Tabla 2.11: Acciones habilitadas en los métodos de control analizados

2.3.4 Sistema de control empleado en cada método de control

Tal y como decíamos al inicio de la sección 2.1, la selección de las tecnologías empleadas en el sistema de control de los experimentos es un elemento clave en el diseño del método de control de un laboratorio remoto. El elemento principal de este sistema es la matriz de conmutación, ya que determina el modo en el que se añaden y seleccionan los distintos experimentos del laboratorio. Su configuración también restringirá el tipo de circuitos que los usuarios podrán realizar, circuitos predefinidos en los que únicamente podrá observar su comportamiento o circuitos construidos por ellos mismos partiendo de las especificaciones del profesor. En cualquiera de las dos situaciones, el sistema de control deberá ofrecer fiabilidad y precisión en las medidas adquiridas, de modo que sean las mismas que las obtenidas en un laboratorio manual.

En el caso del laboratorio NetLab, la matriz de conmutación seleccionada es una matriz de conmutación comercial formada mediante la combinación de varios instrumentos, el mainframe VXI E8408A junto con una matriz de conmutación de formato 16x16 E1465A, ambos equipos de Agilent Technologies. La matriz de conmutación incluida en el mainframe es de doble cara, lo que permite duplicar el número de componentes disponibles para el usuario. Si se utiliza únicamente una cara, el número de componentes de dos conexiones (resistencias, condensadores, bobinas, etc.) disponibles será de 8, ya que cada componente necesita de dos conexiones para formar parte del circuito.

Quizás el número de componentes por cada matriz no sea elevado, pero en este caso, la facilidad para ampliar el número de componentes la proporciona: a) el mainframe E8408A, que dispone de 4 slots para albergar matrices de conmutación y b) el tipo de matrices de conmutación que se pueden conectar en este equipo, ya que existen soluciones basadas en formato 16x16, 4x16, 8x32, 8x8 y 4x4. Por consiguiente incluir nuevos componentes solo requiere añadir más matrices de conmutación al mainframe, ya que si se eligen matrices de conmutación del mismo fabricante y familia, la lógica de control es la misma.

En el RemotElectLab si se quiere añadir un nuevo experimento, se debe primero cablear el circuito en la placa de prototipos y posteriormente conectar a la matriz de conmutación diseñada específicamente para este

laboratorio los puntos del circuito sobre los que se va a permitir realizar las medidas. De este modo, proporcionar nuevos circuitos al laboratorio implica crear un nuevo circuito y configurar de nuevo los puntos de medida conectados a los módulos o matriz de conmutación. En este laboratorio, todos los instrumentos se encuentran incluidos en la plataforma NI-Elvis, luego la añadir nuevos equipos o modificar los existentes implicaría modificar el sistema de control para adaptarse a los nuevos cambios.

Para la plataforma ISILab se ha desarrollado una placa de base en la que se pueden acoplar hasta 17 experimentos. El conector en el cual se inserta la placa que contiene el experimento ya está configurado con los puntos de test disponibles. Gracias a la metodología de descripción de cada experimento basada en XML y a que todos los instrumentos a excepción de la placa base que contiene los experimentos son comerciales y están controlados utilizando drivers IVI, añadir un experimento es relativamente sencillo. Estas dos características hacen posible que sustituir los equipos por otros nuevos sea relativamente sencillo y rápido. El único equipo desarrollado a medida ha sido la placa donde se conectan los experimentos, pero que gracias al protocolo de configuración diseñado, añadir nuevos circuitos también es una tarea sencilla.

En el laboratorio VISIR toda la instrumentación se encuentra embebida en un PXI, por lo que la sustitución de un instrumento implica la adopción de este estándar. Aún así como veíamos anteriormente, la aplicación de control diseñada utiliza los drivers específicos de cada instrumento proporcionados por National Instruments, por lo que la adopción de nuevos instrumentos requerirá la modificación y actualización del software de control. En este laboratorio la matriz de conmutación está basada en una solución propia basada en relés y microcontroladores, por lo que tanto su diseño hardware como su control software es totalmente único y propietario, lo que complica que la ampliación de la matriz esté basada en la ya plataforma ya existente o bien adoptar una solución nueva.

Cada una de las tarjetas de componentes de las que consta esta matriz de conmutación puede albergar hasta un máximo de 10 componentes discretos de 2 conexiones, número depende en función de si se usan componentes de más conexiones. El número de tarjetas de componentes puede ser prácticamente ilimitado, aunque a medida que aumenta este

número la complejidad del cableado y de la definición de los ficheros de configuración aumenta considerablemente, al introducir más componentes y variables en el sistema. En cambio, una de las ventajas de este sistema es que permite ser utilizado tanto con componentes discretos como con circuitos ya prediseñados, sin necesidad de realizar ningún cambio ya sea en el hardware o el software de control de la matriz de conmutación.

En el Microelectronics WebLab del iLab la capacidad para añadir nuevos componentes está determinada por el matriz de conmutación utilizada (Agilent E5250A), la cual puede albergar diferentes tarjetas de matrices (E5252A y E5255A). El número de componentes o experimentos depende por tanto de las tarjetas utilizadas. En el caso del ELVIS iLab se ha utilizado la matriz de conmutación SCXI-1169, la cual dispone de 100 relés simples, lo que indica que se podrían realizar circuitos que dispusieran de hasta 100 nodos, pero esta configuración depende del número de componentes utilizados. La arquitectura ISA del iLab permite diferentes tipos de instrumentos, por lo que la adaptación a nuevos equipos en un laboratorio dependerá de la solución adoptada en cada caso. En el Microelectronics Weblab se trata de instrumentos independientes que pueden ser adquiridos al fabricante, incluida la matriz de conmutación utilizada. En este caso se deberá actualizar el software de control de los equipos, al igual que sucede en el ELVIS iLab donde la plataforma hardware elegida está basada en el NI-Elvis y se utiliza la matriz de conmutación SCXI-1169 para la selección de los componentes o experimentos.

En el caso del laboratorio LEDFE, el número de componentes o experimentos que se pueden añadir está limitado por el número de líneas de control de la tarjeta de adquisición de datos utilizadas. En este caso solamente se están usando tres líneas digitales para la selección del dispositivo a caracterizar, por lo que el número máximo de componentes o experimentos disponibles es de ocho ($2^3=8$). Querer añadir más experimentos implicaría utilizar más líneas de control, lo que necesitaría readaptar el código de control. En este laboratorio no se utilizan instrumentos convencionales sino una tarjeta de adquisición de datos, lo que complicaría la sustitución de este equipo por instrumentos convencionales. Para la selección del experimento bajo test, también se utiliza la misma tarjeta de adquisición de datos por lo que se trata de una solución propietaria.

Plataforma	Tipo matriz de conmutación	Potencia de la matriz	Tipo de circuitos	Nivel escalabilidad	Puntos medida
NetLab	Comercial	Baja	Componentes discretos pero con número limitado	Normal	Libres
RemotElectLab	Propia	Alta	Circuitos predefinidos con posibilidad de modificación	Fácil	Fijos
ISILab	Propia	Alta	Circuitos predefinidos	Fácil	Fijos
VISIR	Propia	Media	Circuitos con componentes discretos o circuitos predefinidos	Normal	Libres
LEDFE	Propia	Media	Circuitos predefinidos sin posibilidad de modificación	Difícil	Fijos
iLab	Comercial	Alta	Circuitos predefinidos sin posibilidad de modificación	Fácil	Fijos

Tabla 2.12: Sistema de control de experimentos empleados y características

A modo de resumen, la Tabla 2.12 muestra las características derivadas de la selección de uno u otro tipo de configuración de la matriz de conmutación empleada en el sistema de control de los experimentos del laboratorio. Se puede observar en esta tabla como no existe una relación directa entre el tipo de matriz empleada (comercial o propia) con su potencia a la hora de realizar circuitos desde cero basados en componentes discretos y de disponer de puntos medida sobre ellos fijos o libres.

Por tanto podemos concluir esta sección diciendo que las características del sistema de control de experimentos no dependen directamente de la adopción de un sistema comercial o desarrollado a medida, sino de la configuración que se haga del mismo.

Para ello tal y como venimos repitiendo a lo largo de este trabajo, es necesario partir de un método de control diseñado a priori para cumplir con aquellas características que queremos que satisfaga nuestro laboratorio y no tomando como punto de partida los experimentos a los cuales se les añaden sucesivas tecnologías y sistemas para su control remoto, que hagan posteriormente difícil su adaptación a nuevos experimentos, ya sea desde el punto de vista del circuito bajo prueba o de los instrumentos utilizados durante las fases de testeo y validación.

2.4 Conclusiones

Un primer punto muy importante a tener en cuenta es que las plataformas analizadas NetLab, RemotElectLab, ISILab, VISIR, LEDFE y los laboratorios pertenecientes al iLab, Microelectronics WebLab y ELVIS iLab son herramientas de aprendizaje que están siendo utilizadas actualmente por los alumnos de las universidades en que han sido desarrolladas, así como en universidades colaboradoras, como es el caso del VISIR desplegado también en la Universidad de Deusto, o la Universidad Nacional de Educación a Distancia (UNED), entre otras o el iLab desplegado en Africa, Australia, China y Europa⁸. Es decir, son desarrollos que en su día comenzaron siendo trabajos de investigación y que a día de hoy se han convertido en herramientas de aprendizaje utilizadas por miles de alumnos de todo el mundo.

Tal y como se indicaba en secciones anteriores, estas plataformas tienen un objetivo común: aportar al alumno nuevas y mejores herramientas para la experimentación remota en el área de la electrónica analógica. Para ello, cada plataforma ha desarrollado un método de control acorde a sus necesidades y objetivos específicos. Por esta razón no se puede afirmar rotundamente qué método es mejor, pues cada uno trata de resolver problemas distintos en función de las necesidades de los experimentos que se quieren proporcionar y de las acciones que se quieren habilitar al alumno.

Aún así, se pueden destacar los siguientes puntos fuertes de cada una de las metodologías empleadas:

- NetLab: cabe remarcar la libertad que se le presenta al usuario para construir circuitos utilizando los componentes proporcionados por la plataforma y conectándolos entre sí en base a unas especificaciones dadas. Además, todos los equipos empleados en el método de control son comerciales, lo que haría posible su adaptación a nuevos instrumentos, pero con la necesidad de realizar modificaciones en la aplicación de control.
- RemotElectLab: es una plataforma completamente integrada y portable en su conjunto al basarse en la solución proporcionada por

⁸ Más información en: https://wikis.mit.edu/confluence/display/ILAB2/ilab_partners

National Instruments y permitir su acceso a través de un sistema de gestión del aprendizaje muy extendido actualmente, como es Moodle.

- ISILab: la modularidad mediante la que se ha desarrollado tanto su hardware como su software hacen posible que añadir nuevos experimentos o equipos sea realmente sencillo, sin necesidad de tener un amplio conocimiento de la arquitectura software o hardware del sistema. Además, su aplicación de gestión ha sido desarrollada utilizando drivers IVI lo que hace facilita aún más su adaptación a nuevos instrumentos.
- VISIR: es quizás la plataforma que más se asemeja a lo que un alumno se encuentra en un laboratorio real ya que puede realizar exactamente las mismas acciones, desde la configuración de los instrumentos a la construcción de circuitos o el testeo de experimentos previamente configurados.
- LEDFE: se trata de un entorno de experimentación basado en la sencillez, y destinado a alumnos que se introducen en el mundo de la electrónica y a los que se les ofrecen experimentos básicos para que comprendan el funcionamiento de componentes utilizados en circuitos electrónicos.
- iLab: ha definido la arquitectura ISA que facilita que cualquier laboratorio, independientemente del tipo y materia en el que esté enfocado, pueda ser accedido de manera remota.

De igual manera, los principales puntos débiles que se pueden citar de cada una de estas plataformas son:

- NetLab: la capacidad de añadir nuevos instrumentos o componentes es limitada debido a que la metodología de control elegida se ciñe exclusivamente a los equipos y fabricantes utilizados. Ampliar la capacidad de la matriz de conmutación implica diseñar un hardware especial que lo haga posible.
- RemotElectLab: su punto fuerte también es su punto débil. Al tratarse de una plataforma basada en la solución integral NI-Elvis, es complicado sustituir o incluir nuevos instrumentos. Además, añadir nuevos experimentos requiere la reconfiguración tanto del hardware como del software de control.
- ISILab: en este laboratorio el alumno es un mero espectador en el que observa una serie de salidas en función de las diferentes entradas que

puede seleccionar. Es decir mediante el método de control empleado, el alumno tiene poca libertad para hacer modificaciones sobre el circuito y comparar y analizar los resultados obtenidos en cada situación.

- VISIR: se trata de una solución desarrollada únicamente para funcionar con equipos de National Instruments y la matriz de conmutación desarrollada en exclusiva para este laboratorio, la cual debido a sus características, presenta un alto precio. Adicionalmente, la integración de nuevos componentes o circuitos, requiere de un conocimiento en profundidad tanto de la matriz de conmutación como del sistema de control y gestión de la plataforma VISIR.
- LEDFE: es una plataforma sencilla y reducida a la que es difícil añadir nuevos experimentos, si no es mediante la sustitución de parte de su hardware. Además, no proporciona a los alumnos la oportunidad de utilizar los mismos instrumentos que en el laboratorio manual por lo que solo pueden observar los resultados presentados en el interfaz del cliente.
- iLab: los dos laboratorios analizados (Microelectronics WebLab y ELVIS iLab) permiten únicamente experimentar con circuitos ya predefinidos sobre los que los alumnos no pueden realizar ninguna modificación estructural. La práctica se limita a la configuración de las señales de alimentación y la observación de las salidas.

Una vez estudiadas cada una de las metodologías de control empleadas en las plataformas descritas, se pueden extraer las siguientes conclusiones:

- La metodología de control debe ser tal que haga posible que el alumno reproduzca de manera remota las mismas acciones que puede realizar en el laboratorio tradicional, tanto a la hora de construir o medir un experimento como al utilizar la instrumentación del laboratorio.
- Esta metodología debe proporcionar herramientas para hacer posible construir circuitos basados en componentes discretos, o bien realizar experimentos sobre circuitos previamente contruidos por el profesor de la asignatura.
- Desde el punto de vista del profesor de la asignatura como usuario del sistema, debe ser sencillo añadir nuevos componentes o nuevos circuitos bajo prueba predefinidos a los experimentos del laboratorio.

- Si la metodología de control de los experimentos se basa en estándares abiertos y en equipos comerciales y no en soluciones propietarias, sustituir un instrumento por otro, en función de las necesidades y del presupuesto, será sencillo y no requerirá modificaciones sustanciales en el sistema ni se verá sujeto a las especificidades concretas de un fabricante.
- Utilizar soluciones propietarias hace que posibles reconfiguraciones y adaptaciones sean complicadas, y que siempre se dependa por completo del equipo de desarrollo, por lo que el despliegue de la plataforma es complicado y requiere de conocimientos específicos tanto hardware como software.

En base a estas conclusiones y retomando los objetivos e hipótesis planteadas en el capítulo primero, el método de control buscado debería satisfacer simultáneamente todas las cuestiones presentadas en la Tabla 2.13, tal y como hemos indicado en la última fila de esta tabla. Si examinamos cada una de las soluciones analizadas en este capítulo, advertimos que todas ellas cumplen con algunos de estos requisitos, pero ninguno de los métodos ya desarrollados cumple satisfactoriamente con todos ellos.

Plataforma	Instrumentos Intercambiables	Fácil añadir experimentos	Uso de estándares	Independiente de instrumentos	Circuitos predefinidos y sin definir	Soluciones no propietarias
NetLab	No	Sí	Sí	No	Sin definir	Sí
RemotElectLab	No	No	Sí	No	Predefinidos	No
ISILab	Sí	Sí	Sí	Sí	Predefinidos	No
VISIR	No	No	Sí	No	Ambos	No
LEDFE	No	No	No	No	Predefinidos	No
iLab	No	Sí	Sí	No	Predefinidos	Sí
Método a desarrollar	Sí	Sí	Sí	Sí	Ambos	Sí

Tabla 2.13: Especificaciones a satisfacer por el método de control deseado

Cada una de las especificaciones indicadas en la Tabla 2.13 están pensadas para que el método de control solución que contemple todas ellas permita lograr los objetivos planteados al comienzo de este documento y así dar solución a la hipótesis principal de este trabajo de investigación. De este modo:

- Con instrumentos intercambiables hacemos referencia a la capacidad de la aplicación de control para ser utilizada con instrumentos diferentes a los que se utilizan en la primera versión del método de control. Si hacemos uso en esta aplicación de drivers tipo IVI, se pueden independizar estos algoritmos de los instrumentos de tal modo que podamos sustituirlos por otros de la misma familia sin alterar el software de control.
- La facilidad para añadir experimentos o circuitos bajo prueba radica en el método de control empleado para el control de los instrumentos, y principalmente en el sistema de matriz de conmutación empleado. Por ejemplo, en el caso del VISIR para añadir nuevos componentes es necesario comprender el diseño de la matriz utilizada, ya que disponer de nuevos componentes implica tener que realizar cableados específicos en la matriz de conmutación para que el alumno obtenga resultados válidos durante sus experimentos.
- Todos los métodos estudiados hacen uso de estándares para el control de los instrumentos y los experimentos, aunque en el LEDFE utiliza una solución basada en tarjeta de adquisición de datos para el control de los experimentos. Recordemos también que este laboratorio tampoco utiliza instrumentos para la adquisición de medidas.
- En lo referente a la independencia del método respecto a los instrumentos, vemos que únicamente el empleado en el laboratorio ISILab satisface este requisito, ya que es el único que emplea drivers IVI en su software de control, de modo que cualquier instrumento controlable mediante este tipo de drivers puede ser usado en el laboratorio. Este requisito está vinculado al intercambio de instrumentos, pero en este caso se trata de un matiz más amplio ya que ciertos fabricantes permiten que modelos diferentes de sus equipos puedan ser controlados mediante el mismo software de control. Así con este requisito queremos indicar también que el

método de control sea independiente incluso del fabricante del instrumento.

- Tal y como se ha analizado durante este capítulo, ciertos métodos contemplan que se pueda practicar únicamente o con circuitos predefinidos o con circuitos que deben ser construidos por el usuario. En este caso, únicamente el método de control empleado en el laboratorio VISIR es independiente del tipo de experimento, pudiendo ser utilizado el laboratorio en ambos casos.
- Finalmente, con soluciones propias se hace referencia a las matrices de conmutación utilizadas para el control de los experimentos. En la Tabla 2.12 se observa que algunas plataformas utilizan matrices diseñadas expresamente para un laboratorio específico y otras emplean matrices comerciales. Ahora bien, puede que la solución a desarrollar utilice matrices comerciales pero que sí incluyan cierto hardware para disponer de los componentes que forman parte de los experimentos. En este caso, generalmente este hardware únicamente es utilizado para conectar los componentes a la matriz por lo que no afecta sustancialmente al método de control

Por tanto, el método de control a desarrollar busca satisfacer todas estas especificaciones, lo que hará que sea un diseño único y diferente a los analizados a lo largo de este capítulo.

*Capítulo***3****Descripción del método de control propuesto**

El objetivo de este capítulo es describir el método de control presentado como alternativa a los analizados en el capítulo anterior. Para diseñar y desarrollar un método propio se parte de las conclusiones a las que hemos llegado una vez estudiadas tanto las tecnologías como las soluciones adoptadas en otros laboratorios remotos, así como de los objetivos marcados al inicio de este trabajo y de las motivaciones que dieron pie al mismo.

En las siguientes secciones vamos a analizar y describir las soluciones propuestas a los problemas identificados en cada uno de los tres sistemas que forman parte del método de control de un laboratorio remoto de electrónica: aplicación software de control, interfaz de comunicaciones y sistema de control de los experimentos.

Para ello el capítulo ha sido dividido en dos partes bien diferenciadas. En la primera, comprendida por la sección 3.1, se describen las soluciones a los problemas encontrados en estos tres sistemas. El objetivo de esta sección es proporcionar una respuesta sencilla y rápida sin entrar en detalles técnicos que pueden hacer que se pierda el objetivo y alcance de las soluciones propuestas. Esta primera parte concluye con un resumen de las alternativas

desarrolladas, así como de las aportaciones realizadas en el área de los métodos de control aplicados en laboratorios remotos de electrónica.

En la segunda parte del capítulo abordamos en profundidad los detalles técnicos tanto de la aplicación software como del sistema de control de experimentos propuestos. Gracias a la naturaleza del interfaz de control utilizado en este método, no es necesario realizar prácticamente ningún desarrollo, por lo que no se ha incluido una sección específica que lo describa.

Los desarrollos llevados a cabo para obtener el método de control propuesto conllevan ciertos procesos de configuración, que si bien no aportan ninguna novedad a las soluciones planteadas, son necesarios para el buen funcionamiento de las mismas. Por esta razón, parte de estas configuraciones, así como ejemplos de las mismas están descritos en apéndices anexos al final de este documento.

3.1 Descripción general del método de control propuesto

El objetivo principal del conjunto de soluciones implementadas en este trabajo de investigación es dar solución a los problemas que aparecen durante el desarrollo y despliegue de los sistemas destinados a controlar los instrumentos y experimentos de un laboratorio remoto de electrónica analógica (Figura 3.1).

La solución propuesta busca también satisfacer las especificaciones descritas al finalizar el capítulo anterior y que aparecen recogidas en la Tabla 2.13, ya que ninguno de los métodos de control analizados las cumple en su totalidad. Estos requisitos pueden ser resumidos de la siguiente manera:

- Emplear básicamente instrumentos comerciales y no soluciones propietarias.
- Permitir sustituir unos instrumentos por otros sin la necesidad de actualizar la aplicación de control, por lo que ésta debe ser independiente de los instrumentos y de los fabricantes de los mismos.
- Ser independiente de los circuitos bajo prueba, ya sean predefinidos o basados en los componentes y conexiones determinados por el propio alumno.

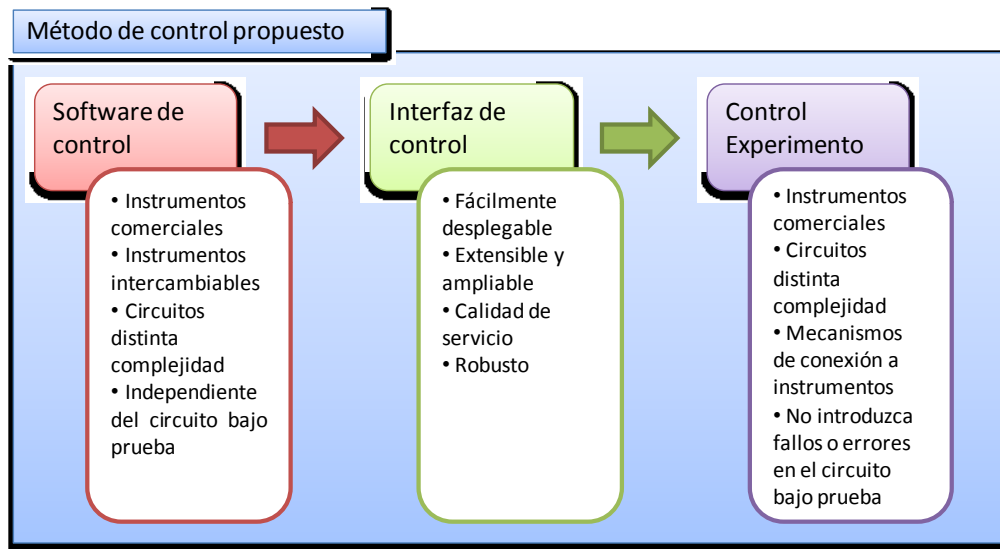


Figura 3.1: Problemáticas a atajar mediante el método de control propuesto

- Proporcionar un sistema de control y medida fiable y robusto de manera que no introduzca errores significativos en los resultados de la experimentación ni altere su funcionamiento, ofreciendo repetitividad en las medidas y velocidad en su respuesta.
- Por tanto debe ser un método de control de calidad y eficiente técnicamente hablando de manera que el laboratorio remoto que lo adopte pueda ser considerado como una herramienta educativa de calidad.
- Desde el punto de vista del profesor usuario del laboratorio, debe ser sencillo añadir nuevos componentes o circuitos bajo prueba a los experimentos contemplados en el laboratorio.

Tal y como se observa en la Figura 3.2, la arquitectura física diseñada está basada en los tres niveles o sistemas que conforman el método de control del hardware de un laboratorio remoto: aplicación de control, interfaz de comunicaciones y un sistema de control de los experimentos.

En el primer nivel se encuentran aquellos procesos y algoritmos que forman la aplicación encargada de monitorizar y controlar tanto los instrumentos como los experimentos del laboratorio. La estructura de este software está basada en la definida en el proyecto VISIR [ZGH07], en donde todos los procesos que debe ejecutar el servidor son distribuidos en dos entidades distintas, los denominados Servidor de Medidas y el Servidor de Equipos y Experimentos.

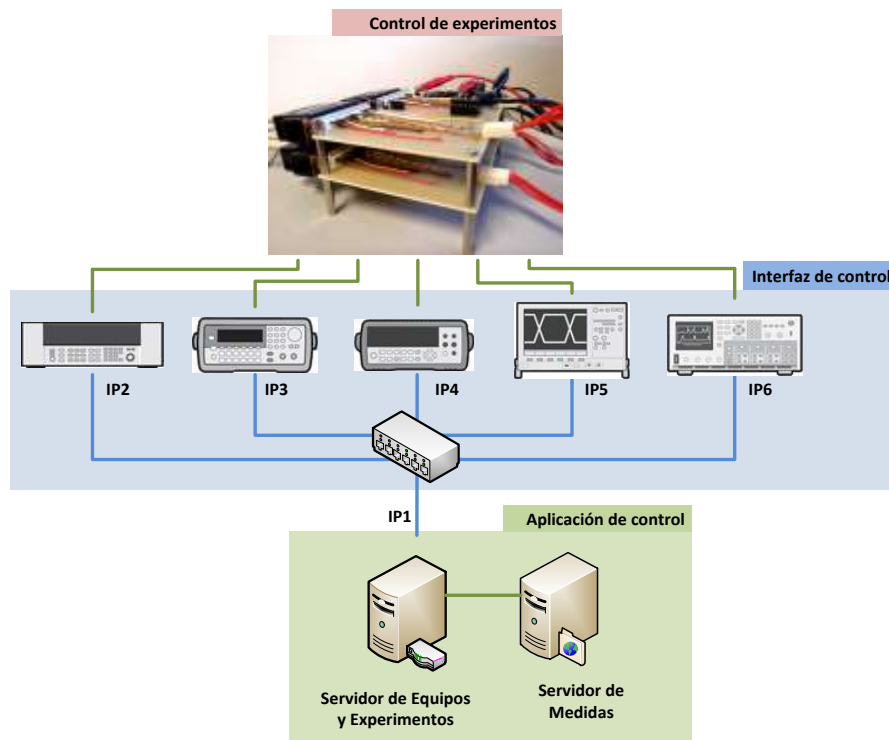


Figura 3.2: Arquitectura física del método propuesto

Recordemos que la necesidad de desarrollar un nuevo método de control surge de la participación en el proyecto VISIR, especialmente de los problemas relativos al Servidor de Equipos y Experimentos, por lo que la distribución de las funciones del servidor en dos entidades ha permitido centrar los desarrollos de este trabajo en éste servidor, manteniendo estables ciertas funciones del Servidor de Medidas.

De este modo se observa en la Figura 3.3 cómo en el Servidor de Medidas no se realiza ninguna modificación en las funciones destinadas a la gestión de usuarios, permisos y conexiones (como ya indicaba la Figura 1.6) y sí sobre las funciones relativas a la validación de peticiones realizadas sobre los instrumentos y los experimentos. En cambio, la lógica del Servidor de Equipos y Experimentos va a ser rediseñada y desarrollada totalmente, exceptuando la función dedicada a gestionar la conexión entre este servidor y el Servidor de Medidas. En definitiva, las funciones que quedan fuera del contexto experiencial de este trabajo no han sido modificadas respecto a los desarrollos realizados en el proyecto VISIR, mientras que el resto de funciones han sido completamente redefinidas y codificadas para adecuarse a las especificaciones del método propuesto.

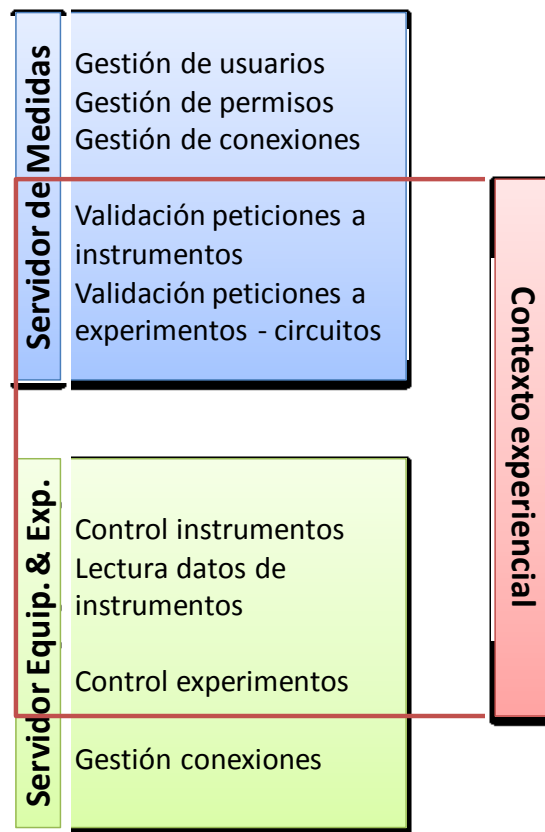


Figura 3.3: Funciones de la aplicación de control contempladas en el método de control propuesto

El segundo nivel de esta arquitectura contempla las comunicaciones con los instrumentos del laboratorio, es decir integra el interfaz de control empleado para conectar la aplicación de gestión con los instrumentos y experimentos del laboratorio. Como describiremos más adelante y en base al estudio de las tecnologías disponibles realizado en el capítulo anterior, el interfaz elegido es el estándar LXI. Mediante este interfaz se ha creado una red de instrumentos de laboratorio basada en Ethernet, en la que cada uno dispone de una dirección IP y está gestionado y controlado mediante la aplicación software definida en el primer nivel de esta arquitectura, la cual reside y se ejecuta en un ordenador que también forma parte de dicha red.

En el tercer nivel se emplaza el sistema de control de los experimentos del laboratorio, el cual está basado en el uso de una matriz de conmutación comercial y de un sencillo hardware diseñado y desarrollado para emplazar los componentes o circuitos prediseñados que forman parte de los experimentos del laboratorio, y en donde se conectan también los

instrumentos empleados para realizar las pruebas y medidas que los ejercicios prácticos requieran.

De manera esquemática y en base al estudio de las tecnologías disponibles realizado en el capítulo anterior, la Tabla 3.1 muestra cuáles han sido las tecnologías empleadas en cada uno de los niveles o sistemas que forman parte de la arquitectura del sistema de control propuesto.

3.1.1 Soluciones planteadas a los problemas descritos

Mediante el empleo y combinación de las tecnologías seleccionadas para el desarrollo de la aplicación de control, el interfaz de comunicaciones y el sistema de control de experimentos, se ha dado solución a los problemas encontrados durante el estudio del estado del arte llevado a cabo en el capítulo anterior.

En esta sección detallaremos cómo se han alcanzado dichas soluciones, así como las técnicas implementadas para satisfacer los objetivos y especificaciones impuestas al comienzo de este trabajo. Estas descripciones no entran en detalle en aspectos técnicos, los cuales serán tratados en secciones posteriores.

3.1.1.1 Soluciones desarrolladas para la aplicación de control

Los problemas asociados a la aplicación de control a resolver mediante el método de control propuesto son principalmente dos:

- a) Obtener una aplicación que pueda ser empleada con diferentes instrumentos, independientemente de su fabricante, de modo que su adaptación a nuevos equipos sea rápida, sencilla y escalable.
- b) Disponer de una aplicación de control independiente de los circuitos bajo prueba que contengan los experimentos del laboratorio remoto.

Aplicación de control	Drivers de control	Interfaz de control	Control de experimentos
LabVIEW	IVI	LXI	Matriz de conmutación comercial y módulo de conexionado

Tabla 3.1: Tecnologías empleadas en el método de control propuesto.

Recordemos que la aplicación de control reúne el conjunto de algoritmos distribuidos en la arquitectura diseñada entre el Servidor de Medidas y el Servidor de Equipos y Experimentos encargados de gestionar y controlar la información enviada hacia y desde los instrumentos y experimentos del laboratorio.

Por tanto, las metodologías y tecnologías empleadas para su diseño y desarrollo determinarán qué tipos de equipos y de experimentos podrán emplearse en el laboratorio, así como las futuras adaptaciones y modificaciones que se requieran realizar para adecuar el laboratorio bien a nuevos instrumentos o bien a los nuevos circuitos bajo prueba contenidos en los experimentos. A continuación describiremos los métodos y técnicas empleadas para dar solución a ambos problemas.

A) Aplicación de control independiente de los instrumentos del laboratorio

Como se observa en la Figura 3.4, la función principal del conjunto de algoritmos que forman parte de la aplicación de control de un laboratorio remoto de electrónica analógica es controlar los experimentos en función de las órdenes enviadas por el usuario a través de un cliente web, y devolverle la respuesta generada por el circuito bajo prueba y monitorizada mediante los instrumentos dispuestos en el experimento.

Tal y como se muestra en la Figura 3.4 y siguiendo el esquema descrito en la Figura 1.6, el desarrollo tanto del cliente como de algunas funciones del servidor quedan fuera del contexto experiencial de este trabajo. Aún así, debemos diseñar un método de control que una vez desarrollado pueda funcionar en un laboratorio remoto real, por lo que tomando como referencia la experiencia adquirida en el proyecto VISIR utilizaremos parte de sus sistemas para comprobar posteriormente el funcionamiento de la solución diseñada.

Entre los elementos del proyecto VISIR empleados para validar el método propuesto será su cliente web, principalmente debido a su alta potencia, rendimiento y usabilidad desde el punto de vista de los alumnos [GZO04], ya que hace posible que éstos puedan realizar prácticamente las mismas acciones que si se encontraran ante los equipos reales de un laboratorio manual.

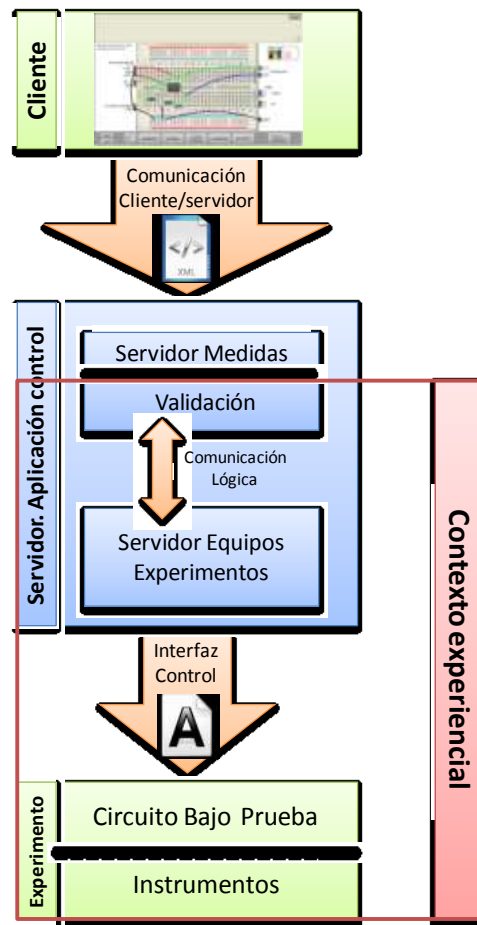


Figura 3.4: Contexto de la aplicación de control en el método desarrollado

Por estas razones no se ha realizado ninguna modificación en el cliente, al igual que ha sucedido con el protocolo de comunicación basado en el estándar XML establecido entre el cliente y el Servidor de Medidas, el cual ha permanecido inalterado respecto al diseño original del proyecto VISIR.

Pero mantener tanto el cliente como este protocolo de comunicación, junto con los problemas que se han querido atajar durante el desarrollo de la aplicación de control del método propuesto, ha requerido realizar ciertas modificaciones en el Servidor de Medidas, el cual aparte de encargarse principalmente de tareas de gestión de los recursos del laboratorio (usuarios, permisos, conexiones), también realiza tareas de validación de los datos enviados a los instrumentos con objeto de su configuración.

Para ello hemos adaptado todos los algoritmos de validación de datos que se ejecutan en el Servidor de Medidas y que una vez validados son enviados al Servidor de Equipos y Experimentos para la configuración de los

instrumentos del laboratorio. Gracias a esto se ha hecho posible que todas las órdenes que sean introducidas por el usuario a través del cliente web puedan ejecutarse sobre cualquier instrumento, independientemente de su tipo y fabricante. Estas validaciones se han realizado en base a los algoritmos de control que residen en el Servidor de Equipos y Experimentos y que controlan la ejecución de comandos sobre los instrumentos y su posterior lectura de medidas.

Como se indica en la Figura 3.4, la comunicación existente entre el Servidor de Medidas y el Servidor de Equipos y Experimentos, se basa en una comunicación lógica implementada sobre un socket. Al igual que las validaciones de los datos que se envían entre ambas entidades, esta comunicación ha sido diseñada y desarrollada completamente para posibilitar que los algoritmos de control encargados de controlar cada tipo de instrumento y que se ejecutan en el Servidor de Equipos y Experimentos, puedan controlar instrumentos de cualquier proveedor. Tanto los procesos de validación como el protocolo de comunicación entre ambos servidores aparecen detallados técnicamente en posteriores secciones de este documento.

Una vez que la información de configuración de los instrumentos ha llegado al Servidor de Equipos y Experimentos, el diseño modular con el que se han realizado los algoritmos de control que forman la aplicación principal de este servidor permite que añadir algoritmos para controlar instrumentos que no hayamos contemplado aún en el laboratorio (por ejemplo un analizador dinámico de señal) sea fácil y no influya en los algoritmos ya desarrollados.

Los algoritmos de control diseñados y desarrollados para el método de control que proponemos utilizan drivers IVI para comunicarse con las funciones de control de los instrumentos. Como explicábamos, los drivers son funciones software que permiten a las aplicaciones comunicarse y controlar los instrumentos al acceder a los recursos que ofrece para su control remoto el firmware del propio instrumento. En el capítulo anterior, se analizaban las alternativas existentes al uso de drivers IVI (*comandos SCPI, drivers VXIPlug&play*), pero principalmente, hemos seleccionado esta tecnología por las siguientes razones, las cuales nos permiten dar solución a parte de los problemas que se quieren solventar:

- Por una parte permiten diseñar y desarrollar algoritmos orientados a controlar familias de instrumentos y no un instrumento en particular. Así, el algoritmo creado para controlar un generador de funciones sirve para controlar cualquier generador de funciones que cumpla con este estándar. Igualmente, como en el Servidor de Medidas se ha validado que los datos de configuración enviados desde el cliente están dentro de los límites especificados por el estándar, nos aseguramos que cualquier instrumento de una familia concreta podrá ser controlado mediante estos algoritmos.
- Al haber sido el estándar IVI adoptado por los principales fabricantes de instrumentos del mercado⁹, mediante los algoritmos de control desarrollados podemos controlar una gran cantidad de instrumentos sin preocuparnos del fabricante de los mismos. De este modo, si queremos sustituir en el laboratorio un instrumento por otro de un fabricante diferente porque se adapta mejor a los requisitos del laboratorio, no se tiene que modificar ninguna línea de control. Gracias a la estructura de configuración del Servidor de Equipos y Experimento adoptada y que es ofrecida por el estándar IVI, únicamente tenemos que instalar el driver IVI específico del instrumento y realizar una sencilla configuración del mismo.
- Además, el hecho de que los drivers IVI empleen la arquitectura VISA permite independizar la aplicación software de control del interfaz de comunicaciones, que, si bien no era un objetivo inicial de este trabajo, permite que la solución LXI empleada pueda ser sustituida por otra en función de las necesidades y disponibilidad de equipos en el momento del despliegue del método de control aquí propuesto.

⁹ En <http://www.ivifoundation.org/> podemos encontrar las referencias a todos los fabricantes pertenecientes a este consorcio.

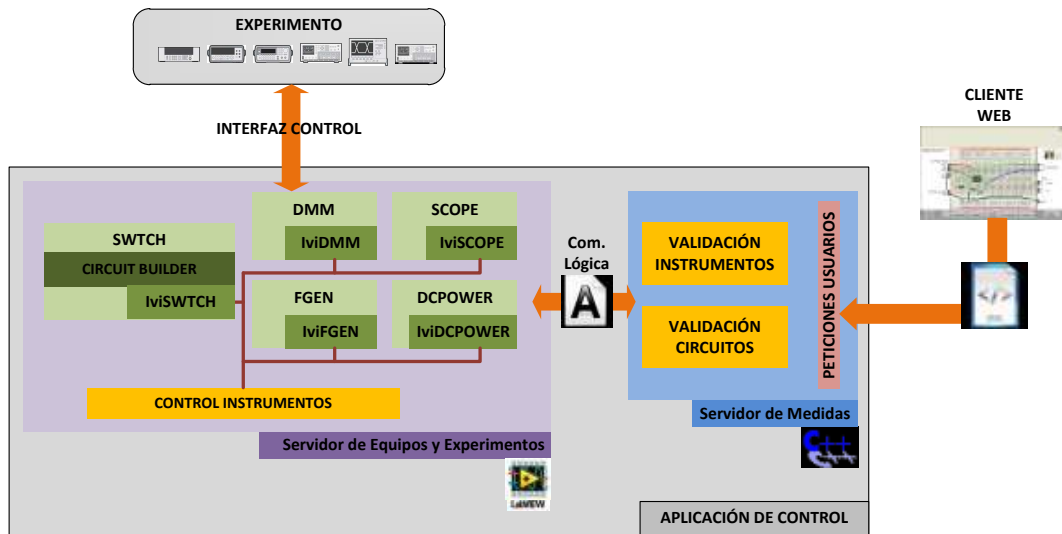


Figura 3.5: Arquitectura lógica de la aplicación de control de la solución propuesta.

De manera gráfica la Figura 3.5 muestra el flujo información así como los bloques funcionales de cada uno de los servidores que forman la aplicación de control y que intervienen en la solución propuesta. Mediante la combinación de estos procesos y la configuración que se hace tanto del intercambio de información como de los algoritmos de control contenidos en la aplicación distribuida tanto en el Servidor de Medidas como en el Servidor de Equipos y Experimentos, se consigue dar solución a los problemas planteados durante el diseño del software de control respecto a la escalabilidad, actualización e independencia del mismo respecto a los instrumentos del laboratorio, tal y como aparece resumido en la Tabla 3.2.

Problema	Solución
P1. Escalabilidad respecto a nuevos instrumentos	Diseño modular de la aplicación de control
P2. Actualización o sustitución de instrumentos por otros.	Desarrollado algoritmos de control basados en familias de instrumentos, no en instrumentos específicos
P3. Independencia del fabricante de los instrumentos	Los algoritmos de control codificados utilizan drivers IVI, independientes de los fabricantes

Tabla 3.2: Soluciones para independizar la aplicación de control de los instrumentos

En definitiva, la combinación de una aplicación de control modular desarrollada en LabVIEW, que hace uso de las capacidades y cualidades de los drivers IVI para el control tanto de los instrumentos como de los circuitos y que analizaremos a continuación, es una solución novedosa para su uso en laboratorios remotos de electrónica analógica.

B) Aplicación de control independiente de los circuitos bajo prueba del laboratorio

Si bien las soluciones anteriores plantean una alternativa para el control de los instrumentos que forman parte de un experimento, a continuación vamos a describir las técnicas propuestas para que la misma aplicación de control pueda ser empleada para que un alumno pueda trabajar remotamente sobre diferentes tipos de circuitos, independientemente de su complejidad y de si se tratan de circuitos preconstruidos o no.

En este punto debemos recordar que mediante el método de control propuesto, el usuario podrá llevar a cabo una experimentación interactiva, obteniendo respuesta a sus acciones una vez que sean ejecutadas sobre el experimento, siendo el tiempo de respuesta dependiente del número de peticiones enviadas al laboratorio.

Para lograr que el método de control sea independiente del tipo de circuitos bajo prueba, se propone una solución dividida en una parte software, que analizaremos en este punto, en combinación con una solución hardware que explicaremos en la sección 3.1.1.3, en la cual estudiamos las técnicas adoptadas para el control de los experimentos.

El control software del experimento está formado por el conjunto de algoritmos desarrollados en el Servidor de Medidas y en el Servidor de Equipos y Experimentos que permiten que el circuito con el que el alumno está practicando en el cliente web sea construido y testeado mediante los instrumentos del laboratorio, utilizando prácticamente los mismos recursos y con las mismas garantías de obtener resultados válidos que si se encontrara en el laboratorio tradicional. Es decir, que haga uso de los mismos equipos y que la respuesta ofrecida por el laboratorio remoto sea lo más parecida posible a la obtenida en el laboratorio manual.

Para ello, al igual que en caso del control de los instrumentos, se parte de la información enviada desde el cliente web que contiene la configuración

del experimento relativa a los componentes utilizados, las conexiones realizadas entre ellos y los puntos del circuito así creado en donde se conectan los instrumentos para su alimentación y toma de medidas.

Una vez recibida esta información en el Servidor de Medidas se ha desarrollado un algoritmo específico denominado *Circuit Solver*, el cual desempeña un papel fundamental en el proceso de configuración e implementación física de un experimento: validar si el circuito creado por el usuario formado por los componentes, conexiones e instrumentos se puede crear físicamente en base a unas especificaciones indicadas por lo que denominamos instructor virtual e implementado por dicho algoritmo.

Al igual que en el laboratorio tradicional en el cual el profesor proporciona unos componentes determinados e indica qué circuito se va a realizar y dónde se va a alimentar y tomar medidas, mediante este método se ofrecen estas mismas restricciones en el laboratorio remoto. Igualmente, este instructor virtual tiene como objetivo evitar daños físicos en el hardware del laboratorio (evitando por ejemplo realizar cortocircuitos), pero siempre dejando cierta libertad en el alumno para cometer fallos y que él mismo busque la solución a los mismos.

Este algoritmo *Circuit Solver* se ha diseñado acorde con el hardware utilizado para controlar los experimentos, ya que es totalmente diferente al diseñado en el proyecto VISIR. Mediante la solución aportada, la definición de los ficheros que alimentan este algoritmo y por ende al instructor virtual, son más simples y sencillos, agilizando la tarea de añadir nuevos componentes y circuitos a los experimentos del laboratorio. En estos ficheros, denominados ficheros .max, únicamente se deben indicar los nodos en los cuales se encuentran conectados los componentes y los instrumentos del circuito bajo prueba.

Una vez que el *Circuit Solver* o instructor virtual comprueba que el circuito se puede construir y medir en base a la información enviada desde el cliente, se envía la descripción del circuito al Servidor de Equipos y Experimentos.

La solución hardware utilizada para el control de los experimentos está basada en el uso de una matriz de conmutación comercial y el desarrollo un hardware muy sencillo cuya única labor es ampliar la capacidad de la

matriz y disponer así de más componentes en el laboratorio. Al igual que el resto de instrumentos de medida del laboratorio (osciloscopio, generador de funciones, multímetro y fuente de alimentación), la matriz de conmutación también está controlada por un conjunto de algoritmos ejecutados en el Servidor de Equipos y Experimentos, tratándose así como un instrumento más.

Esta es una de las principales diferencias respecto a la solución de control adoptada en el proyecto VISIR, donde al utilizar una matriz de conmutación propia se necesita tanto de algoritmos de control en el servidor como en la matriz, en donde se dispone de una serie de microcontroladores encargados de accionar los relés que conectarán unos componentes con otros.

Como se puede observar en la solución de la Figura 3.5, la matriz de conmutación (*switching matrix* en inglés, por ello aparece referenciada en esta figura como “swtch”) se controla como un instrumento más mediante su algoritmo y driver IVI correspondiente. La salida de este algoritmo es el conjunto de comandos encargados de abrir o cerrar los relés correspondientes en la matriz de conmutación en base a la descripción del circuito obtenida a la salida del *Circuit Solver*. De este modo se crean las conexiones entre los componentes que forman el circuito y los instrumentos necesarios para su test y medida. Durante este proceso, no se hace referencia al tipo de circuito que se desea testear, ya sea predefinido o no, ni a su nivel de dificultad, entendida como número de nodos, mallas y componentes del circuito.

Esto es así gracias a que el diseño del conjunto de algoritmos denominados *Circuit Builder* es independiente de la naturaleza y complejidad del circuito bajo prueba. En el caso de tratarse de un circuito predefinido, éste es considerado como un componente más, con un número determinado de terminales y que será conectado donde corresponda.

Respecto a la complejidad del circuito, ésta únicamente está limitada por el número de nodos físicos disponibles en la matriz (actualmente 8) y el número de componentes reales que pueda albergar la matriz.

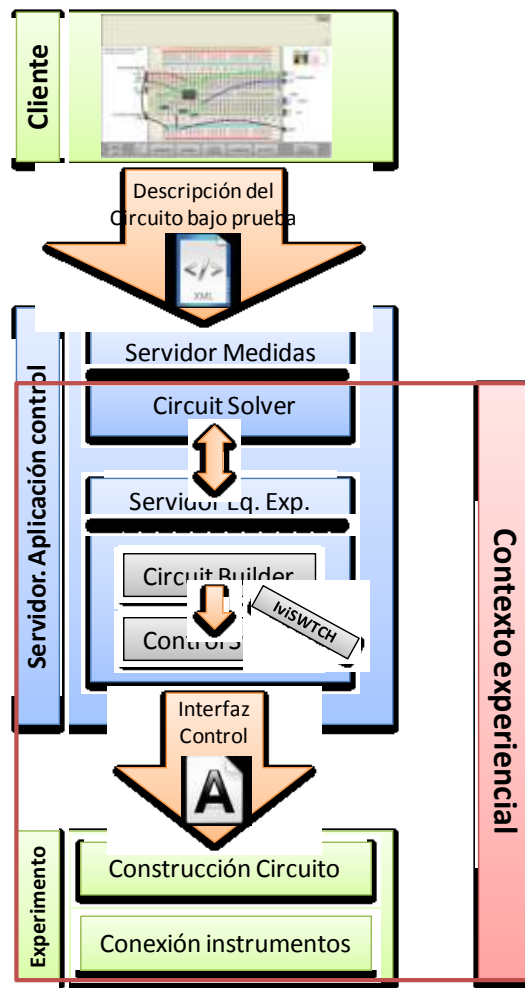


Figura 3.6: Flujo de información para el control software de los experimentos

De manera gráfica, la Figura 3.6 representa el flujo de información desde que el circuito es diseñado por el alumno en el cliente web, hasta que se realiza su construcción física y el conexionado de los instrumentos al circuito en el experimento. Todos los procesos que intervienen, exceptuando la información generada en el cliente, están contemplados en el alcance de este trabajo, por lo que han sido diseñados expresamente para el método de control propuesto. En la sección 3.3.1 se analizan en más detalle aspectos técnicos tanto del *Circuit Solver* como del *Circuit Builder*, conjunto de algoritmos principales utilizados para el control de los experimentos (circuito bajo prueba e instrumentos) del laboratorio.

Problema	Solución
P4. Facilidad para añadir nuevos componentes y circuitos a los experimentos	Se ha desarrollado un algoritmo denominado <i>Circuit Solver</i> en base al hardware de control de los experimentos, en el que únicamente hay que indicar los nodos en los cuales se encuentran conectados los componentes e instrumentos del circuito.
P5. Independencia del tipo de circuito, predefinido o no, y dificultad en función de número de nodos y componentes	Se ha diseñado un conjunto de algoritmos denominados <i>Circuit Builder</i> , que controlan la matriz independientemente del tipo de circuito y su complejidad
P6. Mantener los requisitos respecto a instrumentos de la problemática anterior (escalabilidad, independencia respecto a instrumentos y fabricante)	El control del experimento se realiza en base a una matriz de conmutación comercial, cuyo algoritmo de control está basado en drivers IVI al igual que el resto de instrumentos del laboratorio. La matriz se considera como un instrumento más.

Tabla 3.3: Soluciones para independizar el software de control de los experimentos

Así pues, combinando por una parte el algoritmo diseñado y desarrollado para el control de la matriz de conmutación comercial, y por otra los algoritmos *Circuit Solver* y *Circuit Builder* desplegados en el Servidor de Medidas y en el Servidor de Equipos y Experimentos respectivamente, se consigue dar solución a los problemas relacionados con el control software de los experimentos del laboratorio y que aparecen recogidos en la Tabla 3.3. Obviamente, estas soluciones están directamente relacionadas con las adoptadas desde el punto de vista hardware y que describiremos en la sección 3.1.1.3.

3.1.1.2 Soluciones empleadas en el interfaz de control

En el caso del interfaz de control, su función principal es comunicar la aplicación de control con los experimentos del laboratorio con el objetivo de permitir el intercambio de información, tanto en forma de comandos como de las respuestas generadas en los experimentos y recogidas por los instrumentos. Por consiguiente, el problema principal a resolver es:

- a) Disponer de un interfaz de control y comunicaciones que sea fácilmente desplegable, extensible y que proporcione una calidad de servicio alta en lo referente a la velocidad de transmisión de información en ambos sentidos de la comunicación, es decir desde la aplicación de control hacia los instrumentos y experimentos, y viceversa.

- b) Disponer de un interfaz que permita una rápida configuración de los instrumentos conectados a la red de instrumentación, facilitando su gestión y control.

En el capítulo anterior estudiábamos las diferentes tecnologías utilizadas como interfaz de control para instrumentación de laboratorio: hablábamos de sistemas integrados como PXI y VXI o bien de tecnologías como GPIB y LXI. Una vez analizadas cada una de ellas, razonábamos por qué a nuestro entender y anticipándonos a los problemas que queremos resolver y a los objetivos a satisfacer mediante el método en el cual vamos a integrar el interfaz de control seleccionado, LXI era la tecnología más óptima de las analizadas. De manera resumida y tomando como referencia el estudio anteriormente realizado, hemos llevado a cabo esta selección debido a:

- Coste: LXI es una tecnología basada en Ethernet, estándar ampliamente extendido y que puede ser encontrada prácticamente en cualquier instrumento de laboratorio. Además, dentro de las clases de instrumentos LXI, la más sencilla es la clase C, suficiente para satisfacer los objetivos del método de control propuesto, por lo que incorporar al laboratorio equipos con este interfaz de comunicaciones no supone un precio elevado. A modo de ejemplo, el generador de funciones de Agilent Technologies 33220A, con tecnología LXI y límite de funcionamiento 20MHz, presenta un coste de 1.560,00€ mientras que un equipo similar (Agilent 33521A, 30MHz) pero con tecnología GPIB tiene el mismo coste¹⁰.
- Velocidad de transmisión: como veíamos en el estudio previo, los equipos dotados del interfaz LXI pueden proporcionar una velocidad de transmisión de hasta 10Gb/s en sus clases más altas. Aún así, en la clase C, la velocidad de transmisión puede alcanzar los 100Mb/s superando con creces el resto de interfaces, aunque sí debemos tener en cuenta, que al tratarse de una tecnología basada en Ethernet y que utiliza el protocolo TCP/IP para la comunicación, pueden existir problemas con la latencia en la comunicación.
- Compatibilidad: por definición del estándar todos los equipos LXI deben poder ser controlados mediante drivers IVI, por lo que encaja perfectamente con los requisitos y objetivos que estamos imponiendo

¹⁰ Información obtenida el día 6/09/2011 del distribuidor. Más información en <http://es.rs-online.com>

al método de control. De este modo los equipos LXI utilizados pueden ser controlados mediante una aplicación de control reutilizable en otros equipos.

- **Flexibilidad:** entendida como la capacidad para ampliar la red y aumentar el número de instrumentos a ella conectada. Al estar basada en Ethernet, una red LXI permite conectar un gran número de instrumentos sin tener que rehacer ninguna configuración de la misma. Además, la configuración de la dirección mediante la cual nos comunicaremos con los equipos desde la aplicación de control se puede realizar mediante el servidor web que todos los equipos LXI integran, facilitando esta tarea. Por otro lado, al disponer de protocolo DHCP, la asignación de dirección IP se puede hacer de manera automática en una red local, sin necesidad de involucrar a los servicios técnicos de la universidad.
- **Longevidad:** al tratarse de un estándar basado en Ethernet entendemos que su probabilidad de mantenerse como interfaz de control en equipos de laboratorio es elevada y duradera. Esto nos permitirá asegurar la validez del método de control si utilizamos LXI como interfaz de control.
- **Facilidad de uso:** su facilidad de conexión a la red sin requisitos de hardware específico, así como su rápida configuración mediante DHCP y el servidor web que todos los equipos LXI incorporan, hacen posible que estos instrumentos puedan entrar en funcionamiento rápidamente. Esta característica, combinada con las relativas al uso de drivers IVI para su gestión, hace que la sustitución de un instrumento por otro sea aún más rápida y sencilla, sin la necesidad de complicadas reconfiguraciones o instalaciones. Debido también a su independencia respecto de la aplicación de gestión software, podríamos sustituir este interfaz de comunicaciones por otro, en función de las necesidades sin alterar el funcionamiento del resto del sistema de control.
- **Modularidad:** si bien los equipos LXI no son tan compactos como los equipos que incorporen tecnologías VXI o PXI, esta es una característica importante para el despliegue del método de control propuesto, primando más el resto de características.

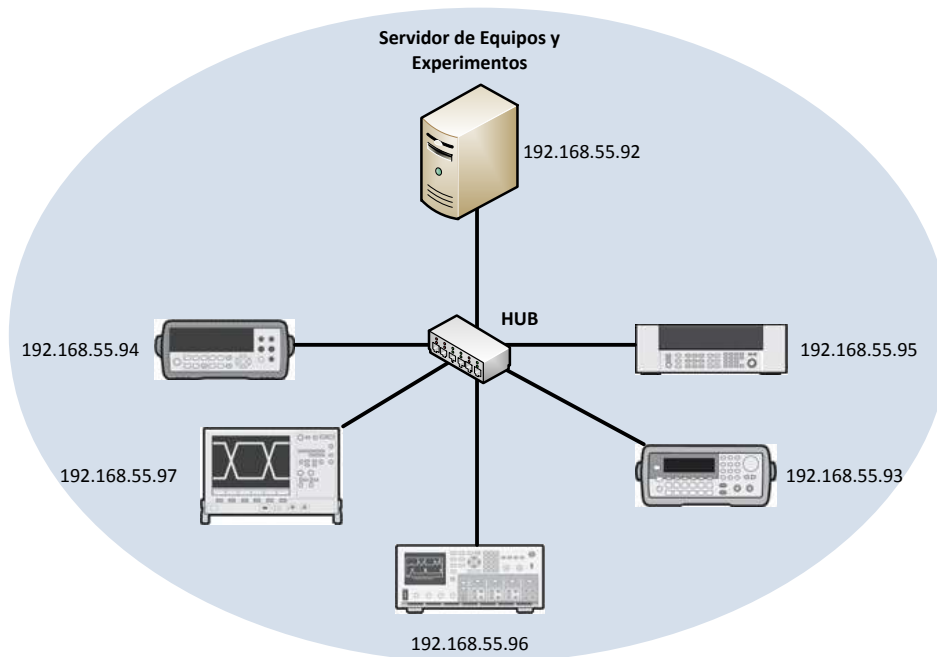


Figura 3.7: Configuración de la red de instrumentación mediante LXI

Como se puede advertir en el repaso a las principales características por las cuales se ha seleccionado LXI como interfaz de control, prácticamente la propia naturaleza del estándar permite dar solución a los problemas planteados a la hora de elegir entre las diferentes tecnologías existentes.

Aún así, se ha desarrollado un algoritmo denominado *LXICom* en el Servidor de Medidas que se encarga principalmente de establecer y gestionar la comunicación entre este servidor y el Servidor de Equipos y Experimentos. Este módulo tiene también como función adecuar la respuesta proporcionada por los instrumentos al protocolo establecido para la comunicación con el cliente. Esta respuesta es recibida por el Servidor de Equipos y Experimentos a través del interfaz LXI ya que en la solución que planteamos y que podemos observar en la Figura 3.7, el Servidor de Equipos y Experimentos forma parte de la red de instrumentación creada mediante el interfaz de LXI.

En la Figura 3.7 se observa como todos los instrumentos disponen de una dirección IP que es utilizada por el Servidor de Equipos y Experimentos para direccionar la información dirigida hacia y desde los equipos de la red del laboratorio. Los equipos LXI incorporan un servidor web por defecto en su clase más sencilla, el cual permite configurar fácilmente esta dirección.

Utilizando este servidor se asigna una IP de manera dinámica a los nodos gracias al protocolo DHCP, por lo que únicamente requeriremos de los Servicios Informáticos de la universidad que asignen siempre la misma IP al mismo equipo a partir de su dirección MAC, operación sencilla y que no requiere tareas complejas. Estamos creando así una red local de instrumentos que únicamente pueden ser accedidos a través del Servidor de Equipos y Experimentos, aislándolos así del exterior y de posibles ataques.

Este servidor web ofrecido por los equipos LXI puede ser utilizado de muy distintas maneras. Por ejemplo, analizábamos en el capítulo anterior cómo el laboratorio remoto eVALIDATE también hacía uso del estándar LXI para comunicar y controlar los instrumentos del laboratorio. En este laboratorio utilizan como paneles frontales virtuales de los instrumentos los servidores web proporcionados por los propios instrumentos LXI. Esta solución presenta la ventaja de la rapidez con la cual se puede desplegar el laboratorio y acceder al control de los instrumentos, pero presenta dos grandes inconvenientes respecto a su empleo en un laboratorio remoto:

1. Los interfaces que proporcionan los servidores web de los instrumentos no se auto-refrescan automáticamente por lo que no siempre muestran al alumno el estado real del instrumento. Además, en muchos instrumentos, como es el caso de los osciloscopios¹¹, el servidor web no proporciona un panel frontal de control si no que el acceso al control del instrumento es a través de una aplicación Virtual Network Computing, la cual no se asemeja al panel frontal del equipo lo que implica que el alumno tenga que aprender a utilizar este nuevo panel virtual.
2. El laboratorio remoto así desplegado es únicamente válido para los instrumentos que se han contemplado, ya que cada fabricante puede proporcionar un tipo de panel frontal virtual basado en diferentes tecnologías (HTML, Java, VNC, etc.), y por otro lado los comandos de control enviados desde el panel remoto virtual al propio equipo están

¹¹ Hemos comprobado en el laboratorio que al menos en los osciloscopios de Agilent DSO5012 y MSO7104, el servidor web no proporciona el acceso a un panel remoto virtual igual que el del equipo, si no que el control se realiza a través de una sesión VNC. En el caso del generador de funciones, hemos comprobado en el equipo Agilent 33220A como el panel virtual ofrecido por el servidor web, dispone de un botón "update display" para refrescar el estado del instrumento

basados en el estándar SCPI, lo que implica que estos comandos sean propios de cada instrumento.

Por otro lado, tal y como venimos indicando a lo largo de este trabajo, el método de control presentado está desarrollado para ser utilizado en un laboratorio remoto de electrónica en donde los experimentos, aún pudiendo ser complejos, no requieren de instrumentos que implementen las funcionalidades más avanzadas proporcionadas por las clases B y A del estándar LXI, aunque gracias a la modularidad de los algoritmos de control creados y al uso de drivers reutilizables es sencillo añadir estas funciones al método de control.

En definitiva, el simple empleo de las características básicas de equipos que implementen el estándar LXI-Clase C como interfaz de control ha permitido dar solución a los problemas planteados en el interfaz de comunicaciones del método propuesto. Únicamente hemos tenido que desarrollar un algoritmo específico encargado de formatear la información enviada desde los instrumentos a través del interfaz LXI y que será enviada de vuelta al cliente, y realizar la configuración de los nodos de la red de instrumentación tal y como mostrábamos en la figura anterior. A modo de resumen, la Tabla 3.4 muestra los problemas y soluciones adoptadas en el ámbito del interfaz de control.

3.1.1.3 Soluciones para el control de los experimentos

Si bien ya se ha descrito en la sección 3.1.1.1 cómo se ha adaptado el software de control para que éste sea independiente de los diferentes circuitos bajo prueba que se pueden encontrar en los experimentos del laboratorio, en esta sección nos centraremos en describir la solución hardware propuesta para poder llevar a cabo el control de los experimentos, es decir, de los circuitos bajo prueba y los instrumentos utilizados para su monitorización.

Mediante el control de experimentos nos estaremos refiriendo a los métodos y técnicas empleados para en un primer momento hacer posible que el usuario construya circuitos bajo prueba y posteriormente realice medidas sobre ellos gracias a los instrumentos de los que dispone el laboratorio y que serán conectados en aquellos puntos del circuito que el usuario considere oportuno.

Problema	Solución
P7. Interfaz de control fácilmente desplegable, extensible y alta velocidad de transmisión	Características propias del interfaz LXI, por lo que su propio uso proporciona la solución a estos problemas
P8. Interfaz de control fácilmente configurable y robusto	Mediante el servidor web y el protocolo DHCP del propio los propios equipos LXI, se accede a la configuración de los equipos de manera sencilla. Creamos de esta forma una red de área local de instrumentos solo accesibles desde el laboratorio y a través del Servidor de Equipos y Experimentos.

Tabla 3.4: Soluciones adoptadas en el interfaz de control

A este respecto, los principales problemas a solucionar mediante el sistema del método de control propuesto son:

- a) Obtener una solución de control que esté basada íntegramente en soluciones comerciales, reduciendo al máximo el uso de soluciones propietarias.
- b) Proporcionar un método que haga posible la construcción de circuitos basados en componentes discretos o circuitos preconstruidos en los experimentos del laboratorio.
- c) Obtener una solución que no introduzca errores significativos en las medidas realizadas sobre los circuitos bajo prueba, así como repetitividad en las medidas obtenidas.
- d) Aumentar el rendimiento del sistema respecto al número de componentes, así como reducir la complejidad existente a la hora de añadir nuevos componentes a la matriz de conmutación empleada.

A continuación describiremos cuáles son las soluciones planteadas a cada uno de estos problemas, de manera que la combinación de todas ellas forma el sistema diseñado para el control y configuración de los experimentos incluidos en un laboratorio remoto de electrónica. En la sección 3.3 se analiza con mayor profundidad los detalles técnicos de esta solución.

A) *Solución basada íntegramente en soluciones comerciales*

En el capítulo anterior cotejábamos las diferentes alternativas existentes para controlar los experimentos del laboratorio: utilizar soluciones comerciales o bien diseñar y desarrollar una solución propia que satisfaga las exigencias de funcionamiento. En este caso se ha optado por una solución hardware cuyo elemento principal es una matriz de conmutación comercial. Para la fase de

pruebas detallada en el capítulo cuarto, esta matriz está formada por el mainframe 34980A y dos módulos de conmutación 34932A, cada uno con 2 matrices de conmutación 4x16, que se encuentran conectados y controlados por el mainframe.

Uno de los objetivos de esta solución es hacer posible que el alumno pueda interconectar remotamente componentes electrónicos reales para poder construir circuitos de manera remota y testarlos mediante los instrumentos del laboratorio. Por lo tanto necesitamos un elemento que nos permita realizar tres acciones muy importantes para el correcto funcionamiento del sistema:

- a) Interconectar las matrices de conmutación disponibles en el mainframe. En la solución adoptada, partiendo de dos matrices de 4x16, se obtiene una matriz de conmutación de 8x32. Si dispusiéramos de otra matriz obtendríamos una matriz de 8x48 y así sucesivamente.
- b) Proporcionar componentes a la matriz de conmutación, la cual se encargará de interconectarlos mediante la activación de los relés correspondientes.
- c) Hacer posible que todos los componentes se pueden interconectar entre sí y poder conectar también los instrumentos al circuito así construido.

Para realizar estas tareas, se han diseñado unos módulos de expansión sin ninguna lógica integrada, cuyas sus únicas funciones son implementar las acciones enumeradas anteriormente. La Figura 3.8 muestra la matriz de conmutación resultante, donde se ve claramente dónde se sitúan los instrumentos y dónde los componentes. En esta configuración, si bien las puntas de medida de los instrumentos siempre están conectadas en los mismos puntos, los componentes pueden ser situados en diferentes posiciones de la matriz.

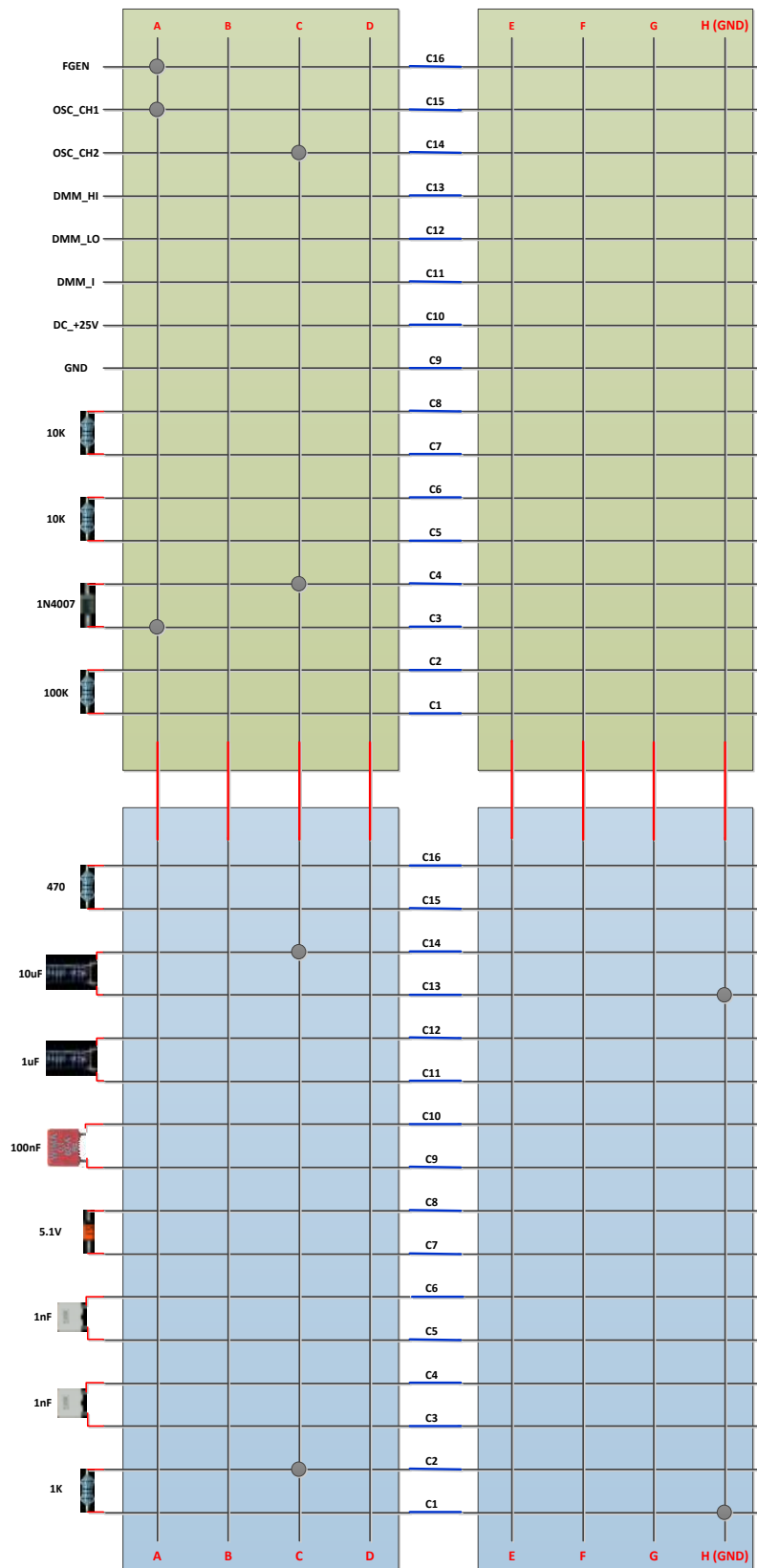


Figura 3.8: Configuración de la matriz 8x32 resultante

Esta interconexión de las matrices es una de las posibles soluciones. Otra posible alternativa es, por ejemplo, optar por una configuración que permita obtener una matriz de 16x16. En el escenario de pruebas planteado únicamente se ha dispuesto de 2 módulos en el mainframe, por lo que se ha optado por esta configuración de 8x32 para tener así más componentes disponibles en la matriz ya que también se deben conectar los instrumentos en la matriz.

Aún así, el sistema de control software diseñado es independiente de la configuración de matriz empleada porque únicamente manteniendo la configuración del Servidor de Equipos y Experimentos, el sistema de control funcionará de igual modo con cualquier configuración de matriz de conmutación empleada.

B) Hacer posible la experimentación con circuitos predefinidos o basados en componentes discretos

Tal y como se ha citado anteriormente, no se ha realizado ninguna distinción entre circuitos predefinidos o circuitos basados en componentes discretos. En el caso del diseño hardware sucede exactamente lo mismo. Un circuito predefinido será considerado como un componente más, el cual dispondrá de 'n' conectores o terminales, que serán conectados a la matriz de conmutación de la Figura 3.8 como un componente más.

Para poder crear circuitos de manera remota, la matriz de conmutación resultante está controlada mediante la aplicación *Circuit Builder* que se ejecuta en el Servidor de Equipos y Experimentos. Estos algoritmos, mediante la lectura de un fichero de configuración específico, conocen donde se encuentran conectados cada uno de los componentes de la matriz de conmutación y los instrumentos del laboratorio, de manera que partiendo de esta información y de la descripción de circuito enviada por el algoritmo *Circuit Solver* del Servidor de Medidas y utilizando las funciones proporcionadas por el driver IVI de la matriz de conmutación, se abren o cierran los relés adecuados para construir así el circuito bajo prueba diseñado por el usuario en el cliente web.

Es decir, gracias a los módulos de expansión que interconectan las matrices y que conectan a la matriz así obtenida los componentes y los instrumentos del laboratorio, junto con el software de control *Circuit Builder*

y *Circuit Solver* de la aplicación de control, es posible trasladar el circuito virtual del alumno a un circuito real que puede ser medido y modificado de manera remota.

En la sección 3.3.1.3 se explica con un ejemplo muy sencillo cómo se realiza esta traducción así como las acciones que llevan a cabo cada uno de los elementos que intervienen en el proceso de construcción de un circuito físico de manera remota.

C) *Obtener una solución que no introduzca errores significativos en las medidas*

Para poder construir físicamente un circuito a partir de los componentes dispuestos en la matriz de conmutación 8x32 obtenida, es necesario abrir y cerrar unos determinados relés en función de la descripción recibida en el Servidor de Equipos y Experimentos del circuito deseado. En estas matrices de conmutación disponemos de relés electromecánicos OMRON EN-G6S-2. Según el fabricante, estos relés presentan una resistencia en cada uno sus contactos de $75\text{m}\Omega$ medidos en un ambiente controlado, fuera de la matriz de conmutación y con una corriente de 10mA y una tensión de 1VDC . Es decir, este valor no tiene en cuenta que el relé se encuentra soldado en una placa electrónica así como la circuitería adicional de la matriz en la que se encuentra (pistas y conectores).

Por esta razón, para caracterizar la resistencia de estos relés hemos empleado el método de 4 hilos para la medida de resistencias de muy pequeño valor. Como adelanto a los resultados de estas medidas, las cuales aparecen descritas en profundidad en el siguiente capítulo, podemos decir que tomando medidas en tres relés diferentes de la matriz de conmutación, los valores obtenidos están en torno a un valor medio de $0,310\Omega$.

Además, teniendo en cuenta que para poder conectar los componentes e instrumentos a la matriz debemos utilizar los módulos de expansión a los que nos referíamos anteriormente, hemos repetido estas medidas en los mismos puntos y con el mismo método. En este caso, el valor medio de la resistencia medida es de $0,320\Omega$.

Por otro lado, hemos realizado medidas sobre circuitos de diferente complejidad en tres escenarios diferentes: el circuito montado manualmente sobre una placa de prototipos, el mismo circuito montado remotamente en el laboratorio VISIR, y el mismo circuito montado mediante el método de

control que aquí describimos. Como resultado de estas medidas, cuyos resultados mostramos en el capítulo, podemos decir que el sistema de control para el ensamblado y medida de circuitos de manera remota no introduce más errores significativos en las medidas que los que se obtienen en una medida sobre el mismo circuito en un laboratorio manual.

Junto con las consideraciones anteriores, mediante los algoritmos de medida diseñados en este método de control hemos solucionado un problema o más bien error que aparecía en la plataforma VISIR a la hora de realizar medidas de tensión de señales periódicas utilizando el multímetro. En estos casos, la medida que debe ofrecer el multímetro es el valor medio de la tensión en un periodo de la señal.

En cambio, si realizamos esta medida mediante la plataforma VISIR, los resultados obtenidos difieren de este valor, ya que al ejecutarse los algoritmos de medida durante un corto instante de tiempo, el valor muestreado de la señal a medir no siempre abarca un periodo, por lo que el algoritmo de medida del multímetro devuelve cualquier valor de los que puede tomar la señal.

Gráficamente, el problema aparece representado en la Figura 3.9, en donde se observa cómo dependiendo del instante de tiempo en el que se realice la medida, el valor obtenido puede ser muy distinto, pudiendo ser V_2 si la medida se ejecuta en t_1 , 0 Voltios si se ejecuta en t_2 o V_1 si la medida se realiza en el instante de tiempo t_3 . En este ejemplo, como la señal es periódica y no tiene ningún offset de tensión, el valor medio medido debería ser de 0 Voltios.

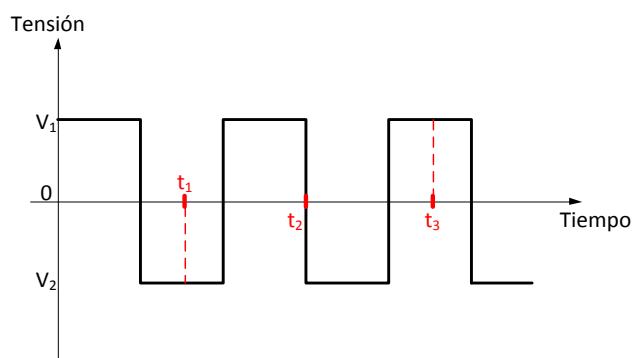


Figura 3.9: Ejemplo de toma de medidas de tensión sobre señal periódica

Mediante el método de control y medida implementado, se consigue solucionar este error modificando el algoritmo de medida del multímetro, de tal forma que ahora es capaz de ofrecer resultados válidos en el caso de querer obtener el valor de tensión media de una señal periódica. Para ello, hemos aumentado el número de muestras tomadas de la señal periódica y posteriormente hemos realizado un proceso estadístico que permite calcular el valor medio de dicha señal. De esta forma, conseguimos solucionar un error propio del sistema de medida VISIR. En el capítulo 4 incluimos una serie de pruebas que nos ha permitido corroborar esta solución.

D) Aumento de la capacidad del sistema y de la facilidad para añadir nuevos componentes

Este problema está directamente relacionado con la experiencia adquirida durante el desarrollo y despliegue del laboratorio VISIR en la Universidad de Deusto, de modo que su solución se presenta también como una mejora al sistema empleado en dicho laboratorio.

Basándonos en la matriz mostrada en la Figura 3.8 en la que se consideran las filas R1 a R8 como nodos del circuito y las columnas como los lugares en los que se conectan los componentes, disponemos actualmente de espacio para albergar 12 componentes de dos terminales cada uno de ellos. Obviamente si utilizamos componentes de más terminales la capacidad disminuye, pero gracias a la simplicidad los módulos de expansión, añadir nuevos módulos es fácil y dispondríamos de al menos 16 columnas más para realizar conexiones. Añadir nuevos módulos de conmutación tampoco implica en absoluto ninguna readaptación del código de control *Circuit Builder* del Servidor de Equipos y Experimentos.

Pero además, este sistema de control basado en matrices comerciales aumenta el rendimiento del sistema respecto al número de componentes necesarios frente al empleado en el laboratorio VISIR. En el sistema de control de experimentos del laboratorio VISIR, un componente se encuentra cableado físicamente en la matriz de conmutación entre dos nodos del circuito de manera permanente. Es decir, si una resistencia de $10K\Omega$ está conectada entre los nodos A y B y necesitamos para un circuito una resistencia del mismo valor entre los nodos C y D, se deberá colocar en la matriz otra resistencia del mismo valor cableada físicamente entre dichos nodos, ocupando por tanto dos relés en la matriz de conmutación. En cambio

en el sistema propuesto, un componente no se encuentra cableado físicamente a ningún nodo, sino que las conexiones se realizan de manera dinámica en función de las necesidades del circuito.

Para ilustrarlo mejor, imaginemos que disponemos de tres resistencias de valores $1k\Omega$, $10k\Omega$ y 470Ω y que se quieren realizar todas las combinaciones posibles y obtener la resistencia equivalente de todas ellas. Los circuitos resultantes serían los mostrados en la Figura 3.10 en donde se indica con una letra cada uno de los nodos del circuito entre los cuales se conectaría cada componente.

La primera diferencia entre el sistema VISIR y la solución propuesta es que para reducir al máximo el número de componentes a utilizar, en el caso del VISIR debemos pensar concienzudamente en qué nodos se va a conectar físicamente cada componente. En cambio, en el sistema planteado este problema no existe, ya que cada componente se conecta en cualquier nodo de la matriz de conmutación en función del diseño realizado en cada momento.

Puede que pensar en una solución óptima para la plataforma VISIR no suponga un gran problema si es que se sabe a priori qué circuitos se quiere permitir construir. En cambio, si se desean añadir circuitos después de un tiempo, se debe partir de las conexiones ya realizadas o bien reiniciar todo el planteamiento. Es decir, si en futuro circuito se va a necesitar una resistencia de $1K\Omega$, quizás no se pueda conectar entre los nodos A y B y se tenga que añadir una resistencia del mismo valor pero conectada entre otros dos nodos.

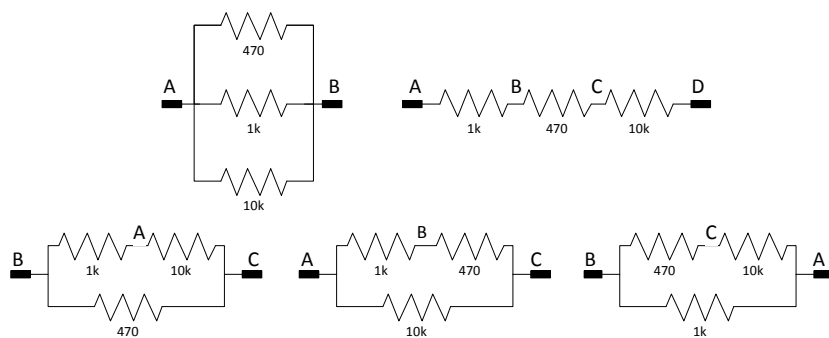


Figura 3.10: Ejemplo de circuitos formados con tres resistencias

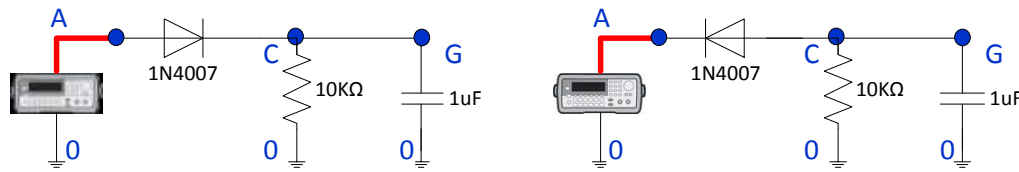


Figura 3.11: Ejemplo de configuración de circuitos con diodos

Por otro lado tal y como se extrae de la Figura 3.11, para construir estos circuitos en la plataforma VISIR necesitamos una resistencia de $1K\Omega$ entre los nodos A y B, 2 resistencias de 470Ω entre los nodos, A - B, B - C, y 5 resistencias de $10K\Omega$ entre los nodos A- B, C- D y A - C. Es decir, se requieren 6 resistencias para poder realizar todos estos circuitos y medidas mientras que en el sistema propuesto, el número de resistencias se reduce a 3. En el siguiente capítulo, en el cual se describen las pruebas a las que hemos sometido este método de control, utilizaremos otro escenario más complejo en el cual también cotejamos ambas alternativas en base a su rendimiento relativo al número de componentes necesarios para realizar ciertos circuitos y medidas.

Además debemos considerar que en este ejemplo las resistencias no tienen polaridad, y estar conectado entre B y C tiene el mismo efecto en la medida que estar entre C y B. En cambio, si tuviéramos otro tipo de componentes tendríamos que hacer esta consideración a la hora de realizar las conexiones físicas en la matriz del laboratorio VISIR. Por ejemplo, en el caso de un diodo, si queremos utilizarlo en la plataforma VISIR polarizado en directa y en inversa, necesitaremos disponer obligatoriamente 2 diodos en la matriz pues cada uno estará conectado a una pareja de nodos diferente. En cambio en el sistema planteado es necesario únicamente un diodo, y su conexión a una pareja de nodos u otra vendrá determinada por la descripción del circuito. A modo de explicación, emplearemos el ejemplo de la Figura 3.11.

Para poder crear estos dos circuitos mediante la matriz del laboratorio VISIR, necesitamos tener un diodo cuyo ánodo esté cableado al nodo A y su cátodo al nodo C y otro diodo que tenga el ánodo en el nodo C y el cátodo en A. En cambio, para realizar los mismos circuitos en nuestra solución, únicamente necesitamos un diodo que se conectará dinámicamente a los nodos necesarios en función de la descripción del circuito. Para ello, el

algoritmo *Circuit Builder* conoce mediante un fichero de configuración dónde se encuentra cada componente y su polaridad.

Por otro lado, y también como elemento diferenciador respecto a la matriz de conmutación del laboratorio VISIR, para realizar medidas de intensidad en circuitos construidos sobre ella, es necesario incluir resistencias de 0Ω entre los dos nodos de la rama en la cual queremos realizar la medida como si se tratara de un componente más, consumiendo así el espacio de un componente en la matriz.

En cambio mediante la configuración del sistema de control de experimentos propuesta, en el fichero utilizado por el *Circuit Solver* y que contiene la descripción de los circuitos posibles, únicamente debemos describir la rama del circuito sobre la que queremos medir la intensidad (por ejemplo, entre los nodos B y C de la Figura 3.12) sin tener que disponer de una resistencia de 0Ω entre ambos terminales, ya que este cortocircuito puede ser implementado sobre la matriz, cerrando los relés apropiados, de lo cual se encarga el algoritmo *Circuit Builder* del Servidor de Equipos y Experimentos.

En resumen, la configuración de las conexiones realizadas entre las matrices de conmutación contenidas en los módulos del mainframe, y que dan como resultado una matriz de conmutación de 8×32 , permite reducir el número de componentes a disponer en la matriz, principalmente debido a que un componente puede estar conectado en cualquier nodo de un circuito independientemente de su polaridad, ya que solo los componentes se conectan a un nodo u otro dinámicamente en función de la descripción del circuito enviado por el usuario, previamente chequeado por el *Circuit Solver* del Servidor de Medidas.

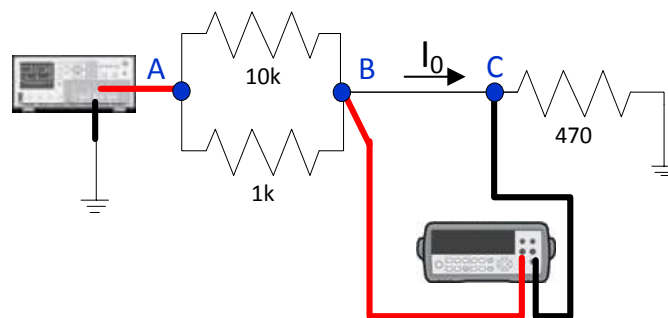


Figura 3.12: Circuito ejemplo para la medida de intensidades

Mediante el sistema de control aquí diseñado no se tiene que realizar el cableado físico de los componentes a unos nodos específicos de la matriz de conmutación, por lo que no es necesario conocer en profundidad la estructura física de la matriz para añadir nuevos componentes, simplificando la tarea de agregar nuevos componentes a la matriz.

En el sistema planteado, una vez que el componente se conecta en los módulos de expansión, únicamente es necesario actualizar el fichero de configuración utilizado por el Servidor de Equipos y Experimentos para saber la posición que ocupan todos los componentes y determinar así qué relés se deben cerrar en función de la descripción del circuito generada por el *Circuit Solver* del Servidor de Medidas. A modo de resumen la Tabla 3.5 muestra cada uno de los problemas planteados a la hora diseñar y desarrollar el sistema de control de los experimentos del laboratorio, así como las soluciones adoptadas para cada uno de ellos.

Problema	Solución
P9. Utilizar únicamente instrumentos comerciales	Utilizamos una matriz de conmutación 8x32 obtenida mediante la interconexión de las matrices de conmutación 34932A contenidas en el mainframe 34980A
P10. Hacer posible la creación de circuitos de diferente complejidad, tanto predefinidos como basados en componentes discretos	Mediante el uso de unos módulos de expansión básicos conectamos los componentes e instrumentos a la matriz 8x32 obtenida previamente. Mediante la aplicación software de control <i>Circuit Builder</i> se activan los relés determinados y se obtiene el circuito bajo prueba mediante la interconexión de componentes e instrumentos
P11. Obtener una solución que no introduzca errores significativos en las medidas obtenidas en el circuito bajo prueba	Tanto las matrices utilizadas como los módulos de expansión diseñados, introducen errores mínimos en las medidas. Apenas hay diferencia entre las medidas obtenidas mediante este sistema, el sistema VISIR y medidas realizadas sobre el circuito montado en una placa de prototipos. Se ha solucionado el problema de medida de la tensión media de una señal periódica mediante DMM.
P12. Aumentar la capacidad de la matriz resultante en comparación con la utilizada en el laboratorio VISIR	Los componentes no están conectados físicamente a ninguno de los nodos de la matriz, sino que esta conexión se hace dinámicamente en función de la descripción del circuito.
P13. Hacer posible que añadir nuevos componentes a la matriz sea una tarea más simple y no requiera conocer de manera técnica el funcionamiento de la matriz	Los módulos de expansión hacen posible que añadir nuevos componentes sea fácil y barato ya que no es necesario el cableado físico de los componentes a nodos específicos

Tabla 3.5: Soluciones adoptadas en el sistema de control de experimentos

3.1.2 Resumen de soluciones y aportaciones del método de control propuesto

En las secciones anteriores se describen las soluciones a los problemas de cada uno de los sistemas contemplados en el alcance de este trabajo (Figura 1.6) y que forman el método de control de un laboratorio remoto: aplicación de control, interfaz de comunicaciones y sistema de control de los experimentos. Para llegar a estas soluciones, se han realizado una serie de desarrollos tanto en la aplicación de control como en el sistema de control de los experimentos. En cambio, en el caso del interfaz de comunicaciones nos hemos beneficiado de las características propias de la tecnología seleccionada para solucionar los problemas presentados.

En el desarrollo de la aplicación de control se ha adoptado la misma arquitectura de servidores utilizada en el proyecto VISIR, ya que permite separar perfectamente las funciones del servidor del laboratorio remoto en dos entidades, Servidor de Medidas y Servidor de Equipos y Experimentos. Esta división ha permitido centrarnos exclusivamente en el desarrollo de este último ya que es en este servidor en cual se ejecutan todas las funciones de control de los circuitos e instrumentos que intervienen en los experimentos del laboratorio. Aún así hemos tenido que realizar ciertas modificaciones en el Servidor de Medidas para adaptar su lógica a las especificaciones del método de control propuesto.

De manera resumida, los desarrollos realizados en la aplicación de control y que han permitido alcanzar las soluciones a los problemas planteados han sido:

- Actualizar funciones del Servidor de Medidas para que aun manteniendo intacta la información de configuración y control enviada desde el cliente web, ésta pudiera ser utilizada por el Servidor de Equipos y Experimentos con cualquier tipo de instrumento independientemente de su fabricante.
- Se han diseñado y desarrollado completamente todos los algoritmos de control de instrumentos que se ejecutan en el Servidor de Equipos y Experimentos. Para el acceso a las funciones de control proporcionadas por el firmware de cada uno de los equipos del laboratorio hemos utilizado drivers IVI, cuyas prestaciones y

características han permitido independizar todas las aplicaciones de control del tipo de instrumentos y circuitos bajo prueba del laboratorio. Gracias a ello, la actualización o reemplazo de un instrumento por otro, no implica ninguna actualización en su algoritmo de gestión, ya que incluso es independiente del interfaz de control empleado.

- Todos los instrumentos del laboratorio, incluso los encargados del control, configuración y medida del circuito bajo prueba, son equipos comerciales, luego todo su control software sigue las mismas especificaciones que el resto de instrumentos del laboratorio.
- Estas aplicaciones de control han sido diseñadas de manera modular y desarrolladas mediante LabVIEW, de manera que añadir nuevos instrumentos al laboratorio no implica la recodificación de ninguno de los algoritmos existentes.

En lo relativo al interfaz de control, el empleo de LXI como tecnología de comunicación entre la aplicación software de gestión y los instrumentos del laboratorio, nos ha permitido solucionar de manera sencilla y rápida los problemas que encontrábamos en otras plataformas que hacían uso de otras tecnologías como PXI, VXI o GPIB. De este modo, gracias a la elección de esta solución se ha obtenido un medio de control y comunicación que nos ha facilitado:

- Obtener una red de área local en la que se encuentran conectados todos los instrumentos que intervienen en un experimento. Al estar fundamentada en Ethernet, esta red es fácilmente desplegable y ampliable desde el punto de vista de número de equipos bajo control.
- Disponer de una red de instrumentos robusta respecto a tiempos de respuesta de los equipos, gracias a la creación de una red en la cual únicamente se encuentran los instrumentos y la aplicación de control. De esta manera hemos conseguido también que los instrumentos solamente puedan ser controlados a través de la aplicación de control, evitando posibles ataques externos.

Finalmente, hemos desarrollado un sistema para el control de los experimentos basado en una matriz de conmutación comercial controlada de igual modo que el resto de instrumentos del laboratorio. Además, se han

diseñado y desarrollado unos módulos de expansión que permiten ampliar el número de componentes y conexiones de los circuitos bajo prueba mediante la interconexión de las matrices de conmutación contenidas en el instrumento mainframe empleado. Esta configuración del hardware de control de experimentos, junto con el software de control de la matriz resultante 8x32 nos ha permitido:

- Basar el control de los circuitos bajo prueba y la conexión de los instrumentos para las medidas en un equipo totalmente comercial que puede ser controlado mediante drivers IVI al igual que el resto de instrumentos del laboratorio. Esta característica nos permitirá sustituir la matriz de conmutación empleada por otros modelos o configuraciones que mejor se adapten a nuestras necesidades en un momento determinado.
- Hacer posible que el alumno pueda experimentar tanto con circuitos predefinidos como con circuitos construidos totalmente por él, ya que el software y hardware de control es independiente del tipo de experimento.
- Disponer de unos módulos de expansión básicos y sencillos, que sirven para obtener una matriz de conmutación 8x32 y poder conectar en dicha matriz los componentes e instrumentos disponibles en los experimentos.
- Obtener un hardware utilizado por el sistema para interconectar los componentes e instrumentos, que apenas introduce errores en las medidas obtenidas, por lo que hace que el sistema sea seguro y válido para su uso en un laboratorio remoto de electrónica analógica.

Además, como mejoras respecto al sistema de control empleado en el laboratorio VISIR:

- Hemos mejorado el rendimiento de la matriz respecto al número de componentes necesarios para realizar circuitos sobre la misma.
- Eliminamos la necesidad de introducir cortocircuitos como si se trataran de componentes que intervienen en un circuito para hacer posible la medida de intensidades.

- Al no estar los componentes conectados física y permanentemente a unos nodos determinados de la matriz, sino que ésta conexión se realiza de manera dinámica en función de la descripción del circuito, añadir nuevos componentes o circuitos predefinidos es más sencillo y rápido que en el laboratorio VISIR.

Observando los puntos anteriores, los cuales recogen la descripción de las soluciones aportadas en cada uno de los sistemas que forman el método de control, podemos decir que los objetivos planteados al comienzo de este trabajo son alcanzados.

En las siguientes secciones 3.2 y 3.3 procederemos a examinar en profundidad tanto los detalles técnicos como de configuración de cada uno de los sistemas desarrollados.

3.2 Aplicación software de control de la solución propuesta

Tal y como adelantábamos, la aplicación software de control que proponemos está basada en la arquitectura definida en el proyecto VISIR, en la cual las funciones de gestión y validación de datos recaen en el Servidor de Medidas y las funciones de control de los experimentos se ejecutan en el Servidor de Equipos y Experimentos. Volviendo sobre la Figura 3.4 vemos cómo para plantear las soluciones descritas en la sección anterior se ha diseñado y desarrollado completamente el Servidor de Equipos y Experimentos así como los algoritmos del Servidor de Medidas que llevan a cabo las funciones de validación de los datos de configuración de los experimentos.

También hemos rediseñado el protocolo de comunicaciones existente entre ambos servidores, ya que la información intercambiada debe poder ser utilizada para controlar y leer los datos de cualquier instrumento que intervenga en un experimento. En esta sección se analiza en profundidad los desarrollos llevados a cabo en cada uno de estos servidores, así como el protocolo de comunicaciones diseñado y establecido entre ellos.

3.2.1 Servidor de Medidas

El diseño del Servidor de Medidas utilizado en la aplicación de control está basado en el propuesto en el proyecto VISIR, ya que aunque las funciones

correspondientes a la gestión de usuarios, conexiones, perfiles, así como la comunicación con el cliente web no entran dentro del alcance de este trabajo y por ello no han sido modificadas, sí son necesarias para comprobar el posterior funcionamiento del sistema.

Como se ha tomado como punto de partida el diseño implementado en el proyecto VISIR, los cambios efectuados en determinadas funciones, así como los nuevos desarrollos introducidos se han realizado en C++, la misma herramienta utilizada en su diseño original¹².

En la Figura 3.13 se indican los bloques principales del Servidor de Medidas. En color azul se han destacado aquellos módulos lógicos que han sido modificados o rediseñados por completo para adaptar el funcionamiento de este servidor a la aplicación de control del método propuesto.

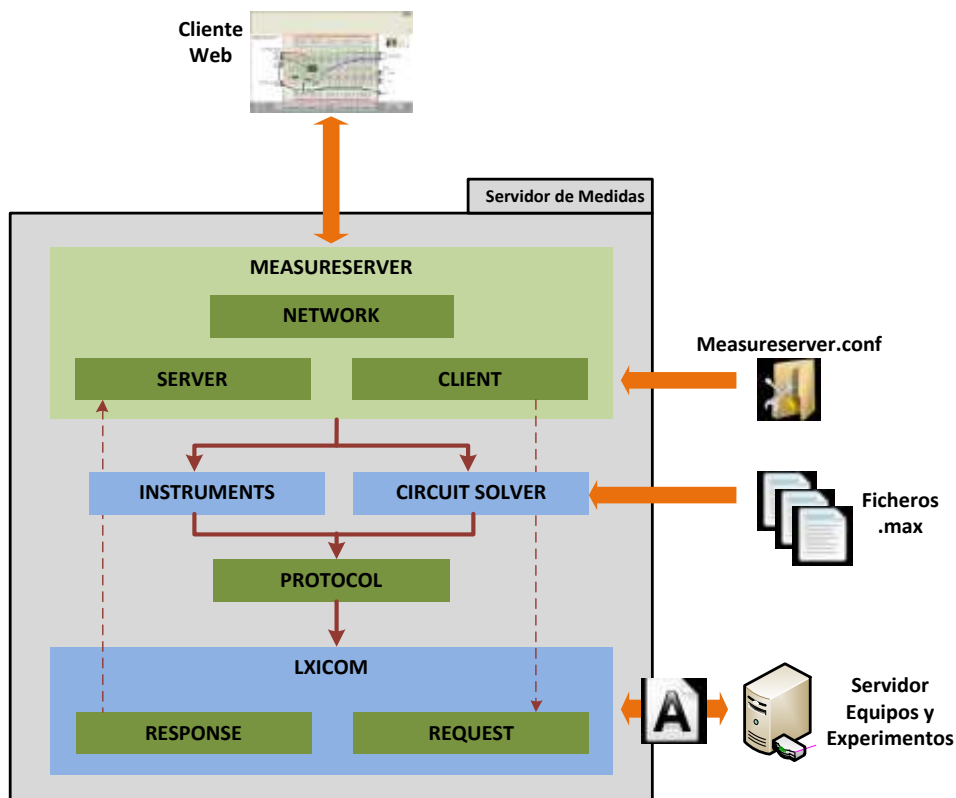


Figura 3.13: Arquitectura lógica del Servidor de Medidas

¹² El código completo del Servidor de Medidas, se encuentra compartido bajo licencia GNU GPL en <http://svn.openlabs.bth.se/trac/measureserver/changeset/485>

A) Funciones no modificadas del Servidor de Medidas

Para la comunicación entre el cliente web y el Servidor de Medidas se ha utilizado el protocolo de comunicaciones basado en XML y definido en el proyecto VISIR. Este protocolo describe tanto las peticiones realizadas por el alumno a través del interfaz web del laboratorio remoto, como las respuestas generadas por el experimento y devueltas al cliente. Esta comunicación se realiza en base a una comunicación LAN gestionada por el algoritmo *Network* ejecutado en el Servidor de Medidas, el cual se encarga de crear y gestionar las comunicaciones tanto ascendentes como descendentes.

En la Figura 3.13 se denomina comunicación descendente a la establecida entre el cliente y los servidores del laboratorio con el objetivo de realizar una petición sobre el laboratorio, mientras que se entiende como comunicación ascendente a la que lleva la información de la respuesta producida por el laboratorio ante la petición anteriormente realizada por el cliente. Por tanto, el algoritmo *Network* es el encargado de establecer estas comunicaciones basadas en un modelo *request-response* entre el cliente web y el laboratorio.

Este algoritmo *Network* se ejecuta en un módulo principal denominado *Measureserver*, el cual se encarga de gestionar todos los procesos que se ejecutan en el servidor, entre ellos el de gestión de la comunicación con el cliente en colaboración con el algoritmo *Network*. Mientras que este proceso se encarga de la gestión de la comunicación LAN entre el servidor y el cliente, el módulo *Measureserver* coordina las peticiones de conexión realizadas por los alumnos que se conectan al laboratorio. El número de alumnos que pueden estar conectados en una sesión del laboratorio y el número máximo de peticiones que se pueden realizar simultáneamente al laboratorio se definen en el archivo de configuración del servidor denominado *measureserver.conf*.

La gestión de las conexiones del laboratorio se realiza manteniendo su número en el valor óptimo para proporcionar a cada uno de los alumnos conectados la respuesta de su propio experimento en el menor tiempo posible y, aún tratándose de experimentos por lotes, se consigue proporcionar al alumno la sensación de estar accediendo al laboratorio en tiempo real.

El *Measureserver* también se encarga de devolver a cada usuario la respuesta generada por el experimento diseñado en el cliente web. Teniendo en cuenta que puede haber varios usuarios a la vez accediendo al laboratorio, debemos disponer de un sistema de identificación para no cometer errores a la hora de enviar las respuestas generadas por el laboratorio. Para ello, cada uno de los paquetes de datos en formato XML intercambiados entre el cliente y el servidor va identificado con una etiqueta denominada *sessionkey*, en la cual se codifica la cookie generada en el cliente y que identifica inequívocamente a cada uno de los usuarios que están accediendo al laboratorio.

Estas funciones tanto de gestión de las comunicaciones como de las conexiones de los usuarios están fuera del alcance de este trabajo, pero son recursos indispensables para el periodo de pruebas que llevaremos a cabo en el capítulo 4.

B) *Funciones diseñadas y desarrolladas en el método de control propuesto*

Las funciones que describiremos a continuación son aquellas que se encargan de la validación de los datos enviados desde el cliente a los experimentos, tanto en lo que respecta a la configuración de los instrumentos como de la construcción remota del circuito bajo prueba mediante el sistema de control de experimentos propuesto.

Estas comprobaciones son realizadas por los algoritmos recogidos en el módulo denominado *Instruments*. Como veíamos en la sección anterior, el control de los instrumentos se realiza gracias a las funciones proporcionadas por los drivers IVI de cada tipo de experimento. Para alimentar estas funciones de control necesitamos los datos de configuración, los cuales son proporcionados por el alumno a través del cliente web. Así pues, los algoritmos del módulo *Instruments* aseguran que tanto los valores como el formato de estos datos puedan servir para controlar cualquier instrumento independientemente de su tipo y fabricante.

Los instrumentos cuyos datos son controlados por este módulo son: osciloscopio, generador de funciones, multímetro digital, fuente de alimentación continua y matriz de conmutación. Los datos de control de las matrices de conmutación empleadas para el control de los experimentos son validados por el algoritmo *Circuit Solver* que analizamos a continuación.

También este grupo de algoritmos enmarcados en el módulo *Instruments* se encarga de comprobar y validar que los datos de respuesta generados por los instrumentos del laboratorio y enviados a través del Servidor de Equipos y Experimentos pueden ser representados en el interfaz del cliente. La respuesta generada por los instrumentos también ha tenido que ser modificada, ya que al estar controlados mediante las funciones proporcionadas por los drivers IVI, debían ser formateadas para que pudieran ser mostradas correctamente en el cliente web.

Por su parte el conjunto de algoritmos denominados *Circuit Solver* tiene como función chequear si el circuito que se envía al laboratorio se puede construir con los componentes de que dispones en los experimentos o no. Debido a que este algoritmo ha sido desarrollado específicamente para la metodología de control propuesta lo vamos a detallar más en profundidad en la sección 3.3.1.2.

Una vez que se ha validado que es posible configurar los instrumentos en función de los datos asignados a cada uno de ellos por el cliente y que el circuito se puede construir físicamente con los componentes del laboratorio y las conexiones habilitadas por el instructor o profesor, se ejecuta el algoritmo denominado *Protocol*. Este algoritmo crea los paquetes de información que serán enviados al Servidor de Equipos y Experimentos con los datos de configuración de los instrumentos y del circuito que se desea crear. El formato de los datos y su significado está definido en base a un protocolo de comunicación descrito en la sección 3.2.3 en la que se analiza la comunicación entre ambos servidores.

Los algoritmos *Request* y *Response* tienen como función generar la petición de configuración de los equipos que se va a ejecutar en el Servidor de Equipos y Experimentos, y recibir su respuesta respectivamente. Por tanto ambos algoritmos se encargan de escribir y leer los paquetes de datos enviados a través del socket creado entre las aplicaciones del Servidor de Medidas y el Servidor de Equipos y Experimentos.

Por su parte el algoritmo *LXICom* establece y gestiona la comunicación entre ambos servidores. Además, este módulo tiene como función adecuar la respuesta proporcionada por los instrumentos al protocolo establecido para la comunicación con el cliente. Por esta razón y para adaptar el Servidor de

Medidas al método de control propuesto se ha desarrollado este módulo nuevo.

Pero la razón principal de disponer de este módulo es que tiene también por objetivo proporcionar al Servidor de Equipos y Medidas la información necesaria para construir físicamente los circuitos que el alumno ha realizado en el interfaz web, una vez resuelto el *Circuit Solver*. Estas acciones las realiza concretamente la función *circuit.cpp*, que ha sido diseñada para adecuarse al funcionamiento a los nuevos algoritmos del *Circuit Solver* y del instructor virtual, el cual analizaremos también más adelante ya que ha sido rediseñado de nuevo para la solución propuesta.

De este modo en función de los algoritmos principales que definen la arquitectura del Servidor de Medidas, su secuencia de ejecución desde que le llega una petición por parte del cliente hasta que ésta se envía al Servidor de Equipos y Experimentos para su ejecución sobre los instrumentos y experimentos es la siguiente:

1. Autenticación: en cada petición realizada por el cliente el Servidor de Medidas verifica que se trata de un usuario validado. Cada usuario viene identificado por una cookie única, la cual va asociada inequívocamente a un único usuario.
2. Gestión del tiempo de conexión: el Servidor de Medidas se encarga de gestionar el tiempo de posesión de los recursos del laboratorio de cada uno de los usuarios conectados en cada momento al laboratorio, asignando a cada usuario un tiempo determinado que depende directamente del número de peticiones realizadas en el mismo instante sobre el laboratorio.
3. Validación: el Servidor de Medidas se encarga también de validar que los datos con los que el alumno quiere configurar los instrumentos están dentro de los valores permitidos. Además se comprueba que el circuito que el alumno ha construido en el interfaz del cliente se puede realizar físicamente en base a unos ficheros de configuración (*.max*). Posteriormente veremos en detalle cómo se realiza esta validación.

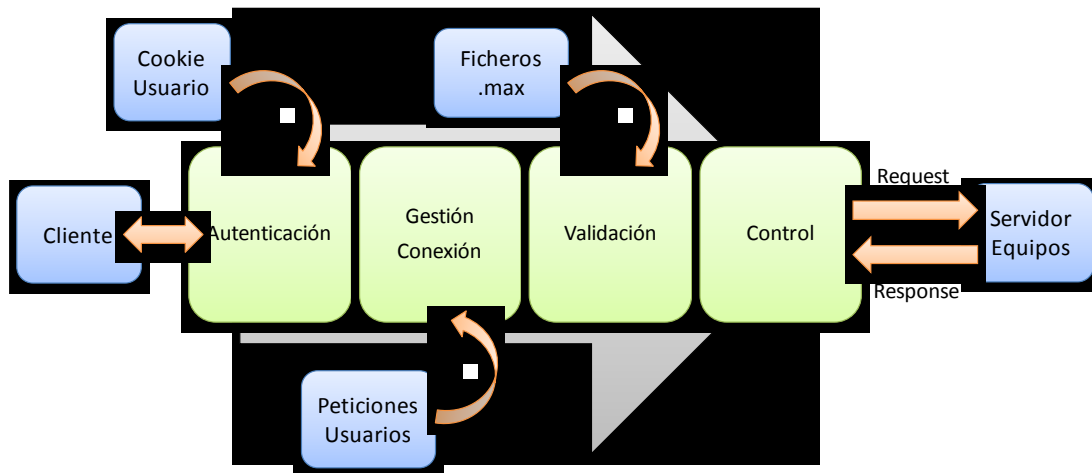


Figura 3.14: Secuencia de ejecución de procesos en el Servidor de Medidas

4. Envío de datos de configuración de los instrumentos y del experimento al Servidor de Equipos y Experimentos. Una vez que el usuario es identificado y sus peticiones son validadas, el Servidor de Medidas envía al Servidor de Equipos y Experimentos las peticiones de configuración a través de una conexión lógica o física si ambos servidores se encuentran ejecutándose en diferentes ordenadores, basada en la creación de un socket.
5. Lectura de la respuesta generada por el experimento, la cual es recogida por los instrumentos a través del Servidor de Equipos y Experimentos y enviada de vuelta al Servidor de Medidas para su envío al cliente web y que éste la muestre al alumno. Esta respuesta también es validada por el Servidor de Medidas para comprobar que los datos se podrán visualizar correctamente en el interfaz.

Gráficamente la Figura 3.14 muestra esta secuencia de ejecución, desde que llega una petición al Servidor de Medidas y ésta se envía una vez validada al Servidor de Equipos y Experimentos.

3.2.1.1 Configuración del Servidor de Medidas

Como se puede observar a lo largo de la descripción del Servidor de Medidas, para su correcto funcionamiento y para poder interoperar tanto con el cliente web como con el Servidor de Equipos y Experimentos es necesario configurar ciertos parámetros del servidor. Para ello existen dos tipos de ficheros de configuración:

- Ficheros `.conf`: son los ficheros `measureserver.conf` y `maxlists.conf`. El primero de ellos es utilizado para configurar el funcionamiento del Servidor de Medidas y permitir su comunicación tanto con el cliente web como con el Servidor de Equipos y Experimentos. El segundo fichero es utilizado por el instructor virtual y ejecutado por el *Circuit Solver*, para chequear la validez del circuito enviado desde el cliente.
- Ficheros `.max`: junto con el fichero `maxlist.conf`, los ficheros con extensión `.max` especifican y describen los circuitos que se pueden construir físicamente en el laboratorio. Los analizamos en el contexto del instructor virtual en la siguiente sección.

En primer lugar, se debe configurar el Servidor de Medidas para que pueda comunicarse con el cliente web. Ambas entidades se comunican bajo el paradigma cliente-servidor, por lo que en este caso en el Servidor de Medidas únicamente debemos especificar el puerto en el cual la aplicación encargada de gestionar la comunicación con el cliente escuchará sus peticiones y enviará las respuestas generadas en el laboratorio. De esta manera, en el fichero `measureserver.conf` se debe configurar el puerto de la siguiente manera:

<code>Port</code>	2324
<code>HTTPPort</code>	8080

Se especifican dos puertos, ya que el Servidor de Medidas permite la comunicación con el cliente, bien mediante sockets o bien mediante HTTP. En el primer caso, se indica mediante la variable `Port` el puerto a través de cual se efectuará la comunicación (en este ejemplo 2324), mientras que en el segundo caso el puerto será el 8080.

En el cliente web se deberán definir tanto la dirección IP en la cual se ejecuta el Servidor de Medidas, como el puerto donde la aplicación del servidor escucha las peticiones del cliente el cual coincidirá con el número de puerto que hemos configurado anteriormente. El fichero de configuración del cliente web se denomina `config.xml`, y en él se identifican los parámetros de comunicación con el Servidor de Medidas. A continuación vemos un ejemplo de este fichero de configuración, en donde se observa cómo el puerto se corresponde con el mismo que se ha configurado previamente en el fichero `measureserver.conf` y la dirección IP en la cual se ejecuta éste servidor es la 130.206.138.35:

```
<configuration>
<hostname>130.206.138.35</hostname>
<port>2324</port>
</configuration>
```

Como se detallará en la sección 3.2.3, la comunicación entre el Servidor de Medidas y el Servidor de Equipos y Experimentos se realiza a nivel de socket. En este caso el Servidor de Medidas actúa como cliente y el Servidor de Equipos y Experimentos como servidor ante las peticiones enviadas por el Servidor de Medidas.

Por tanto, en el fichero `measureserver.conf` deberemos configurar tanto la IP donde se ejecuta el Servidor de Equipos y Experimentos, como el número de puerto usado por las aplicaciones de ambos servidores para el intercambio de información. En el caso de la solución propuesta, el módulo encargado de realizar la comunicación con el Servidor de Equipos y Experimentos es el algoritmo *LXICom*, el cual utiliza los datos de configuración siguientes:

```
LXI module config
LXI.Host             localhost
LXI.Port             5001
LXI.RetryCount       4
LXI.RetryTimeout     10
```

En esta configuración la variable *LXI.Host* contiene la dirección IP de la máquina en la cual se ejecuta el Servidor de Equipos y Experimentos. En el caso de que ambos servidores se ejecuten en la misma máquina, esta dirección podrá ser la de *loopback* (127.0.0.1) o lo que es lo mismo el nombre reservado *localhost*. La variable *LXI.Port* contiene el puerto a través del cual se comunicarán las aplicaciones de ambos servidores, que en este caso corresponde al puerto 5001. Las variables *LXI.RetryCount* y *LXI.RetryTimeout* determinan en la configuración del socket cuántas veces se trata de establecer la conexión y el tiempo en segundos que el Servidor de Medidas espera para reintentar la conexión en caso de que ésta no se realice con éxito, respectivamente.

Así quedan configuradas las comunicaciones entre el cliente web y el Servidor de Medidas, y entre éste y el Servidor de Equipos y Experimentos. Pero tal y como decíamos, el fichero `measureserver.conf`, también sirve para configurar ciertos parámetros relativos a la operativa del Servidor de Medidas.

El Servidor de Medidas se encarga también de la gestión de las peticiones realizadas por el cliente web. Para ello, el servidor se configura para admitir un número máximo de peticiones realizadas al mismo tiempo sobre el laboratorio, y el número máximo de alumnos que pueden estar dentro del laboratorio a la vez. Estos dos parámetros, fundamentales para el buen funcionamiento del laboratorio son definidos en el fichero `measureserver.conf`:

```
MaxClients 16
MaxSessions 50
```

La variable `MaxClients` determina el número de conexiones de red simultáneas permitidas por el servidor, mientras que la variable `MaxSessions` especifica el número de sesiones iniciadas y que permanecen activas sobre el servidor al mismo tiempo.

Adicionalmente, este fichero de configuración permite definir la ubicación de los ficheros necesarios para el buen funcionamiento del servidor:

```
CompTypes      component.types
MaxListConfig  maxlists.conf
SaveCircuits   savedcircuits/
LoadModules    lxicom
LogDir         logs
```

La variable `CompTypes` identifica el fichero que contiene los tipos de componentes que pueden ser usados para la creación de circuitos. En este caso, tal y como explicábamos anteriormente este fichero se denomina `component.types`.

Por su parte la variable `MaxListConfig` determina el nombre del fichero encargado de identificar los ficheros `.max` utilizados por el instructor virtual que analizaremos posteriormente. En este caso, el fichero se denomina `maxlists.conf`.

La variable `LoadModules` es utilizada por el fichero de configuración para indicarle al Servidor de Medidas que tiene que ejecutar ciertos módulos desarrollados especialmente para el laboratorio. En este caso, tal y como hemos explicado anteriormente, se ha desarrollado un módulo específico denominado `Lxicom` para hacer posible el funcionamiento del Servidor de Medidas mediante la metodología de control propuesta.

Finalmente las variables *LogDir* y *SaveCircuits* permiten indicar respectivamente los directorios en los cuales se almacenan los ficheros que permiten llevar un registro de las acciones llevadas a cabo en el laboratorio y guardar los circuitos diseñados en el cliente web.

3.2.2 Servidor de Equipos y Experimentos

Cómo venimos diciendo a lo largo de este documento, es el Servidor de Equipos y Experimentos en donde se ejecutan los algoritmos destinados a controlar los instrumentos y experimentos disponibles en el laboratorio remoto. Es decir, es este servidor el que se encarga de recoger las peticiones realizadas por el usuario en el cliente web, que previamente han sido validadas por el Servidor de Medidas, y ejecutarlas sobre los equipos disponibles en el laboratorio. Este servidor también dispone de las rutinas necesarias para leer los resultados del experimento y devolvérselos al cliente, a través del Servidor de Medidas, el cual se encarga de adecuar estas respuestas para que puedan ser comprensibles por el usuario a través del cliente web dispuesto a tal efecto.

El Servidor de Equipos y Experimentos está desarrollado íntegramente en LabVIEW, principalmente por las siguientes razones:

- LabVIEW dispone de un conjunto de herramientas y utilidades que facilitan enormemente el control y la monitorización de los instrumentos que son gestionados mediante las aplicaciones ejecutadas desde el servidor.
- Proporciona un interfaz gráfico de programación fácil e intuitivo que hace posible una programación muy estructurada y modular. Estas características son ideales en una arquitectura de control como la que aquí planteamos, la cual tiene que estar preparada para ser fácilmente actualizable y escalable con nuevos instrumentos.
- La metodología que se presenta en este trabajo está basada en la utilización de las funciones proporcionadas por los drivers IVI para el control de los instrumentos y LXI como interfaz de comunicaciones entre ellos. El entorno de programación en LabVIEW está perfectamente adaptado para trabajar con estas tecnologías y proporciona también funciones específicas para ello, lo que facilita el control de cada dispositivo gestionado por el servidor. Esta

característica hace que añadir nuevos instrumentos o adecuar, si fuera necesario, el código a eventuales cambios, pueda ser realizado de manera sencilla y rápida sin tener porqué comprender la estructura completa de funcionamiento del servidor.

- Los drivers compatibles con IVI definidos por el fabricante de los instrumentos que hemos utilizado para validar este método de control están preparados para ser accedidos desde LabVIEW, lo cual junto con las dos primeras características aquí expuestas, nos ha hecho inclinarnos por utilizar este entorno de programación.
- Para realizar la configuración de los instrumentos necesitamos relacionar, entre otros aspectos, el driver del instrumento con la aplicación que hace uso del mismo. Para ello, National Instruments proporciona una herramienta denominada Measurement & Automation, la cual se encuentra perfectamente integrada con LabVIEW, facilitando la configuración de ciertas variables de entorno fundamentales para acceder a la programación remota de los instrumentos.
- Existe una amplia red de desarrolladores en LabVIEW¹³, lo que permite acceder a una gran cantidad de información que nos puede ayudar a resolver los problemas que pueden aparecer tanto a la hora de realizar la configuración de los equipos, como a implementar cualquier tipo de algoritmo realizado mediante esta tecnología.

Previamente a comenzar la codificación del algoritmo que implementa el método de control ejecutado sobre el Servidor de Equipos y Experimentos, hemos diseñado la arquitectura lógica del mismo, la cual se muestra en la Figura 3.15.

Al igual que realizábamos con el Servidor de Medidas, en esta figura se pueden distinguir en color azul todos aquellos módulos que han sido diseñados y definidos completamente para adaptar el servidor al método de control propuesto. Se observa cómo únicamente se ha mantenido el módulo encargado de gestionar la comunicación con el Servidor de Medidas.

¹³ <http://zone.ni.com/dzhp/app/main>

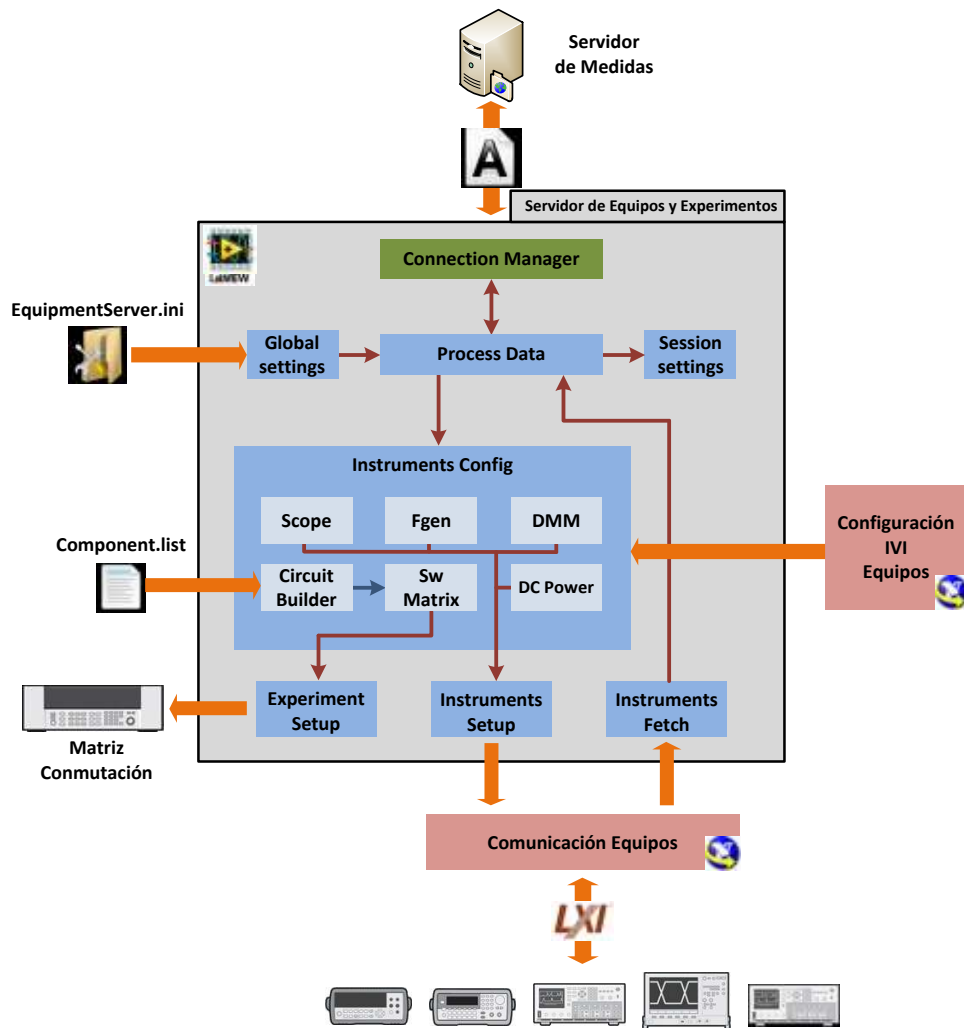


Figura 3.15: Arquitectura lógica del Servidor de Equipos y Experimentos

El Servidor de Equipos y Experimentos se comunica por una parte con el Servidor de Medidas y por otra con los instrumentos y experimentos del laboratorio, los cuales se encuentran controlados a su vez por la matriz de conmutación.

Tal y como ya se indicaba en la Figura 3.2, el Servidor de Equipos y Experimentos se comunica con los instrumentos a través de una red de comunicaciones Ethernet mediante el interfaz de control LXI que éstos equipos implementan. Como podemos ver en el Anexo A, será la aplicación Measurement & Automation (MAX), proporcionada e integrada en el entorno de desarrollo de LabVIEW, la que utilizaremos para configurar tanto esta red LAN como la configuración de los drivers IVI de cada uno de los instrumentos disponibles en el laboratorio.

En esta arquitectura lógica el módulo *Connection Manager* contiene aquellos procesos que se ejecutan en el servidor y tiene como objetivo gestionar la comunicación desde y hacia el Servidor de Medidas. Es decir, este módulo se encarga de procesar cada una de las peticiones que le llegan a través del socket abierto con el Servidor de Medidas y proporcionarle a su vez la respuesta generada por cada uno de los instrumentos del laboratorio.

El tipo de experimentos que se ejecutan en este laboratorio es por lotes ya que una vez que el usuario ha configurado el experimento en el cliente web, realiza la petición al laboratorio y éste le devuelve la respuesta generada por el hardware del laboratorio de manera inmediata. Este conjunto de acciones se realiza rápidamente, de modo que el usuario tiene la percepción de realizar un experimento interactivo y prácticamente en tiempo real. Pues bien, quien se encarga de gestionar estas conexiones y mantener ese sincronismo usuario-petición-respuesta-usuario es el módulo *Connection Manager* implementado en el Servidor de Equipos y Experimentos en colaboración con el algoritmo *Network* que analizábamos en el Servidor de Medidas.

Una vez configurada y establecida la comunicación entre ambos servidores, los datos enviados son recibidos por el Servidor de Equipos y Experimentos y son procesados por el módulo *Process Data*, el cual contiene los algoritmos encargados de procesar el paquete de datos enviado desde el Servidor de Medidas, y que pueden ser de tipo *"data"* o *"information"*. Los paquetes tipo *"error"* pueden ser generados por el servidor como respuesta a una petición que no se puede procesar y generándose así un mensaje de error. Una vez determinado el tipo de información solicitada por el Servidor de Medidas, se ejecutarán las acciones pertinentes en cada caso: si es tipo *"information"* se devolverá información sobre el estado del servidor, y si es tipo *"data"*, se procederá a configurar los instrumentos en función de las acciones solicitadas por el usuario.

Pero, previamente a realizar cualquier acción sobre los instrumentos del laboratorio, es necesario haber configurado una serie de variables globales que contienen datos de configuración del servidor e información que permite acceder a los instrumentos. Estas variables globales se encuentran todas agrupadas en el módulo denominado *Global Settings* y

obtienen sus valores del fichero de configuración *EquipmentServer.ini* que analizaremos en la sección 3.2.2.1.

Así, una vez que tanto el servidor como las referencias a los instrumentos se encuentran configuradas, se crean para cada sesión del laboratorio una serie de variables denominadas *Session Settings* encargadas de contener información de cada uno de los instrumentos del laboratorio. Como indicábamos, los experimentos ejecutados son por lotes aunque debido a la rapidez con la que se ejecutan las acciones solicitadas por el usuario, éste percibe sensación de estar interactuando con el laboratorio en tiempo real. Parte de esta velocidad se consigue gracias a que se ha implementado un algoritmo que se encarga de comparar la petición de configuración del laboratorio enviada por el usuario con el estado del laboratorio en ese momento. Pues bien, el estado del laboratorio después de cada configuración se almacena en estas variables de sesión, de modo que comparando la información contenida en las variables *Session Settings* con los datos recibidos, se determina si se puede ganar tiempo en generar la respuesta, pues puede que la configuración sea la misma previamente realizada o muy similar.

Esta comparación se hace instrumento a instrumento. En cada petición enviada desde el cliente se incluye la información de todos los instrumentos del laboratorio. Se hace de este modo porque generalmente una vez que el circuito que se quiere testear se ha enviado al laboratorio, las siguientes peticiones son pequeñas modificaciones respecto a la primera, por lo que los cambios a efectuar sobre los instrumentos son mínimos. De este modo se gana tiempo configurando únicamente los cambios respecto a la petición anterior y generar y enviar al usuario una respuesta casi de forma instantánea.

Una vez procesado el tipo de paquete de información enviado desde el Servidor de Medidas, se realiza la configuración de los equipos del laboratorio. Para ello se ha definido el módulo *Instrument Config*, el cual en función del identificativo de cada uno de los instrumentos contemplados en el laboratorio, ejecuta el algoritmo de control de cada uno de ellos.

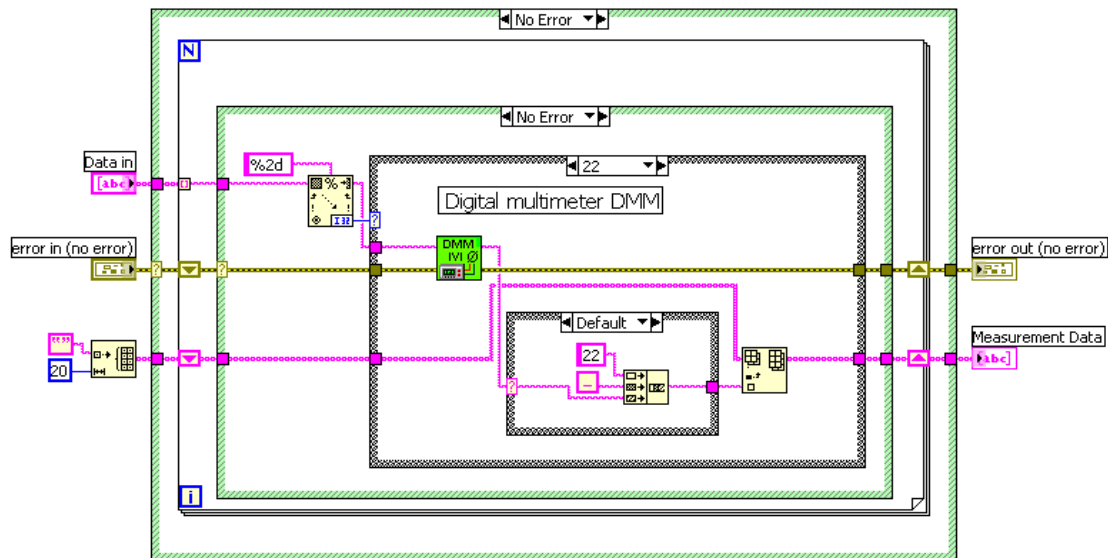


Figura 3.16: Llamada al algoritmo de control de cada instrumento

Se ha implementado este módulo de tal manera que añadir la lógica de control de un nuevo instrumento no requiere la recodificación de ninguno de los algoritmos ya implementados, solamente incluir la llamada a este nuevo algoritmo desde el *Instrument Config*.

En la Figura 3.16 podemos ver la llamada al algoritmo encargado de controlar el multímetro presente en el laboratorio. Partiendo de un String de datos de entrada denominado “Data in”, se selecciona la parte de texto correspondiente al *InstrumentID* y se realiza mediante una estructura *Case* la llamada al algoritmo de control correspondiente, que en este caso al tratarse de un *InstrumentID=22*, corresponde al multímetro o DMM.

Una vez realizada la configuración del instrumento mediante los algoritmos correspondientes a cada uno de ellos, y recogidos en el módulo *Instruments Setup* de la Figura 3.15, se devuelve al Servidor de Medidas un *string* de datos que contiene la respuesta generada por el instrumento, la cual aparece reflejada en la Figura 3.16 mediante la variable “Measurement Data”.

Esta estructura se repite para cada uno de los instrumentos disponibles en el laboratorio remoto, que en el caso del empleado para validar el método de control y medida propuesto son: osciloscopio, multímetro, generador de funciones, fuente de alimentación y matriz de conmutación.

Todos estos instrumentos, excepto la matriz de conmutación, siguen el mismo proceso de configuración: en función de los datos recibidos en el *string* de entrada a continuación del *InstrumentID*, se configura el instrumento, se realizan las medidas y se devuelven las lecturas al Servidor de Medidas mediante otro *string* de datos enviado sobre el socket existente entre ambos servidores.

Además, la configuración de los instrumentos, se realiza siempre en el mismo orden: matriz de conmutación, multímetro, generador de funciones, osciloscopio, y fuente de alimentación. Este orden guarda una lógica que tiene que ver con los pasos que se realizan en el laboratorio tradicional al construir el mismo circuito: se crea el circuito y a continuación se conectan los instrumentos para poder posteriormente realizar las medidas oportunas.

La matriz de conmutación, o mejor dicho el mainframe que contiene las matrices de conmutación, es el equipo encargado de interconectar los componentes, así como de conectar los instrumentos en el circuito bajo test en los lugares seleccionados por el usuario. Para ello necesita saber qué componentes hay disponibles y dónde se encuentran colocados. Esta información reside en el fichero de configuración denominado *Component.list*.

Este fichero junto con los datos de salida del algoritmo *Circuit Solver* del Servidor de Medidas son usados por el conjunto de algoritmos que conforman el *Circuit Builder* para definir las conexiones que se deben realizar en la matriz de conmutación. El proceso completo está explicado en la sección 3.3.1.2.

Así pues, en función de los algoritmos principales diseñados y desarrollados para definir la arquitectura del Servidor de Equipos y Experimentos, su secuencia de ejecución (Figura 3.17) desde que le llega un paquete de datos a través del socket de comunicación establecido con el Servidor de Medidas, hasta que le devuelve a éste la respuesta capturada por cada uno de los instrumentos del laboratorio será:

1. Establecer las variables de cada sesión: partiendo de la configuración de cada uno de los instrumentos, se almacena su estado en variables de sesión, de manera ante cada petición de configuración solicitada por el usuario, se comprueba el estado de los instrumentos, para evitar redundancia en las configuraciones.

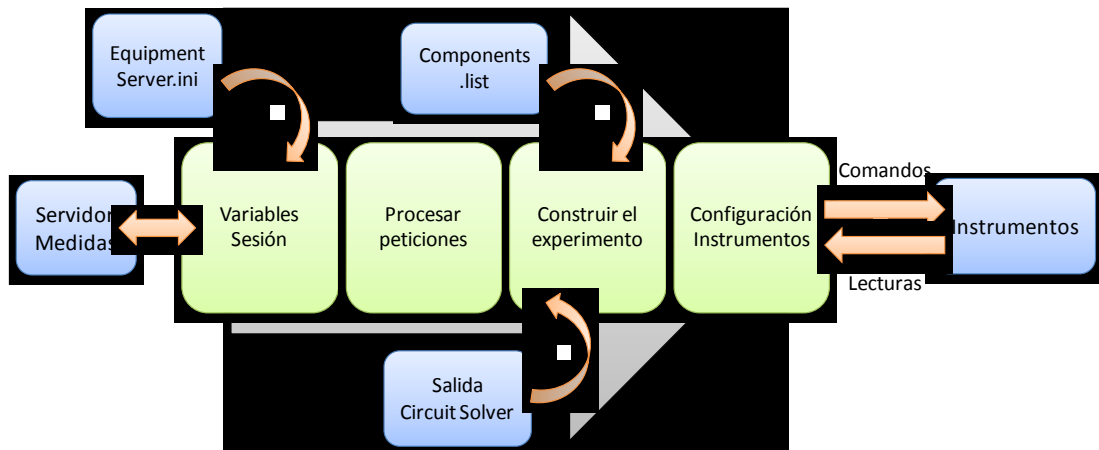


Figura 3.17: Secuencia de ejecución de procesos del Servidor de Experimentos y Equipos

2. Procesar peticiones: se analiza el paquete de datos enviado desde el Servidor de Medidas. En función de la información que contenga se configuran los instrumentos del laboratorio.
3. Construcción del experimento: partiendo del paquete de datos enviado por el Servidor de Medidas en el cual se describe el fichero realizado por el usuario en el cliente web (recordemos que es la salida del *Circuit Solver*) y del fichero de configuración *Components.list*, el algoritmo *Circuit Builder* obtiene las instrucciones necesarias para controlar la matriz de conmutación y obtener la implementación física de dicho circuito.
4. Configuración de los instrumentos: se realiza la configuración de los instrumentos, previa comprobación del estado de cada uno de ellos. La matriz de conmutación realiza las conexiones necesarias entre componentes e instrumentos en función de la descripción del circuito enviada desde el Servidor de Medidas y el fichero *components_IVI.list*.
5. Lectura de los instrumentos: una vez implementado el circuito físico a través de la matriz de conmutación, se procede a realizar la lectura de medidas sobre el circuito bajo prueba.
6. Envío de respuesta al Servidor de Medidas: cada uno de los instrumentos genera una respuesta, la cual puede ser correcta o puede ser un error. En el caso de ser correcta, cada instrumento devolverá bien su estado de configuración, en el caso del generador de funciones y de la fuente de alimentación, bien la lectura de las medidas tomadas, en el caso del osciloscopio y el multímetro, o bien devolverá un código

indicando que ha ido todo bien, en el caso de la matriz de conmutación. Si sucede un error en la configuración de alguno de los equipos, se devolverá un paquete tipo error al Servidor de Medidas, el cual contiene el código de error generado por el instrumento y la descripción de dicho error.

3.2.2.1 Configuración del Servidor de Equipos y Experimentos

Tal y como venimos exponiendo a lo largo de todo este documento la finalidad del método de control propuesto es permitir que cada laboratorio remoto disponga de sus propios instrumentos, según su disponibilidad, sin estar atado a una solución o unos instrumentos específicos.

Para ello se ha explicado en secciones anteriores cómo hemos realizado el Servidor de Equipos y Experimentos de modo que en ninguno de los algoritmos de control implementados para cada equipo se accede directamente a un instrumento concreto, sino que se emplean referencias creadas para cada instrumento durante el proceso de configuración de los drivers gracias a las funciones proporcionadas por los drivers IVI. En el Apéndice A podemos ver cuáles son estas funciones así como ejemplos de las mismas.

Para hacer aún más sencilla esta configuración y aumentar la modularidad del sistema, se ha definido un fichero de configuración denominado *EquipmentServer.ini* cuyo cometido es contener, entre otra información, los Nombres Lógicos de todos los instrumentos disponibles en el laboratorio remoto, los cuales son empleados en los algoritmos de control para acceder a los instrumentos. De este modo no se accede a un instrumento en particular, si no a un instrumento genérico referenciado por este Nombre Lógico.

Como se indica en el ejemplo siguiente, este fichero dispone de varios apartados, todos ellos destinados a informar sobre la configuración del Servidor de Equipos y Experimentos. El [*Instrument Address*] proporciona al servidor los Nombres Lógicos de los instrumentos a controlar. Esta información es leída y almacenada en las variables *Global Settings* a las que se hacía referencia en la Figura 3.15. En este ejemplo se observa cómo todas las referencias a los equipos se realizan a través de Nombres Lógicos, los cuales

hemos configurado previamente: *MyFgen*, *MyPowerDC*, *MyScope*, *MySwitch*, *MyDMM*.

```
[Global]
  Port=5001
  Reset Time = 300 [s]
  Maximum Delay = 1000 [ms]

[Log]
  Log Level = 4
  Log File = C:\MeasuereServer.log

[Instrument Address]
  Function Generator 1 = MyFgen
  DC Power Supply 1 = MyPowerDC
  Digital Oscilloscope 1 = MyScope
  Switching Matrix 1 = MySwitch
  Digital Multimeter 1 = MyDMM

[Component type]
  Component type = R:2,C:2,VDCCOM:1,OP:8,VFGENA:2,SHORTCUT:2,VDC25V:1,
  VDC-25V:1,VDC6V:1,VFGENB:2,L:2,5VA:2,5VB:2,Q:3,BLACKBOX:3,

[Components list]
  Components list = Components.list

[Matrix version]
  Matrix version = IVI_Matrix 1.0
```

De manera resumida la Figura 3.18 representa cada uno de los agentes definidos para realizar la configuración de un instrumento mediante la metodología aquí propuesta. Modificando los parámetros de configuración que podemos ver en el Apéndice A no es necesario realizar ningún cambio en los algoritmos de control ante cambios de los instrumentos físicos.

En el fichero de configuración *EquipmentServer.ini* también encontramos otros parámetros necesarios para el buen funcionamiento del sistema. Recordamos que el Servidor de Medidas está comunicado con el Servidor de Equipos y Experimentos mediante un esquema cliente-servidor. Por ello, es necesario configurar los parámetros correspondientes al servidor en un paradigma como el aquí planteado, los cuales están agrupados bajo la etiqueta [*Global*] y especifican:

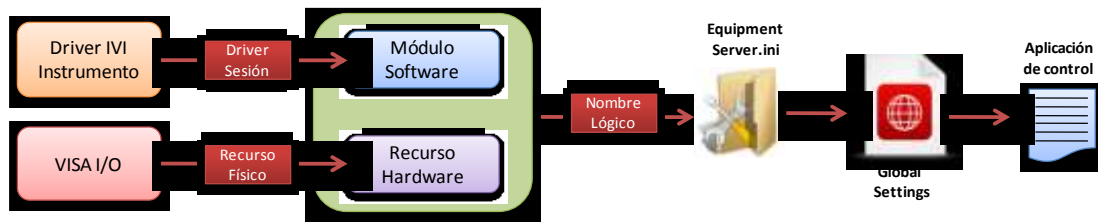


Figura 3.18: Pasos para la configuración del Servidor de Equipos y Experimentos

- El puerto en el cual el servidor escucha peticiones del cliente, en este caso el Servidor de Medidas. Como vemos este puerto es el 5001, el mismo que se configuraba en el fichero *measureserver.conf* en el módulo LXI.
- Los parámetros *Reset Time* y *Maximum Delay* son parámetros de configuración del socket establecido entre ambos servidores y especifican el tiempo de espera antes de reiniciar la comunicación en caso de que el servidor no responda, y el tiempo máximo de espera ante una respuesta del servidor.

A continuación bajo la etiqueta [Log] se indica el nombre del fichero utilizado para registrar la información intercambiada entre ambos servidores, así como cantidad de información que queremos que quede registrada. Este volumen de información va indicado en la variable *Log Level*, y puede tener un valor de 1 (mínima información registrada) a 5 (se registra toda la información que se envía a través del socket entre ambos servidores).

Como vimos anteriormente el Servidor de Medidas hacía uso de un fichero de configuración denominado *component.types* para obtener información acerca de los tipos de componentes disponibles en el laboratorio. El Servidor de Equipos y Experimentos también necesita esta información para determinar las instrucciones que tiene que enviar a la matriz de conmutación para incluir el componente en el circuito. Los datos que necesita el servidor se le proporcionan mediante la variable *Component type*, en donde se especifica el tipo de componente (obviamente debe coincidir con el definido en el fichero homólogo del Servidor de Medidas), seguido del número de terminales de dicho componente. Es decir, si se escribe R:2 estamos definiendo el componente resistencia el cual dispone de dos terminales y por tanto se conecta al circuito mediante dos conexiones. En el caso del amplificador operacional, OP:8, se trata de un circuito integrado de 8 terminales, por lo que tendrá 8 posibles conexiones en el circuito.

A continuación, bajo la etiqueta [*components list*] definimos el otro fichero necesario para que funcione correctamente el Servidor de Equipos y Experimentos, pues contiene la información acerca de qué componentes se encuentran emplazados en la matriz de conmutación, y en qué lugar de la misma están conectados. En el ejemplo, este fichero se denomina *Components.list*. La información incluida en este fichero depende directamente de la configuración realizada en la matriz de conmutación, por lo que lo describimos conjuntamente en las secciones 3.3.1.2 y 3.3.1.3.

La última información que encontramos en el fichero de configuración bajo la etiqueta [*Matrix version*] hace referencia a la versión de la matriz de conmutación utilizada por el servidor para la construcción del circuito bajo prueba diseñado por el alumno en el cliente web.

3.2.3 Protocolo de comunicación entre ambos servidores

Si bien se ha partido de la arquitectura basada en dos servidores, Servidor de Medidas y Servidor de Equipos y Experimentos, al haber modificado por completo los algoritmos de control de los instrumentos ejecutados en el Servidor de Equipos y Experimentos y al estar basado este control en las funciones proporcionadas por los drivers IVI, se ha tenido que adecuar el formato de los datos intercambiados entre ambos servidores a los requisitos impuestos por estas funciones.

Tal y como se explicaba previamente, en el caso del Servidor de Medidas son los algoritmos *Request* y *Response* los encargados de gestionar esta comunicación, siendo el algoritmo *TCP.vi* el encargado de gestionar las comunicaciones por parte del Servidor de Equipos y Experimentos.

Este protocolo de intercambio de datos ha sido rediseñado para adaptarlo a las especificaciones del estándar IVI para cada una de las familias de instrumentos y poder funcionar correctamente con el conjunto de algoritmos de control diseñados en este trabajo de investigación. Esto quiere decir, por ejemplo, que para el caso del multímetro digital la especificación IVI correspondiente a esta familia [IVI10b] determina sus atributos de configuración. De este modo, el protocolo contempla estos mismos atributos, los cuales también se han incluido para el resto de instrumentos contemplados en la solución propuesta, en base a sus propias definiciones en el estándar IVI.

Instrumento	Valor Atributo <i>InstrumentID</i>
Generador de Funciones	11
Fuente de Alimentación DC	12
Osciloscopio	21
Multímetro Digital	22
Matriz de Conmutación	41

Tabla 3.6: Identificadores de los instrumentos en el método propuesto

Como ya se mencionaba anteriormente, los paquetes de información definidos para ser intercambiados entre ambos servidores pueden ser de tres tipos: datos, información y error, siendo su sintaxis la siguiente:

```
<Length><\n><Type><\n><Content>
```

El atributo *Length* hace referencia a la longitud del paquete enviado, medido en número de caracteres de texto. El atributo *Type* indica el tipo de información enviada (*data*, *information*, *error*), y por último el atributo *Content* contiene la información o carga útil enviada entre aplicaciones.

Por tanto en el cuerpo o la carga útil del mensaje en donde reside la información intercambiada entre las aplicaciones. Para identificar al instrumento al que va dirigido la información o el instrumento que ha generado una respuesta, el primer dato enviado en el atributo *Content* es el correspondiente al atributo definido como *InstrumentID*.

En el caso de la metodología de control propuesta, únicamente se han considerado los instrumentos que aparecen en la Tabla 3.6 en la cual aparece en la columna de la derecha el identificativo asignado a cada uno de ellos y que se corresponde con el atributo *InstrumentID*.

A continuación del atributo *InstrumentID* se envían bien los datos de configuración del instrumento, en caso de tratarse una petición, o bien la respuesta del instrumento ante la petición previamente realizada. A modo de ejemplo en el caso del generador de funciones, si se trata de un paquete de

petición tipo *data*, es decir de configuración del equipo, los atributos de la información que seguiría al *InstrumentID* serían los siguientes:

```
<Function><Waveform><Amplitude><Frequency><DCOffset><Startphase><Trigger  
mode><Trigger source><Burst count><Duty Cycle High><User defined waveform>
```

El atributo *Function* hace referencia a la acción que se quiere realizar sobre el instrumento. Podrá ser de dos tipos *Setup*, en cuyo caso su valor será '0' e implica una configuración de los equipos, o bien *Fetch*, tomando el valor '1', en cuyo caso se pide una lectura de los parámetros con los que están configurados los equipos. El método de control diseñado está implementado de manera que después de la de configuración del instrumento (*Function=Setup*), se ejecuta siempre una lectura del mismo (*Function=Fetch*), ya que en los laboratorios la configuración de un instrumento siempre se realiza con objeto de realizar a continuación una medida sobre el experimento bajo prueba, lo que en el laboratorio remoto se traduce en una lectura de datos del instrumento. Este atributo y sus valores son comunes a todos los instrumentos del laboratorio y contemplados para todos ellos en la metodología de control propuesta.

Los valores posibles para el resto de estos atributos están establecidos por el estándar IVI dedicado a la familia de los generadores de funciones [IVI09], de modo que el atributo *Waveform*, el cual describe la forma de onda generada, puede presentar los valores definidos en la Tabla 3.7.

El atributo *Amplitude* hace referencia a la diferencia existente entre el valor máximo y el valor mínimo de tensión de la señal generada. En este caso, para evitar posibles fallos o el mal funcionamiento del instrumento, el valor máximo permitido es de 10V. Si se definen valores máximos y mínimos para los atributos, como sucede con éste, el valor enviado por el cliente es chequeado previamente por el Servidor de Medidas, mediante el algoritmo correspondiente al generador de funciones, perteneciente al conjunto de algoritmos *Instruments*, al que se hace referencia anteriormente.

Atributo <i>waveform</i>	Valor
Señal senoidal	1
Señal cuadrada	2
Señal triangular	3
Rampa ascendente	4
Rampa descendente	5
Señal DC	6

Tabla 3.7: Formas de onda definidas por el estándar IVI para generadores de señales.

El atributo *Frequency* indica la el número de ciclos por segundo de la señal generada, cuyo valor máximo en caso de ser una señal cuadrada o senoidal es de 20MHz y, de 1MHz en el resto de casos.

El atributo *DC Offset* indica la diferencia entre el valor medio de los valores máximo y mínimo de la señal generada, respecto a 0V. En este caso, el límite establecido está comprendido entre los -5V y 5V siempre y cuando se cumpla la igualdad $|DC\ Offset + Amplitud| < 10V$.

El atributo *Start Phase* especifica el offset horizontal de la señal generada, teniendo por unidades grados de un ciclo de la señal. Sus valores deben estar comprendidos entre -180° y 180° .

El atributo *Trigger Mode* especifica el modo en el cual se genera la señal de salida del generador de funciones, mientras que el atributo *Trigger Source* indica la fuente que dispara el evento de generación de la señal de salida. Los valores de estos atributos pueden ser consultados en la especificación del estándar [IVI09].

El atributo *Burst Count* permite configurar el número de ciclos de señal que se quieren generar. Si su valor es 0, está desactivado.

El atributo *Duty Cycle High* indica en una onda cuadrada el porcentaje de tiempo correspondiente a un ciclo de la señal, para el cual la señal está en estado alto. Su valor por defecto es de 50%.

Por último, el atributo *User defined Waveform*, permite definir al usuario una forma de onda arbitraria. Su valor por defecto es 0.

A modo de ejemplo, supongamos que el alumno a través del interfaz web genera una señal cuadrada de 6V de amplitud, 10kHz de frecuencia, fase inicial 0°, un offset de +0,5V, y un ciclo de trabajo del 45%. En este caso, una vez que el Servidor de Medidas valide estos datos en función de los límites que previamente definidos, se enviará una petición de configuración al Servidor de Equipos y Experimentos, la cual llevara el siguiente paquete de información:

```
000038data\n11 0 2 10000 0,5 0 1 0 0 0,45 0\n
```

En una primera parte del mensaje se envía la longitud del paquete de datos y se indica que se trata de un paquete tipo *data*. A continuación se envía el cuerpo del mensaje, separando mediante espacios los valores correspondientes a cada uno de los atributos. En este caso, el 11 indica que el paquete va dirigido al generador de funciones con objeto de configurarlo (indicado por el 0 a continuación del 11) para generar una señal cuadrada (indicado por el 2), de 10kHz, con un offset de +0,5V, fase inicial 0, un modo de disparo continuo (1) que genera la señal de salida internamente (0), y con un ciclo de trabajo de 0,45 (equivalente al 45% definido por el usuario).

Cada paquete de datos puede albergar varias peticiones o respuestas, a uno o varios equipos, pero siempre separados mediante el limitador “/n”. El siguiente ejemplo, muestra un paquete de datos enviado entre el Servidor de Medidas y el Servidor de Equipos y Experimentos, en el cual se solicita la configuración y lectura de datos del osciloscopio (*InstrumentID=21*) y a continuación la configuración del generador de funciones (*InstrumentID=11*). En el ejemplo se han separado en diferentes líneas las peticiones sobre cada instrumento, con el objetivo de diferenciarlas unas de otras:

```
000112data\n
21 0 1 20 50 1000 1 0 10 0 1 0 0 0 0 1 0,0 0,0 2 1 0 2 0 2 0 2\n
21 1\n
11 0 2 10000 0 0 1 0 0 0,5 0\n
```

Si la configuración del instrumento se ha producido satisfactoriamente, el Servidor de Equipos y Experimentos devuelve al Servidor de Medidas un paquete de datos en cuya carga útil aparece un ‘0’ por cada uno de los equipos sobre los que se ha solicitado una configuración. Es decir, en el

ejemplo anterior, el Servidor de Equipos y Experimentos, antes de devolver un paquete de datos con las lecturas de las medidas realizadas, devolvería un paquete de datos de la siguiente manera:

```
21 0\n → configurado satisfactoriamente el osciloscopio
11 0\n → configurado satisfactoriamente el generador de funciones
```

En este mismo ejemplo, si en vez de solicitar una petición de configuración al generador de funciones se realizase una petición de lectura del estado de sus variables (*Fetch*), es decir, si el atributo *Function* tuviera valor '1', el formato de la respuesta sería igual, indicando en cada uno de los atributos su valor correspondiente en el siguiente orden:

```
<InstrumentID><Function><Waveform><Amplitude><Frequency><DCOffset><Start
phase><Trigger mode><Trigger source><Duty Cycle High>
```

Cada uno de los atributos definidos para cada instrumento tendrá un valor concreto, los cuales vienen determinados por las funciones de control proporcionadas por los drivers IVI y utilizadas por los algoritmos de control del Servidor de Equipos y Experimentos. De este modo, cualquier instrumento de la misma familia ejecutará las mismas acciones y producirá las respuestas con idénticos valores en base a los comandos y valor de los atributos enviados desde el Servidor de Medidas al Servidor de Equipos y Experimentos. Este es un factor clave en la solución que proponemos en este trabajo de investigación, pues así nos aseguramos que el mismo protocolo de comunicación establecido entre ambos servidores es válido para cualquier instrumento contemplado en la metodología de control propuesta.

Se puede observar como no entran en juego en ningún momento comandos o sintaxis que sean exclusivos del fabricante de los equipos, como sucedía en soluciones que hemos analizado en el capítulo anterior y que dependían de la sintaxis y comandos dictados, por ejemplo por el estándar SCPI, por lo que mediante esta solución contribuimos también a independizar el hardware del laboratorio del proveedor de equipos, abstrayendo el software de control de los instrumentos y circuitos bajo prueba de los experimentos, ya que como se indicaba previamente, las matrices de conmutación utilizadas para la implementación remota de los circuitos son controladas como un instrumento más del laboratorio.

3.3 Sistema de control y configuración de experimentos de la solución propuesta

En esta sección son descritos cada uno de los elementos que forman parte del sistema de control y configuración de experimentos utilizados en el método de control propuesto, cuyos objetivos funcionales son:

- a) Proveer al sistema de los mecanismos necesarios para interconectar entre sí los componentes y poder implementar físicamente los circuitos definidos por el usuario en el entorno web, así como para conectar los instrumentos en los puntos apropiados del circuito para su testeo y medida.
- b) Proporcionar una plataforma para la disposición de los componentes electrónicos del laboratorio (condensadores, resistencias, etc.)

El elemento principal de este sistema y que marca una gran diferencia frente a la solución implementada en el proyecto VISIR es la matriz de conmutación utilizada. Pero, ¿por qué utilizamos una matriz de conmutación en el sistema? Para dar respuesta a esta pregunta, tenemos primero que entender cómo se construye un circuito electrónico formado por componentes discretos, como condensadores, resistencias, transistores, etc.

Un circuito eléctrico está formado por una serie de componentes interconectados entre sí y al menos una toma de tierra común para todos. A modo de esquema representativo podría usarse la Figura 3.19, en la que el circuito está compuesto por 5 nodos y 10 ramas. Cada una de estas ramas representa un componente con 2 conexiones o terminales. Es obvio entender cómo a más componentes, el número de posibles nodos aumenta y por tanto la complejidad del circuito. De esta manera, si un circuito está formado por N nodos y el usuario desea añadir un componente más, por ejemplo una resistencia, dispone de $N(N-1)/2$ ramas en las que puede colocarla. Esto sugiere por tanto que si se dispone de 16 nodos, existen 120 posibilidades diferentes donde colocar dicha resistencia.

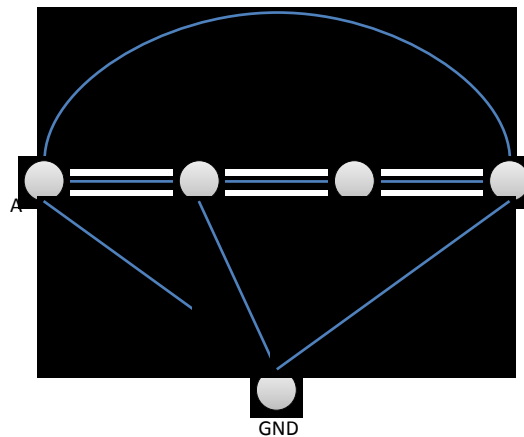


Figura 3.19: Representación de un esquema eléctrico

Respondiendo a la pregunta anteriormente planteada, la función principal de la matriz de conmutación en el sistema de control de experimentos es proporcionar estos nodos o puntos de conexión que son necesarios crear para interconectar los diferentes componentes entre sí, ya que una matriz de conmutación está formada por relés que se abren o cierran en función de las órdenes gestionadas por el mainframe que la contiene.

Ciertamente, el número de conexiones posibles en un circuito de 16 nodos es realmente alto, aunque durante una sesión práctica de una asignatura relativa a la electrónica analógica, el estudiante únicamente debe construir circuitos sencillos en los que no intervienen un excesivo número de componentes, por lo que no se necesita que la matriz de conmutación empleada pueda implementar circuitos con un número elevado de nodos.

En el escenario planteado para comprobar la solución propuesta se han incluido en el mainframe dos módulos 34932A, cada uno de los cuales disponen de dos matrices independientes de 4×16 , es decir de 4 filas y 16 columnas. Para aumentar la capacidad de las matrices de conmutación, se han diseñado unos módulos de expansión para interconectarlas con el objetivo de aumentar la capacidad de la matriz de conmutación, tanto en número de componentes como en número de nodos del circuito.

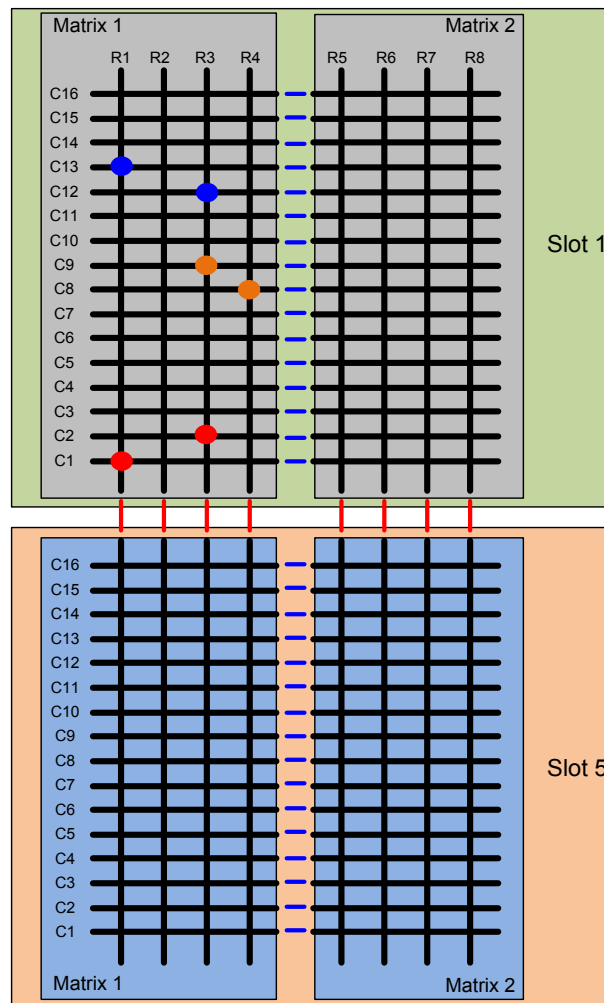


Figura 3.20: Sistema de interconexión de las matrices de conmutación

La matriz resultante se muestra en la Figura 3.20, donde cada uno de los módulos aparece representado respectivamente en verde (módulo conectado en el Slot 1 del mainframe) y en naranja (Slot 5). El mainframe utilizado dispone de 8 slots o ranuras para conectar módulos, por lo que el sistema se podría ampliar con al menos 6 módulos más, cada uno de ellos con sus matrices correspondientes.

En cada uno de los módulos se interconectan las matrices a través de las columnas de cada una de ellas (líneas azules). Se observa en la Figura 3.20 como en cada uno de los módulos la matriz 1 dispone de 16 columnas (de C1 a C16) y 4 filas (de R1 a R4), mientras que en la matriz 2, la numeración de columnas es la misma y las filas van numeradas de R5 a R8.

Al mismo tiempo, se han interconectado cada uno de los módulos a través de sus filas (líneas rojas), de modo que las filas R1 a R4 de la matriz 1 del módulo situado en el slot 1 están conectadas respectivamente a las filas R1 a R4 de la matriz 1 del módulo en el slot 5. Igualmente sucede con las filas R5 a R8 de las matrices 2 de ambos módulos. Como decíamos, el objetivo de estas conexiones es aumentar el tamaño de la matriz y con ello el número posible de nodos e interconexiones a disponer en los posibles circuitos diseñados por el usuario. El resultado obtenido es el mismo que si dispusiéramos una matriz de conmutación de 8 filas y 32 columnas.

Para justificar este aumento de la capacidad de la matriz de conmutación resultante, supongamos que disponemos únicamente de componentes de dos terminales, y que uno de los nodos comunes a todos los circuitos de electrónica analógica básica, es el nodo de tierra o GND. Si consideramos que únicamente tenemos una matriz de 4x16 y que es entre las diferentes columnas donde conectamos los componentes, únicamente podríamos colocar $16/2=8$ componentes en la matriz. De igual manera, si utilizamos las filas utilizadas para identificar las nodos del circuito, únicamente tendríamos 3 nodos genéricos más el de GND, común a todos los circuitos de electrónica.

Es decir, imaginemos que en la Figura 3.20 la matriz 1 no está conectada con el resto y que tenemos un elemento E1 conectado entre las columnas C13 y C12, otro elemento denominado E2 entre las columnas C9 y C8 y un último elemento denominado E3, entre las columnas C2 y C1. La matriz de conmutación funciona de tal modo que conecta las columnas con las filas según los comandos recibidos. Esta conexión se implementa mediante relés electromecánicos que se abren o cierran en función de los comandos recibidos en la matriz o configurados a través del panel frontal. Es decir, en la Figura 3.20 los puntos azules conectan C13 con R1 y C12 con R3, los puntos naranjas C9 con R3 y C8 con R4 y finalmente los puntos rojos C2 con R3 y C1 con R1.

Suponiendo la configuración descrita previamente, el circuito que quedaría construido mediante el cierre de los relés que conectan las columnas con las filas del modo que se ha propuesto anteriormente, y considerando los elementos conectados en los pares de columnas C13-C12 (E1), C9-C8 (E2) y C2-C1 (E3), sería el mostrado en la Figura 3.21, en donde el

elemento E1 estaría en paralelo al elemento E2 y la combinación de ambos en serie con el elemento E3.

Vemos claramente en esta representación como las filas R1, R3 y R4 representan los nodos del circuito, es decir los puntos de interconexión de los componentes, de modo que cuantas más filas tengamos, podrán realizarse con más nodos y por tanto aumentar el número de ramas del circuito de modo que si, por simplificar, consideramos componentes de dos terminales, dispondremos de $N(N-1)/2$ posibles ramas.

Si solamente se dispusiera de una matriz de 4×16 únicamente se dispondría de 4 nodos, lo que da lugar a 6 ramas. Si se considera el esquema de conexión propuesto en la Figura 3.20, se obtienen 8 posibles nodos (de R1 a R8), lo que da lugar a 28 posibles ramas en el circuito aumentando así la complejidad y dimensión de los circuitos a construir mediante la matriz de conmutación, ya que disponemos para ello de más nodos. Además, mediante este esquema de conexión podemos disponer teóricamente de hasta $32/2=16$ componentes de dos terminales, el doble de los 8 que únicamente dispondríamos sobre una matriz 4×16 . Por consiguiente, añadiendo más matrices en los slots del mainframe se dispondrá de más componentes en los experimentos.

Este número de componentes, aunque es válido para justificar por qué se ha adoptado este modelo de conexión, es teórico ya que: a) hemos supuesto solamente componentes de dos terminales y obviamente el sistema debe soportar componentes con más terminales, b) y más importante, porque no se ha considerado que los instrumentos de test y medida del laboratorio, también se deben conectar al circuito, tanto para alimentarlo o excitarlo con determinadas señales, como para obtener medidas de su funcionamiento.

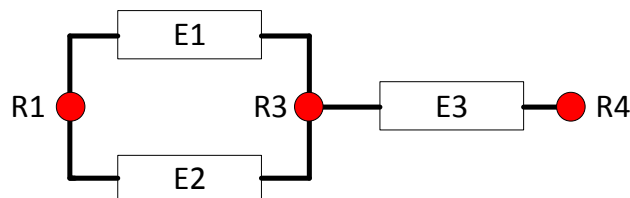


Figura 3.21: Ejemplo de creación de un circuito simple

Así pues, se deben conectar a la matriz de conmutación 8x32 las sondas y puntas de test de los instrumentos del laboratorio. En la Figura 3.8 se muestra la configuración de la matriz resultante empleada durante la fase de pruebas, para la cual se han realizado las siguientes consideraciones:

- Las filas de la matriz resultante (R1 a R8) son los posibles nodos del circuito, y los referenciamos de A a H en orden ascendente, siendo el último nodo H (R8), el común a todos los circuitos y que se encuentra conectado a tierra (GND)
- Todos los instrumentos pueden ser a priori conectados en cualquier nodo del circuito. Aún así el instructor o profesor del laboratorio puede incluir restricciones de montaje, definidas en los ficheros .max
- Todos los instrumentos van a estar siempre conectados en la matriz 1 del módulo conectado en el slot 1 del mainframe de la matriz de conmutación. Esta configuración no se podrá alterar para el correcto funcionamiento del sistema
- Todas las medidas de los instrumentos están referenciadas a un terminal de tierra común que será la columna número nueve (C9), de la matriz 1 situada en el slot 1 del mainframe, en donde están conectadas físicamente las sondas de tierra de los instrumentos.
- El sistema está preparado para albergar componentes de hasta 8 conectores, aunque está diseñado de tal modo que añadir componentes de más conectores sea sencillo y rápido. Se ha elegido este número de conectores ya que era el componente de más tamaño usado actualmente en las prácticas de electrónica analógica básica.

Mediante esta configuración las conexiones de los instrumentos con la matriz de conmutación quedarían de la siguiente manera, estando todas sus puntas de test conectados en la matriz 1 del slot1:

- El generador de funciones está conectado físicamente en la Columna 16 y su punta de tierra a la Columna 9.
- El canal 1 del osciloscopio está conectado a la Columna 15, el canal 2 a la Columna 14. La punta de tierra del osciloscopio, común a ambos canales, está conectada en la Columna 9.
- La sonda del DMM conectada a su canal HI está conectada en la Columna 13, el canal LOW está conectado a la Columna 12 y el canal para medida de intensidades I está conectado a la Columna 11.

- El terminal positivo de la fuente de alimentación está conectado en la Columna 10 y su terminal negativo o tierra, en la Columna 9.

El resto de columnas disponibles en la matriz de 8x32 son empleadas para colocar los componentes disponibles en el laboratorio, de manera que en la configuración actual se pueden tener hasta 12 componentes de dos terminales, tal y como se muestra en la Figura 3.8.

Tal y como describíamos en una sección anterior y volveremos a probar en el capítulo siguiente, la configuración de la matriz de conmutación que hemos realizado permite mejorar el rendimiento de la misma respecto al número de componentes a utilizar para realizar diferentes circuitos, así como a la hora de realizar medidas de intensidad, ya que esa solución no necesita aportar cortocircuitos como si se trataran de componentes de dos pines, tal y como sucede en la plataforma VISIR.

Mediante este modelo de control de experimentos basado en el uso de una matriz de conmutación comercial configurada según los pasos descritos anteriormente, se proporcionan al usuario las herramientas necesarias para crear físicamente los circuitos que diseña en el cliente web, y para conectar instrumentos al circuito con objeto de alimentarlo, excitarlo con señales y comprobar su funcionamiento en condiciones reales.

3.3.1 Ejemplo de configuración y realización remota de un circuito bajo prueba

En esta sección van a ser descritos mediante un ejemplo práctico los pasos que se deben seguir para, partiendo del esquemático de un circuito electrónico, colocar los componentes en la matriz de conmutación, cómo describir los ficheros de configuración necesarios y finalmente explicaremos cómo funciona el algoritmo de control diseñado que actúa sobre el mainframe que contiene la matriz de conmutación y que se encuentra conectado al Servidor de Equipos y Experimentos.

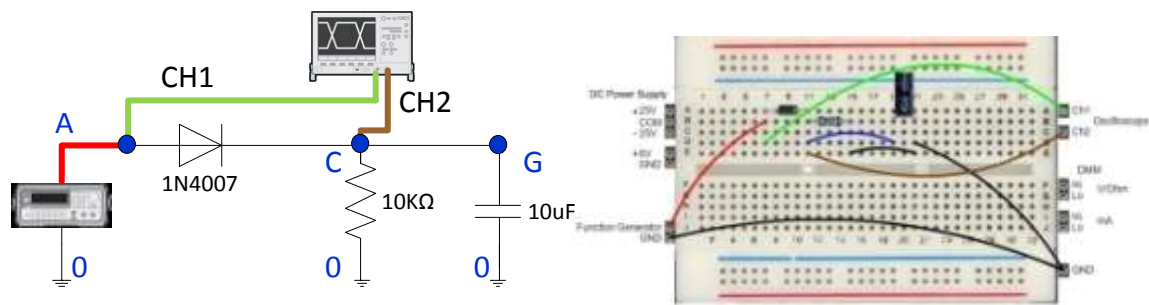


Figura 3.22: Ejemplo de circuito: rectificador de media onda

3.3.1.1 Configuración de los ficheros del Servidor de Equipos y Experimentos

Partiendo de la Figura 3.22 en la que se muestra un rectificador de media onda compuesto por un diodo 1N4007, una resistencia de $10\text{k}\Omega$ y un condensador de $10\mu\text{F}$ que hará las funciones de filtrado de la señal a la salida. Para comprobar el funcionamiento del circuito vamos a excitarlo con una señal sinusoidal de 100Hz y 10Vpp que introduciremos al circuito mediante el generador de funciones y mediante los dos canales del osciloscopio. Chequearemos su comportamiento conectado el canal 1 a la entrada y el canal 2 a la salida. A la derecha del esquema electrónico vemos como sería la implementación de dicho circuito en el cliente web del laboratorio.

El primer paso es identificar los nodos del circuito. En el ejemplo se han denominado A, C y G, además del nodo común de tierra denotado como 0. A continuación debemos configurar el fichero `.max` que utilizará en *Circuit Solver* del Servidor de Medidas para comprobar la validez del circuito enviado desde el cliente web.

Para configurar este fichero `.max`, únicamente debemos transcribir el esquemático obtenido en la Figura 3.22, dando como resultado en este caso el siguiente fichero:

```
VFGENA_FGENA1 A 0
SHORTCUT_S1 G C
D_D1 C A 1N4007
C_C4 G 0 10u
R_R7 C 0 1K
```

Es decir, tal y como ya habíamos descrito, este fichero `.max` sirve para que el instructor decida cómo se pueden interconectar los componentes que hay

dispuestos en el laboratorio. Puede que haya otras posibles combinaciones, como en este caso donde el diodo está polarizado en inversa, pero en este ejemplo el instructor ha decidido que únicamente se pueda excitar el diodo a través de su ánodo, por eso la secuencia de configuración del diodo es D_D1 C A, ya que el primer nodo (C), indica dónde está conectado el cátodo del diodo. En caso de querer dar la posibilidad de practicar con esta otra opción, deberíamos incluir otra línea en el fichero de configuración .max con una sentencia como la siguiente:

```
D_D2 A C 1N4007
```

Esto no implica que debamos incluir otro diodo más en la matriz, tal y como sucedería en el VISIR, ya que un componente solo puede estar entre dos nodos, tal y como explicábamos en la sección anterior. En el método propuesto un componente puede estar conectado entre todos los nodos del circuito. Las restricciones que definen en qué nodos podrá estar conectado el componente en un circuito vendrán determinadas en el fichero .max

Además vemos en el ejemplo dos nodos llamados C y G que eléctricamente están al mismo potencial por lo que son realmente el mismo nodo. Lo hemos introducido en el ejemplo para ilustrar como se configuraría un experimento en el cual queremos realizar una medida de intensidad. En este caso, se estaría haciendo posible la medida de la intensidad que circula por el condensador de 10uF. Para ello, únicamente se define en el fichero .max esta rama del circuito (SHORTCUT_S1) sin tener que hacer ninguna modificación en la configuración de los componentes de la matriz o introducir resistencias de 0Ω como hay que realizar en la matriz de conmutación del proyecto VISIR.

Una vez configurado el fichero .max, ¿cómo determinamos qué componentes tenemos en la matriz y cómo están conectados? Para ello disponemos del fichero *Components.list*, del cual ya hablábamos brevemente en secciones anteriores. Este fichero es el utilizado por el Servidor de Equipos y Experimentos, y en concreto el algoritmo *Circuit Builder*, para obtener a su salida las instrucciones o comandos a enviar a la matriz de conmutación, y realizar las conexiones físicas del circuito, es decir, cerrar los relés adecuados para interconectar los componentes tal y como ha realizado el alumno en el cliente web.

A continuación podemos ver un ejemplo del fichero *Components.list*, el cual describe exactamente cómo están conectados los componentes en la matriz de conmutación mostrada en la Figura 3.8:

```
R_1_1 8      7      10k
R_1_1 6      5      10k
D_1_1 4      3      1N4007
R_1_1 2      1      100k
R_5_1 16     15     470
C_5_1 14     13     10u
C_5_1 12     11     1u
C_5_1 10     9      100n
D_5_1 7      8      Z5.1V
C_5_1 6      5      1n
C_5_1 4      3      1n
R_5_1 2      1      1k
%--Ejemplos de incorporación de un componente de 3 terminales y uno de 8--%
Q_5_1 16     15     14     2n2222
OP_5_1 NC    2      3      4      NC    6      7      NC    UA741
```

En este fichero describimos qué componentes tenemos y dónde están conectados en la matriz de conmutación. Para ello, utilizamos la siguiente sintaxis:

TipoComponente_Slot_Matriz	Columna1	Columna2	Valor	Componente
----------------------------	----------	----------	-------	------------

De este modo, con la siguiente descripción - *R_5_1 16 15 470* - se está indicando que en la matriz de conmutación hay disponible una resistencia en la Matriz 1 del módulo conectado en el Slot 5 del mainframe, entre las columnas 16 y 15, teniendo esta resistencia un valor de 470Ω .

En el caso de tener componentes con una polaridad concreta, como es un diodo o un condensador electrolítico, hemos determinado que el terminal considerado como negativo debe ir conectado en la Columna1.

En las dos últimas filas del ejemplo de fichero propuesto se puede observar cómo sería la configuración para componentes de 3 terminales (un transistor) y de 8 (un amplificador operacional). Vemos que la sintaxis es la misma, solo que añadiendo más columnas hasta cubrir la totalidad de terminales del componente. En caso de que algún terminal del componente no esté conectado en el circuito utilizaremos el identificador "NC".

De esta sencilla manera quedarían definidos qué componentes tenemos en la matriz y dónde están situados en ella. Si se añaden más módulos en los restantes slots del mainframe, únicamente hay que añadir

más líneas al fichero *Components.list*, describiendo mediante la sintaxis definida los nuevos componentes.

3.3.1.2 Configuración del *Circuit Solver* o Instructor virtual

Durante la descripción anterior del funcionamiento y configuración del Servidor de Medidas, se ha hecho referencia en varias ocasiones al denominado instructor virtual el cual no es más que el nombre otorgado al proceso que se ejecuta en el algoritmo al que nos referíamos como *Circuit Solver*.

El instructor virtual es el encargado de determinar si es posible construir físicamente con los componentes e instrumentos de que dispone el laboratorio el circuito que el alumno ha creado en el interfaz web.

La primera cuestión que se plantea el instructor virtual es la misma que la que sucede en el laboratorio real: ¿se dispone de todos los componentes necesarios para crear el circuito? Pero en el caso de un laboratorio remoto se plantea otra cuestión un tanto diferente a la que se podría plantear en el laboratorio tradicional: ¿se pueden interconectar los componentes e instrumentos de todas las maneras posibles? Es obvio que si nos encontramos físicamente en un laboratorio, a priori, los componentes pueden ser interconectados como queramos, realizando todas las configuraciones que consideremos oportunas y conectando los instrumentos allí donde pensemos que es necesario obtener alguna medida o bien introducir una señal en el circuito.

Pero decimos a priori porque generalmente aunque seamos libres para llevar a cabo tantas modificaciones como queramos, en el laboratorio es el profesor o instructor del laboratorio quien determina qué circuito debemos realizar en cada momento, con qué componentes, en qué configuración y de qué instrumentos disponemos para realizar los ejercicios de test y medida planteados durante la clase práctica. Pues bien, estas limitaciones también pueden existir en el laboratorio remoto mediante el instructor virtual.

Por tanto, en el entorno remoto se va a disponer también de un instructor que va a determinar si el laboratorio se puede crear o no en función de una serie de ficheros de configuración, los cuales estarán definidos por el profesor de la asignatura o el instructor del laboratorio,

quien determina los componentes del ejercicio y qué circuitos se pueden crear con esos componentes, así como los puntos de los circuitos en los que se podrán conectar los instrumentos del laboratorio.

La Figura 3.23 representa gráficamente cómo el instructor virtual, en función del circuito enviado desde el cliente y en base a los ficheros de configuración que determinan cómo están interconectados los componentes y los instrumentos, comprueba la validez del circuito construido por el usuario.

Estos ficheros de configuración son a los que hacíamos referencia anteriormente y denotábamos como ficheros .max. Es decir, los ficheros .max serán aquellos que utilice el instructor del laboratorio para determinar cómo se pueden interconectar los diferentes componentes disponibles en el laboratorio. Una vez definidos estos ficheros .max, el archivo de configuración *maxlists.conf* lista estos ficheros, de modo que el *Circuit Solver* puede saber qué circuitos se pueden implementar en el laboratorio en todo momento.

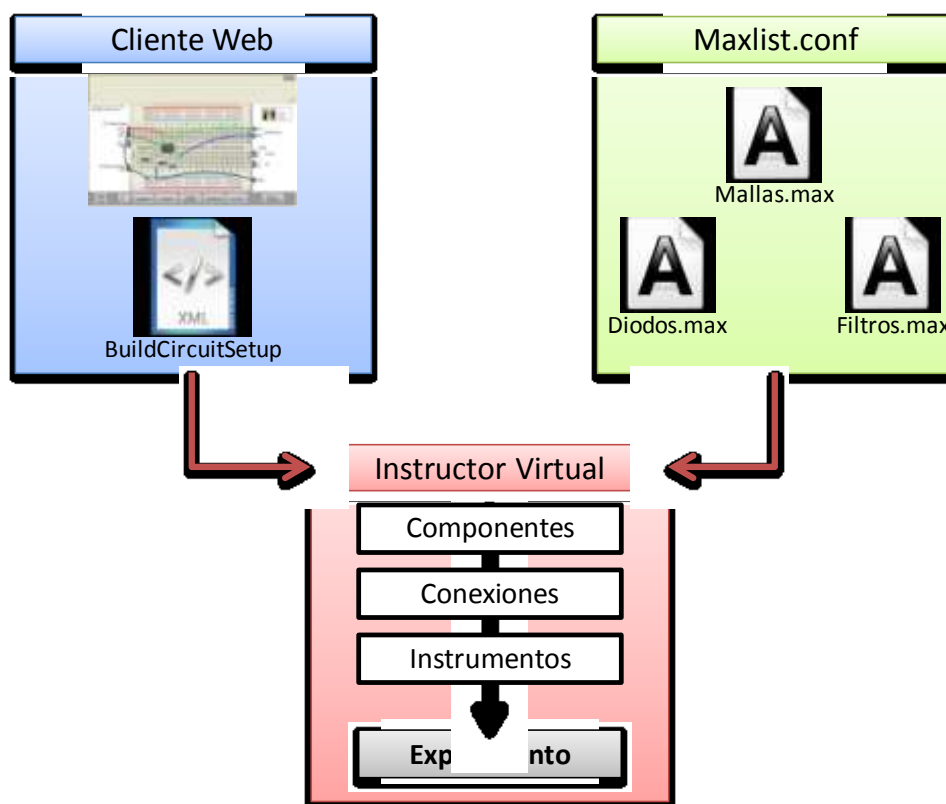


Figura 3.23: Instructor virtual del Servidor de Medidas

A continuación podemos ver un ejemplo de este fichero de configuración en el que se emplean los ficheros `Mallas.max`, `Diodos.max` y `Filtros.max`, los cuales definen una serie de tipos de ejercicios de electrónica. Es decir, puede haber tantos ficheros `.max` como se desee, lo que nos sirve para organizar por prácticas temáticas las sesiones del laboratorio, de modo que actualizando únicamente el fichero `maxlists.conf`, el laboratorio permite realizar unos circuitos u otros.

```
maxlists/Mallas.max  
maxlists/Diodos.max  
maxlists/Filtros.max
```


Pero estos ficheros `.max` deben cumplir siempre una condición: deben siempre contener un subconjunto de los componentes que definiremos en el Servidor de Equipos y Experimentos, pues es en este servidor donde configuraremos qué componentes tenemos en el laboratorio. Es decir, el Servidor de Medidas comprueba mediante la comparación de los datos que le llegan en el fichero XML procedente del cliente con la información proporcionada por ficheros de configuración propuestos en el fichero `maxlists.max`, y determina si es posible o no construir físicamente el circuito con los componentes y conexiones determinadas por el alumno. Si la validación no es positiva, no se enviará ninguna información al Servidor de Equipos y Experimentos y se informará al usuario de que no es posible realizar el circuito mediante un mensaje de error.

Por el contrario, si es posible construir el circuito que el alumno ha generado en el interfaz web el proceso sigue adelante, generando el instructor virtual, a través del *Circuit Solver*, un *netlist* que describe qué componentes e instrumentos se han usado en el circuito y cómo se han realizado las conexiones. Si seguimos con el ejemplo propuesto en la Figura 3.22 y suponemos que el circuito se puede realizar, el *netlist* generado a la salida del *Circuit Solver* será:

```

<circuit>
  <circuitlist>
    W_X FGEN_A A7
    W_X 0 0
    W_X OSC_1 A7
    W_X OSC_2 A11
    W_X 0 A21
    W_X A11 A19
    W_X A15 A21
    R_X A11 A15 1k
    D_X A7 A11 1N4007
    C_X A19 A21 10U
  </circuitlist>
</circuit>

```



```

Solved list is:
VFGENA_FGENA1 A 0
R_R7 C 0 1K
SHORTCUT_S1 G C
D_D1 C A 1N4007
C_C4 G 0 10U
PROBE1_1 A
PROBE2_2 C

```

Se puede observar en esta muestra de código cómo el *Circuit Solver*, aparte de chequear que se puede construir el circuito, también se encarga de realizar dos traducciones:

- Los puntos en los cuales han sido conectados los componentes e instrumentos en la placa de prototipos virtual son transformados en nodos que puedan ser interpretados posteriormente por el Servidor de Equipos y Experimentos para la construcción física del circuito y la conexión de los instrumentos en los puntos de test y medida elegidos por el alumno.
- Se asigna a cada uno de los componentes del circuito un nombre único que sirve para identificarlos entre todos los componentes. Es decir, podremos tener dos resistencias de $1k\Omega$, pero una se denominará R_R1 y otra R_R2, por ejemplo.

Esta asignación de nodos y de nombres a cada una de las conexiones y componentes del circuito se realiza en base a los datos de configuración indicados en alguno de los ficheros .max utilizados en el laboratorio. Así, en este ejemplo tal y como ya decíamos anteriormente, el fichero diodos.max consta, entre otros datos, de los siguientes:

```

VDC+25V_1    F      max:20 imax:0.5
VDCCOM_1     0
VFGENA_FGENA1 A 0
SHORTCUT_S1  G C
D_D1        C A 1N4007
C_C4        H 0 10u
R_R7        C 0 1K

```

Se puede ver en este extracto del fichero de configuración `diodos.max`, cómo además de describir las conexiones de cada componente, se hace referencia a las conexiones tanto del terminal de +25V de la fuente de alimentación como del generador de funciones. Estos son los dos únicos instrumentos de los que se debe especificar dónde deben ser conectarlos en el circuito, ya que son instrumentos generadores de señales y que, en función de dónde se coloquen puede suponer un peligro para el circuito. Por esta razón el profesor o instructor debe indicar en qué puntos del circuito se pueden introducir señales. En cambio, esto no sucede con el osciloscopio o el multímetro ya que son instrumentos únicamente de lectura de datos, por lo que su inclusión en el circuito no es peligrosa y pueden conectarse en cualquier parte del mismo. Únicamente debemos tener en cuenta que para medir intensidades mediante el multímetro necesitamos romper una de las ramas del circuito e introducir en dicha rama el instrumento. Para ello, es necesaria la definición de cortocircuitos (*shortcuts*) en el fichero de configuración `.max`, habilitando de esta manera que se pueda realizar la conexión del multímetro.

A continuación, una vez que el *Circuit Solver* ha comprobado que es posible conectar los componentes y los instrumentos, el módulo `LXICom`, a través de la función `circuit.cpp` a la que hacíamos anteriormente referencia, se encarga de construir el paquete de información que se enviará al Servidor de Equipos y Experimentos y que éste interpretará y utilizará para realizar las conexiones físicas sobre el módulo de conexiones que hemos realizado, a través de la matriz de conmutación. Siguiendo con el ejemplo anterior, la información obtenida a la salida del módulo `LXICom` y que se envía al Servidor de Equipos y Experimentos, sería:

```
BuildCircuitSetup should send:
VFGENA_FGENA1 A 0&R_R2 C 0 10K&SHORTCUT_S1 G C&D_D1 C A 1N4007&C_C4 G 0
10U&PROBE1_1 A&PROBE2_2 C
```

3.3.1.3 Control de la matriz de conmutación

Una vez que el *Circuit Solver* del Servidor de Medidas ha comprobado que el circuito enviado desde el cliente web se puede implementar, le toca al turno al Servidor de Equipos y Experimentos determinar los comandos necesarios para llevar a cabo la construcción física del circuito.

Para ello, el servidor cuenta con el algoritmo denominado *Circuit Builder* definido en el método de control propuesto, por lo que es válido para

ser utilizado con cualquier matriz de conmutación que sea compatible con drivers IVI. Es por tanto, la matriz de conmutación el instrumento encargado de realizar las conexiones oportunas entre componentes para obtener el circuito diseñado por el usuario, así como de conectar los instrumentos al circuito allá donde se requiera. Para realizar estas conexiones es necesario saber qué relés de la matriz de conmutación son los que se tienen que cerrar. Para ello se dispone de:

- a) La información de salida del *Circuit Solver* y denominada *Solved list*, la cual es enviada del Servidor de Medidas al Servidor de Equipos y Experimentos mediante el socket habilitado entre ambos. Con estos datos, el Servidor de Equipos y Experimentos sabe qué componentes e instrumentos está solicitando el usuario, y cómo quiere interconectarlos.
- b) El fichero de configuración *Components.list*, el cual proporciona al Servidor de Equipos y Experimentos de información referente a dónde se encuentran los componentes en la matriz de conmutación con la que estamos trabajando.

Mediante la combinación de una serie de algoritmos que hemos realizado y que están encargados de procesar la descripción procedente del Servidor de Medidas y el fichero *Components.list* se obtienen los comandos que determinan cuáles son los relés que se deben cerrar en la matriz de conmutación. Cada relé en la matriz de conmutación viene representado por un cruce entre una columna y una fila. De tal modo que en la configuración de la matriz propuesta de la Figura 3.8, lo que estamos realizando es la conexión del terminal de un componente conectado en una columna X, con un nodo Y en el circuito correspondiente. Es decir, para saber qué relé se debe cerrar se necesitan dos datos, uno la columna y otro la fila, así como saber también en qué slot y en qué matriz de cada módulo están dichas filas y dichas columnas.

Por esta razón, para activar un relé se necesitan dos comandos enviados sucesivamente a la matriz de conmutación: uno que determina la columna y otro que indica la fila, definiendo la interconexión a realizar. La Figura 3.24 representa una parte de la matriz de conmutación 8x32 definida previamente, y sirve de ejemplo para ilustrar cuáles serían los comandos necesarios para conectar en serie los dos componentes de la figura. Como

cada componente tiene dos terminales, necesitamos cerrar los cuatro relés identificados en la figura. Como ambos componentes tienen un terminal en la misma fila, la conexión entre ambos será en serie.

En este caso el relé 1 se encuentra en la Columna 15, en la matriz 1 del slot 1, por lo que el primer comando a enviar a la matriz será: s1m1c15. El siguiente comando servirá para identificar la fila, que en este caso es la R4, también en la matriz 1 del slot 1, por lo que el comando a enviar será: s1m1r4. Es decir, la sintaxis de los comandos a enviar a la matriz que definen qué relé o conexión hay que cerrar es: sXmYcUU - sXmYrTT, donde X y Y determinan el slot y matriz donde se encuentran la columna UU y la fila TT que definen la conexión a ejecutar.

De manera análoga, el resto de relés de la Figura 3.24 quedarían definidos de la siguiente manera:

- Relé 2: s1m2c14 - s1m2r6
- Relé 3: s5m1c2 - s5m1r4
- Relé 4: s5m2c1 - s4m2r5

Por tanto para definir el relé que cierra una conexión en la matriz de conmutación se necesitan dos comandos, determinados por la relación entre la columna y la fila correspondiente:

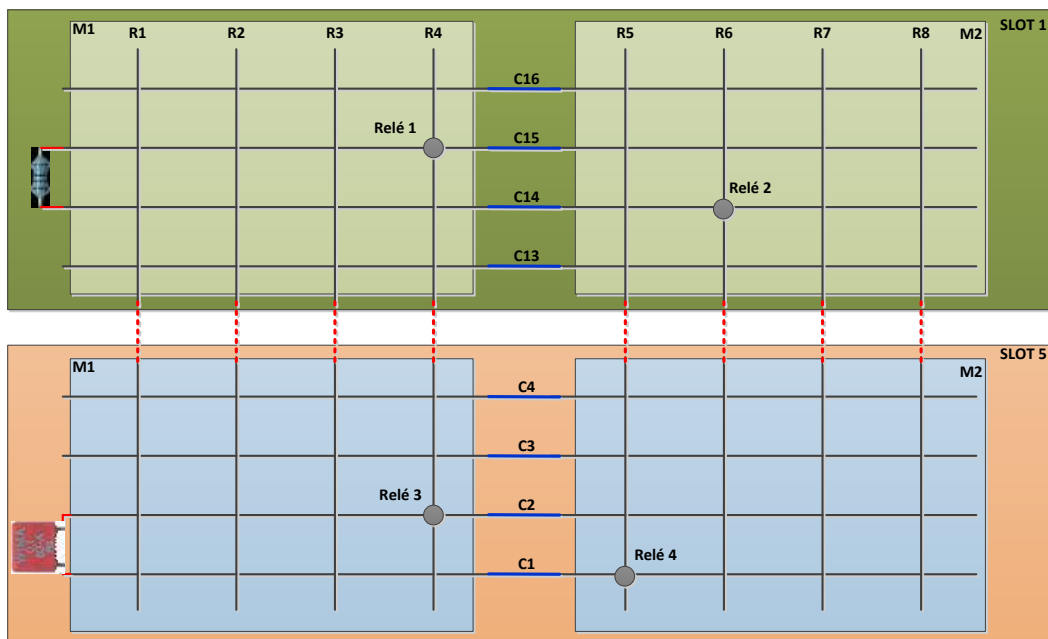


Figura 3.24: Ejemplo de conexión serie de 2 componentes en la matriz de conmutación

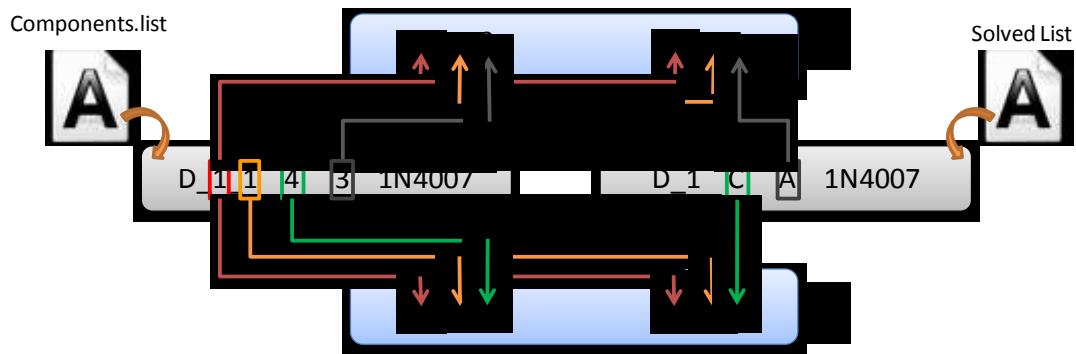


Figura 3.25: Ejemplo de formación de comandos de control de la matriz

1. El primer comando es obtenido del fichero de configuración *Components.list*, ya que si recordamos cual es la sintaxis que define en qué columnas se encuentra cada componente en la matriz, se debe indicar el slot, la matriz y las columnas en las cuales se encuentra conectado.
2. El segundo comando lo obtenemos del *Solved List* enviado desde el Servidor de Medidas, ya que indica los nodos del circuito en los cuales se encuentra conectado el componente, que en nuestra solución determinan las filas en las cuales se deben conectar terminales de cada componente.

De manera gráfica, y volviendo al circuito ejemplo (Figura 3.22) del que partíamos al comienzo de esta sección, los comandos que sirven para conectar el diodo 1N4007 definido en el circuito entre los nodos C y A, y situado en la matriz de conmutación entre las columnas 4 y 3 de la matriz 1 del módulo que se encuentra en el slot 1 del mainframe, serían los mostrados en la Figura 3.25.

Si representamos sobre la matriz de conmutación todos los relés necesarios para crear el circuito ejemplo se obtienen como resultado todos los puntos marcados en gris sobre la Figura 3.8, siendo los comandos necesarios para cerrarlos, los mostrados en la Tabla 3.8.

Instrumentos y Componentes	Comandos			
	Terminal 1		Terminal 2	
Generador de Funciones	s1m1c16	s1m1r1		
Canal 1 Osciloscopio	s1m1c15	s1m1r1		
Canal 2 Osciloscopio	s1m1c14	s1m1r3		
Diodo 1N4007	s1m1c4	s1m1r3	s1m1c3	s1m1r1
Resistencia 1K Ω	s5m1c2	s5m1r3	s5m2c1	s5m2r8
Condensador 10uF	s5m1c14	s5m1r3	s5m2c13	s5m2r8

Tabla 3.8: Comandos para construir un rectificador de media onda y filtro a la salida

De este modo, mediante el método propuesto es posible ampliar la capacidad de la matriz de conmutación de una manera sencilla sin alterar el algoritmo de control pues las órdenes necesarias para crear los circuitos dependen únicamente del fichero de configuración que determina qué componentes hay en la matriz, y dónde están conectados sobre ella.

Finalmente y a modo de resumen, la Figura 3.26 muestra todos los procesos explicados y analizados en secciones anteriores y que se deben realizar para implementar físicamente y de manera remota sobre la matriz de conmutación del laboratorio, el circuito que el usuario ha realizado en el cliente web, tomando como ejemplo el mostrado en la Figura 3.22.

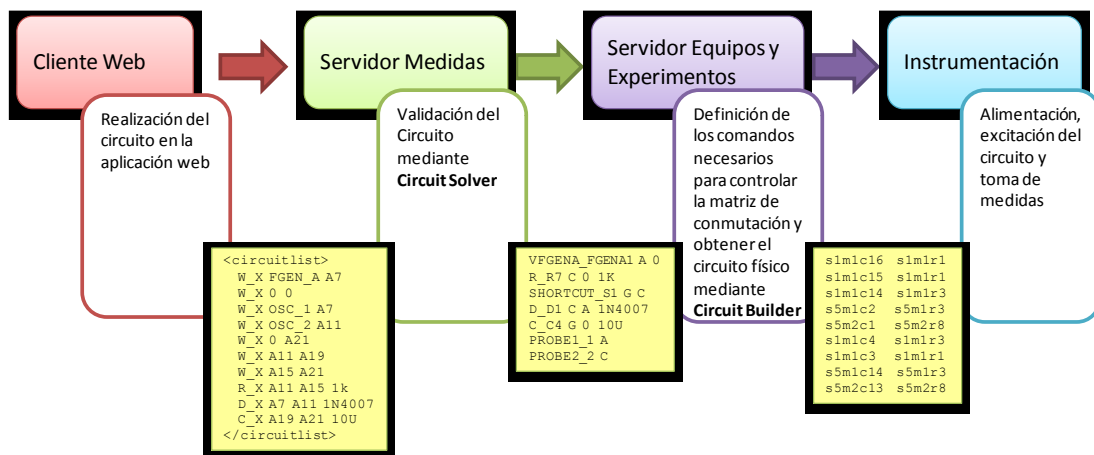


Figura 3.26: Procesos para la construcción física del circuito

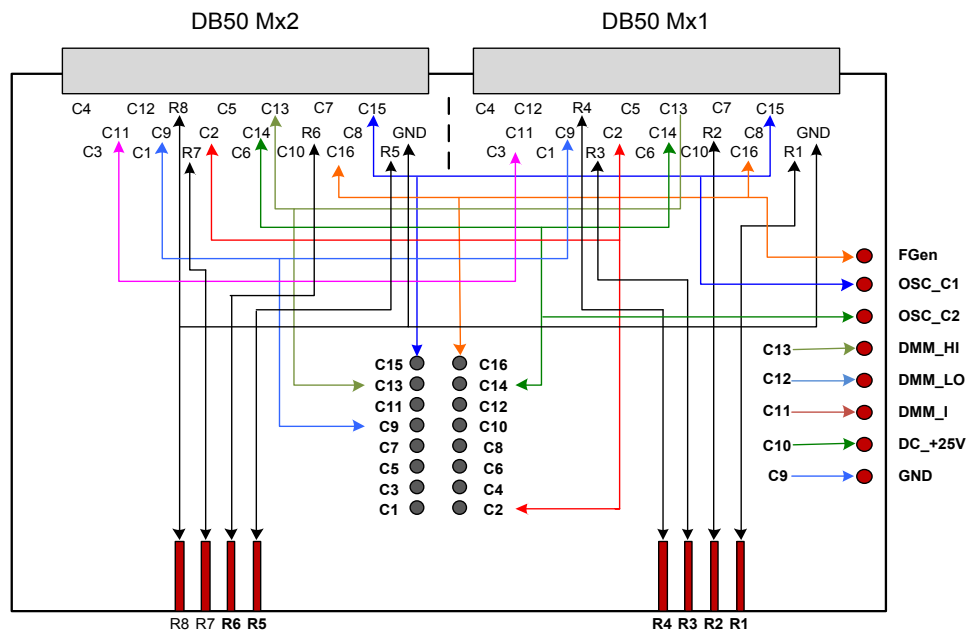


Figura 3.27: Esquema de conexionado de los módulos de expansión

3.3.1.4 Módulos de expansión para componentes

Tal y como hemos explicado, las matrices de conmutación de los módulos contenidos en el mainframe han sido interconectados de manera que obtenemos una matriz de conmutación de 8x32 en la que conectamos los instrumentos y los componentes que deseamos proporcionar a los usuarios del laboratorio para crear circuitos y experimentar con ellos.

Por tanto es necesario disponer de un hardware que permita realizar estas interconexiones físicas entre las matrices de los módulos del modo descrito en la Figura 3.20, así como contener los componentes del laboratorio como mostramos en la Figura 3.8. Para ello hemos diseñado y desarrollado los módulos de expansión cuyo esquema de conexionado aparece representado en la Figura 3.27.

Lo primero a destacar de los módulos de expansión es que no contienen ninguna lógica tal y como se puede apreciar en las fotografías mostradas en la Figura 3.28, por lo que las únicas funciones que desempeñan son:

- Interconectar las columnas de las matrices contenidas en un mismo módulo. En la Figura 2.27 están incluidas todas las conexiones, pero como ejemplo se puede ver en color azul como las columnas C14 están

conectadas entre sí al igual que las columnas C11 o C2. Del mismo modo se ha realizado físicamente la interconexión de las 16 columnas de que disponen cada una de las matrices de los módulos utilizados para validar la solución propuesta.

- b) Interconectar las filas (R1 a R8) de las matrices contenidas en los diferentes módulos conectados en los slots del mainframe. Para ello en la parte inferior del esquema de la Figura 2.27 se han representado dos conectores, uno para cada una de las dos matrices de los módulos utilizados, de manera que con un simple cable se conectan las filas de matrices de los diferentes módulos disponibles en el mainframe de la matriz de conmutación empleada.
- c) Proporcionar una serie de zócalos dispuestos en el centro de los módulos de expansión en donde colocar cada uno de los componentes con los que queremos dotar al laboratorio. Cada uno de los pines de los zócalos está conectado a una determinada columna de la matriz resultante 8×32 , de modo que al conectar los terminales de los componentes en los zócalos, los estamos conectando en la matriz, tal y se indica en la Figura 3.29.
- d) Proporcionar los terminales dispuestos a la derecha de los módulos de expansión (Figura 2.27) y que son necesarios para conectar los instrumentos del laboratorio (generador de funciones, osciloscopio, multímetro y fuente de alimentación) al circuito, a través de la matriz de conmutación 8×32 dispuesta, del modo en que también aparecen representados en la Figura 3.29 c).

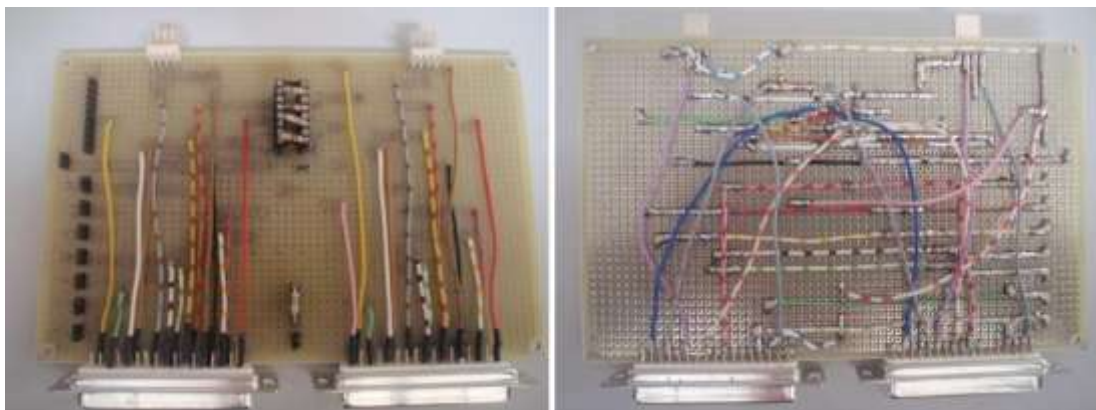


Figura 3.28: Vista frontal y trasera de los módulos de expansión

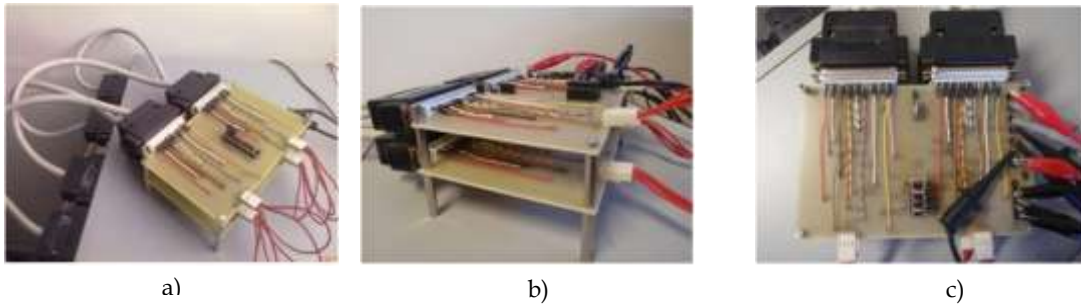


Figura 3.29: Imágenes de los módulos de expansión diseñados y desarrollados

En la combinación de módulos de expansión resultante, únicamente en uno de ellos hay que disponer los terminales para conectar los instrumentos. Este módulo de expansión dispondrá de menos zócalos para conectar componentes que el resto de módulos, ya que como se observa en la Figura 2.27 los instrumentos están conectados a la matriz desde la columna C16 a la C9. En este módulo de expansión dedicado a los instrumentos, esta última columna C9 está conectada a tierra, es decir a la fila R8 y a los terminales de tierra de cada una de las matrices de conmutación disponibles, formando por tanto una referencia de tensión común tanto de los circuitos como de los instrumentos. El resto de módulos de expansión están dedicados íntegramente a albergar los componentes del laboratorio, por lo que disponen para ellos de zócalos de conexionado de la columna C16 a C1.

En la imagen mostrada en la Figura 3.29 a) se pueden ver los dos módulos de expansión desarrollados, los cuales se encuentran conectados a los módulos del mainframe a través de sendos conectores DB50. En la misma imagen, en la parte derecha se observa el conjunto de cables (en rojo) y conectores utilizados para interconectar ambos módulos. En la imagen Figura 3.29 b), se puede ver con una mejor perspectiva ambos módulos de expansión, y de nuevo en la parte derecha aparecen los conectores entre ambos módulos que aparecen representados en la parte inferior del esquema de la Figura 3.27. Por último, a la derecha de la Figura 3.29 c) aparecen conectadas las puntas de prueba de los instrumentos en uno de los módulos de expansión, y en el centro de la fotografía se encuentran los zócalos dispuestos para emplazar los componentes del laboratorio y que se encuentran conectados según se ha definido en el fichero *Components.list* y que podemos ver en la Figura 3.8.

Mediante la implementación de estos sencillos módulos de expansión se pueden reconfigurar fácilmente los experimentos del laboratorio, ya que para añadir o sustituir unos componentes por otros no se tiene más que reemplazar los existentes en los zócalos dispuestos a tal efecto y a continuación actualizar el fichero Components.list. En el caso de añadir un módulo de matrices más en el mainframe, únicamente es necesario añadir un módulo de expansión más y conectarlo a los ya existentes previamente.

De este modo mediante la combinación de la matriz de conmutación comercial y los módulos de expansión construidos, se obtiene un sistema que permite la reconfiguración remota de los circuitos bajo prueba contenidos en los experimentos, y la toma de medidas utilizando los instrumentos reales del laboratorio.

Una vez descrito el sistema de control propuesto, queda por comprobar que se trata de un sistema válido y en el que el hardware adicional necesario para llevar a cabo la experimentación remota no introduce errores significativos en las medidas adquiridas. Para ello, el capítulo cuarto describe todas las pruebas realizadas a tal efecto, las cuales han permitido utilizar el método de control propuesto en un laboratorio remoto con usuarios reales y comparar las medidas obtenidas con las de otro laboratorio remoto de referencia y un laboratorio manual.

*Capítulo***4****Evaluación de la solución**

En el capítulo primero se ha descrito la metodología de investigación seguida en el desarrollo de este trabajo, la cual es del tipo investigación-acción. Inicialmente hemos estudiado las alternativas tecnológicas y soluciones adoptadas por otros grupos de investigación relevantes en el área de la experimentación remota con el objetivo de identificar las limitaciones y problemas que aparecen a la hora de ejercer el control sobre los experimentos incluidos en sus laboratorios, y así justificar la necesidad de una nueva solución para el control de los equipos y circuitos bajo prueba que forman parte de dichos experimentos.

Como resultado de estos estudios y con el objetivo tanto de satisfacer las motivaciones que dieron pie a este trabajo, así como de dar solución a los problemas detectados en áreas concretas de la arquitectura de un laboratorio remoto, se han llevado a cabo una serie de desarrollos que afectan a la aplicación de control, interfaz de comunicaciones y el sistema de control de experimentos que pueden ser empleados en un laboratorio remoto. El conjunto de estas acciones y desarrollos conforman el método de control propuesto.

Siguiendo el método de investigación empleado (Figura 1.10), quedan aún por llevar a cabo las fases de experimentación y validación de la solución propuesta, las cuales son abordadas en este capítulo, donde se describen tanto los escenarios como las pruebas realizadas sobre el método propuesto, con el objetivo de comprobar que se trata de una alternativa válida para ser

desplegada en un laboratorio remoto de electrónica analógica. Pero, para poder llevar a cabo estas pruebas, es necesario disponer del resto de sistemas no contemplados en el alcance de este trabajo, y que son necesarios para el funcionamiento de un laboratorio remoto.

Por esta razón se ha integrado el método propuesto en el laboratorio remoto WebLab-Deusto. De este modo se dispone de un prototipo de laboratorio remoto que será usado como escenario de evaluación de la solución implementada.

Una vez descrito el escenario, analizaremos las pruebas a las que hemos sometido al método de control con el objetivo de comprobar su validez ante los problemas y objetivos planteados al comienzo de este trabajo. Finalizaremos el capítulo con las conclusiones alcanzadas tras analizar de los resultados obtenidos.

4.1 Escenario de evaluación

El WebLab-Deusto es el concepto de laboratorio remoto diseñado y desarrollado por el grupo de investigación homónimo de la Universidad de Deusto. Desde el año 2001 se han implementado diferentes versiones del laboratorio [GOA08] [GLO08] [GLO+06b], todas ellas basadas en el paradigma cliente-servidor, hasta llegar finalmente a una versión estable y de código abierto¹⁴ que actualmente se encuentra disponible en el sitio web del grupo.

Mediante este código es posible ofrecer a los estudiantes el acceso a experimentos reales, principalmente en el área de la electrónica digital. Así los alumnos pueden desarrollar proyectos en sus casas, con todas las ventajas proporcionadas por la experimentación remota, y testear sus programas sobre dispositivos reales a través de los servicios disponibles a tal efecto en la web de WebLab-Deusto. Actualmente, los experimentos sobre los que se puede interactuar a través de esta plataforma son:

- Programación y testeo del dispositivo Xilinx CPLD XC9572 [GLO+06b].

¹⁴ El proyecto de desarrollo WebLab-Deusto se encuentra actualmente bajo Licencia Pública General (GPL) GNU, v2, en el sitio web <http://code.google.com/p/weblabdeusto/>

- Programación y testeo del dispositivo Xilinx Spartan FPGA [GLO+06b].
- Programación y testeo del dispositivo Microchip PIC 18F97J60 [GAH08].
- Experimentos basados en la programación en C++ para el control remotos de instrumentos mediante en el estándar GPIB [GLH+07a]
- Experimentos basados en microbots controlados mediante el dispositivo Microchip 18F2550 PIC, el cual puede ser programado a través de Second Life (SecondLab project) [GIO+10].
- Experimentos de electrónica analógica, ya que se ha integrado el laboratorio VISIR en la plataforma WebLab-Deusto.

Hoy en día WebLab-Deusto ofrece sus laboratorios a los alumnos matriculados en las asignaturas de Electrónica Digital, Estructura de Computadores, Microprocesadores y Lógica Programable, lo que la convierte en una herramienta usada en todos los Grados que actualmente se imparten en Deusto Ingeniería (excepto en el Grado en Ingeniería en Organización Industrial), y que anualmente es accedida por más de 300 alumnos.

4.1.1 Integración del método propuesto en WebLab-Deusto

Como introducíamos al comienzo de este capítulo, el objetivo de esta fase de experimentación es comprobar la validez del método de control propuesto en un escenario real, es decir en un laboratorio remoto con todas funcionalidades necesarias para poder ser utilizado por usuarios reales.

Para ello se han integrado los desarrollos realizados en este trabajo de investigación en la arquitectura del WebLab-Deusto. Se ha tomado esta decisión ya que el WebLab-Deusto dispone de las herramientas necesarias para llevar a cabo el despliegue de un laboratorio remoto, de una manera sencilla y rápida, de modo que facilita y proporciona un banco de pruebas idóneo para testear y comprobar la validez del método de control propuesto. De este modo, el WebLab-Deusto facilita todos aquellos sistemas y utilidades que quedan fuera del alcance de este trabajo de investigación.

Como cliente del laboratorio remoto se utilizará el interfaz de usuario desarrollado en el proyecto VISIR, ya que su potencia gráfica, capacidad de interacción y similitud al escenario propuesto en un laboratorio real son de

muy alta calidad, lo que nos proporciona una excelente aplicación web que actúa como cliente en la arquitectura del laboratorio remoto.

Utilizando como plataforma tecnológica el WebLab-Deusto, se necesita incluir por una parte el cliente web en Adobe Flash y por otra parte los desarrollos realizados y que conforman el método de control propuesto. Gracias al sistema modular empleado en el diseño de la metodología propuesta, en el cual las funciones del servidor son divididas en dos entidades, el Servidor de Medidas y el Servidor de Equipos y Experimentos, esta integración es sencilla y rápida, obteniendo una arquitectura como la mostrada en la Figura 4.1, en la cual se muestra cómo queda embebida la solución propuesta en el dominio definido por el WebLab-Deusto.

4.2 Pruebas de evaluación del método propuesto

El objetivo de esta sección es demostrar de manera cualitativa y cuantitativa que el conjunto de soluciones que hemos desarrollado y forman el método de control propuesto son válidas ante los problemas y objetivos marcados al comienzo de este trabajo. Para ello se han diseñado un conjunto de pruebas y experimentos, cada uno de los cuales está destinado a comprobar que los objetivos planteados han sido alcanzados, así como para validar las soluciones de los problemas descritos en el capítulo primero de este documento, tal y como se recoge en la Tabla 4.1.

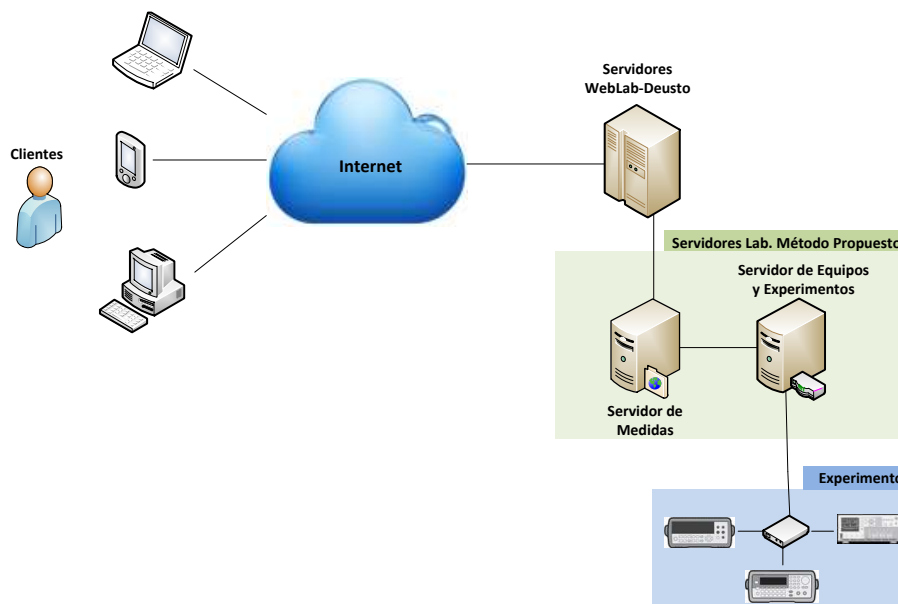


Figura 4.1: Integración del método de control propuesto en WebLab-Deusto

Prueba realizada	Objetivo específico a verificar	Objetivo operativo a verificar	Problema a solucionar
Operación con instrumentos intercambiables	OE1	OO1	P1 – P2 – P3 – P9
Operación con usuarios reales y diferentes tipos de experimentos	OE1 – OE4	OO2 – OO6	P5 – P10
Tiempos de respuesta del método de control	OE2	OO7	P7
Comparativa de medidas respecto a escenario manual y laboratorio VISIR	OE3	OO4 – OO7	P11
Caracterización del sistema de control de experimentos	OE3	OO4	P4 – P6 – P11
Rendimiento del sistema de control respecto al sistema empleado en el laboratorio VISIR	OE2	OO3 – OO5 – OO7	P8 – P12 – P13

Tabla 4.1: Conjunto de pruebas y objetivos y soluciones a verificar

Como decíamos en el capítulo segundo no hemos podido cotejar cuantitativamente todos los laboratorios analizados al carecer de acceso a los mismos. En cambio, gracias a que disponemos de una plataforma VISIR en nuestros laboratorios, hemos podido realizar una serie de pruebas que han servido para comparar la solución de control propuesta con la empleada en este laboratorio y con un sistema totalmente manual. Empleamos la plataforma VISIR como referencia, ya que actualmente es utilizada y validada por docentes y alumnos como herramienta docente [TSM⁺11] [CAM⁺11] [GGH⁺10]. Aún así, y tal y como veíamos en el capítulo anterior, introduce un error en la medida de tensión de señales periódicas que, como veremos en las pruebas realizadas, hemos solucionado mediante el método de medida planteado.

Junto a estas pruebas destinadas a verificar cuantitativamente la repetitividad, error y velocidad de respuesta del sistema de medida y control desarrollado, hemos realizado otra serie de pruebas que reportarán datos cualitativos del sistema, ya que no se pueden proporcionar datos numéricos con los experimentos realizados. En concreto nos estamos refiriendo a aquellas que tienen como objetivo verificar que el método de control puede ser empleado con instrumentos de diferentes fabricantes y aquellas mediante

las cuáles mostraremos cómo añadir nuevos componentes y experimentos al laboratorio es más sencillo que el laboratorio VISIR. En cada uno de los test que ejecutaremos durante esta fase del trabajo de investigación se plantearán en primer lugar sus objetivos, seguidamente se describe el escenario de la prueba, finalizando con los resultados obtenidos en cada una de ellas.

4.2.1 Operación con instrumentos intercambiables

El objetivo principal de esta prueba es validar que el método de control propuesto puede ser utilizado con instrumentos de diferentes fabricantes y verificar por tanto que es completamente independiente de los equipos empleados, siempre y cuando cumplan un única característica: que puedan ser controlados mediante drivers IVI.

Adicionalmente, si se logra este objetivo estaremos asegurando que el laboratorio es fácilmente escalable respecto a la posibilidad de sustituir unos instrumentos por otros sin tener que hacer ninguna actualización en la aplicación de control. Para ello, y tal y como vimos en el capítulo anterior, se ha desarrollado un software de control que emplea las funcionalidades proporcionadas por los drivers IVI para el control remoto de los equipos y experimentos del laboratorio.

Para llevar a cabo esta prueba hemos dotado al método de control propuesto, integrado en el WebLab-Deusto, con los siguientes instrumentos, los cuales también serán empleados en el resto de pruebas descritas en este capítulo:

- Generador de funciones 33220A
- Multímetro digital 34410A
- Osciloscopio DSO5012A
- Fuente de alimentación N5746A
- Matriz de conmutación formada por el mainframe 34980A y dos módulos 34932A.



Figura 4.2: Equipos utilizados en las pruebas del método de control propuesto

Todos estos instrumentos de Agilent Technologies empleados para testear la el método propuesto (Figura 4.2) tienen en común que disponen del interfaz de control LXI-C, lo que implica que pueden ser gestionados remotamente a través de una red Ethernet y que el fabricante proporciona drivers IVI para que la aplicación de control pueda hacer uso de sus funciones para su control a distancia.

Si bien todos los algoritmos de control desarrollados para cada uno de estos instrumentos son independientes del fabricante, antes de realizar cualquier codificación hemos analizado y estudiado cada una de las funciones proporcionadas por los drivers IVI, de modo que la aplicación de control puede utilizarlas para gestionar los instrumentos en base a los comandos enviados desde el cliente web proporcionado por el laboratorio VISIR. El objetivo de esta acción es no tener que realizar ninguna modificación en este interfaz web y al mismo tiempo poder interpretar y ejecutar estos comandos mediante la aplicación de control con independencia del fabricante de los equipos.

Las pruebas realizadas en el laboratorio utilizando los instrumentos enumerados anteriormente han sido totalmente satisfactorias. Como justificación, podemos decir que todas las pruebas que a continuación vamos a realizar en esta fase de validación están realizadas con este juego de instrumentos, por lo que se puede dar como válida la arquitectura software

desarrollada para controlar los equipos del laboratorio, que, tal y como describíamos en el capítulo anterior, son todos ellos comerciales.

Para validar que esta misma aplicación puede funcionar correctamente con otros instrumentos que no pertenezcan a este fabricante, únicamente debemos tener en cuenta a la hora de seleccionarlos que puedan ser controlados mediante drivers IVI. Gracias al diseño modular de la metodología seguida y al funcionamiento y configuración de la aplicación de control, el interfaz de comunicaciones empleado no tiene por qué ser obligatoriamente LXI, ya que la aplicación software es totalmente independiente de la tecnología de comunicación empleada. Al tratarse de un laboratorio de electrónica analógica en el que no se trabajará con señales de alta frecuencia, no es necesario atender a ninguna otra especificación especial a la hora de incorporar nuevos instrumentos.

Para las pruebas con instrumentos diferentes a los que disponemos en la configuración original hemos trabajado a nivel de simulación, ya que no disponemos de otros equipos físicos para estas pruebas. Aún así, estas pruebas son totalmente fiables ya que una de las funciones proporcionadas por los drivers IVI es la de simulación. Para ello, hemos llevado a cabo la configuración de los siguientes instrumentos:

- Generador de funciones DG5351 de Rigol Technologies
- Multímetro digital DM3058 de Rigol Technologies
- Osciloscopio DPO7104C de Tektronix
- Fuente de alimentación PS2520G de Tektronix
- Matriz de conmutación 1260-40 de Racal Instruments

Una vez descargados los drivers de cada uno de estos instrumentos y realizada su configuración tal y como explicamos en el apéndice A, podemos ejecutar la aplicación de control y el resto de aplicaciones del laboratorio remoto tal y como si estuviéramos trabajando con equipos reales. La diferencia radica en que ahora la *Driver Session* de cada uno de los instrumentos que estamos utilizando se está ejecutando en modo de simulación.

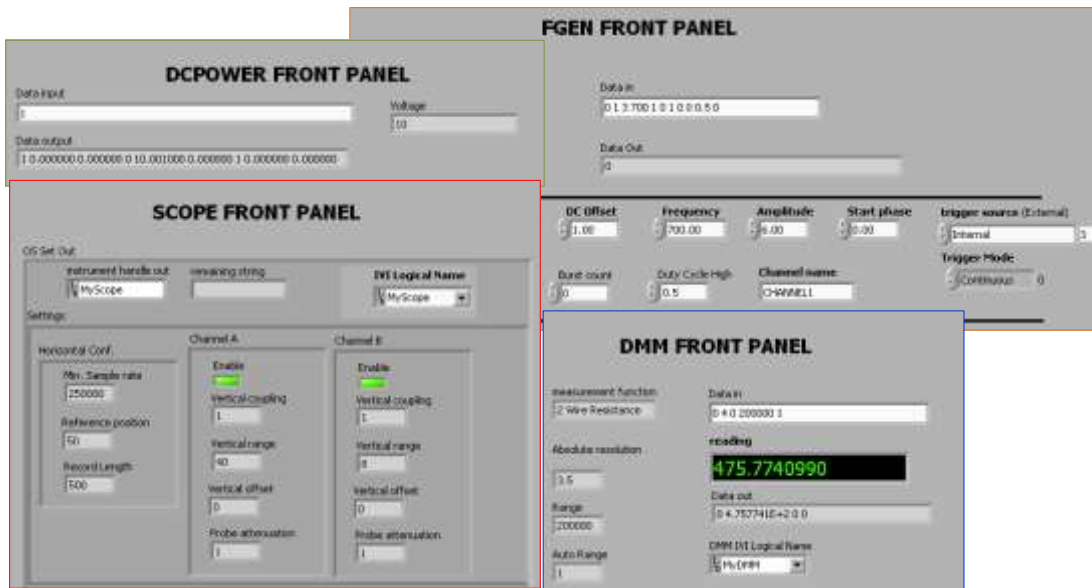


Figura 4.3: Paneles frontales de instrumentos del laboratorio

Obviamente, en este experimento no se han obtenido medidas reales de circuitos bajo prueba reales, ya que no disponemos de los instrumentos físicos para ello, pero sí que se puede observar cómo los comandos y configuraciones enviadas desde el interfaz web se ejecutan sobre los paneles virtuales de cada uno de los algoritmos de control incluidos en la aplicación del Servidor de Equipos y Experimentos, tal y como se muestra en la Figura 4.3.

De este modo, observando la respuesta de los instrumentos a estos datos de configuración, se puede ver cómo, sin realizar ninguna modificación en los algoritmos y únicamente mediante la instalación y configuración de los drivers IVI de cada uno de los equipos utilizados, la aplicación de control diseñada y desarrollada en el método de control sirve para controlar remotamente equipos de diferente fabricante.

Por tanto, a tenor de las pruebas y resultados obtenidos al sustituir los instrumentos originales por otros, que aunque simulados permiten analizar el comportamiento de los mismos ante las órdenes del usuario remoto, podemos decir que la aplicación de control desarrollada permite que el método de control obtenido sea independiente del proveedor de los instrumentos, haciendo posible que la tarea de sustitución de unos equipos por otros sea sencilla y rápida y no conlleve la recodificación de ninguno de los algoritmos de control desarrollados. Además se trata de una solución

desplegada haciendo uso exclusivo de equipos comerciales, lo que confirma, tal y como veremos más adelante, que se puede obtener un laboratorio remoto de electrónica analógica basado únicamente en equipos comerciales, sin la necesidad de diseñar o desarrollar soluciones propietarias.

4.2.2 Operación con usuarios y experimentos reales

Esta prueba tiene como objetivo comprobar que la metodología diseñada puede ser desplegada en un laboratorio remoto y ser utilizada por usuarios reales en un escenario totalmente operativo de experimentación remota, de modo que puedan llevar a cabo experimentos independientemente de la complejidad de los circuitos bajo prueba y de los instrumentos empleados.

Para el ello, el usuario tiene que poder construir circuitos electrónicos de manera remota mediante las funcionalidades proporcionadas por el método de control desarrollado, sin que el hecho de emplear una plataforma de control remota introduzca desviaciones o errores en las medidas obtenidas. En el desarrollo de este experimento se ha contado con un total de 15 usuarios de ambos sexos, todos ellos sin experiencia en el uso de laboratorios remotos y alguno de ellos sin conocimientos relativos al diseño y medida de circuitos electrónicos analógicos. El experimento se llevó a cabo en uno de los laboratorios de la Facultad de Ingeniería de la Universidad de Deusto, en el cual cada usuario disponía de un ordenador, así como un nombre de usuario y contraseña para acceder al laboratorio remoto a través de la plataforma WebLab-Deusto.

A cada grupo de cinco alumnos se entregó un modelo diferente de ejercicios prácticos a realizar sobre el laboratorio remoto. Estos ejercicios consisten en la creación y medida de circuitos electrónicos de diferente dificultad, considerando el grado de dificultad en función del número de componentes, nodos y mallas del circuito.

Esta prueba se planteó de tal modo que cada grupo de usuarios tuviera que construir diferentes circuitos y tomar distintas medidas, con el objetivo de obtener información relativa a la capacidad de respuesta del sistema al tener que ejecutar simultáneamente distintas configuraciones de circuitos y adquirir distintas medidas utilizando para ello diferentes instrumentos. La Tabla 4.2 muestra el set de ejercicios y modelos utilizados en el ensayo.

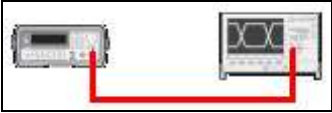


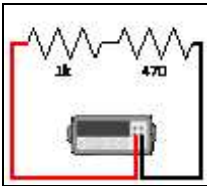
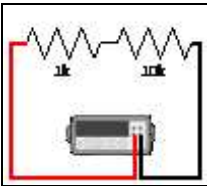
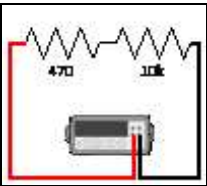
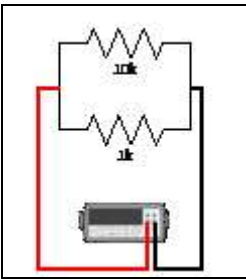
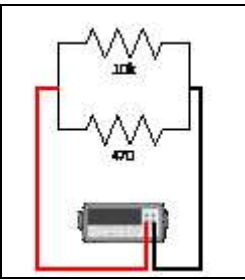
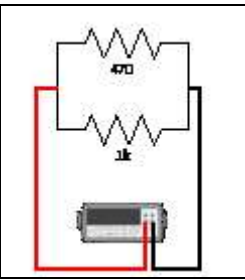
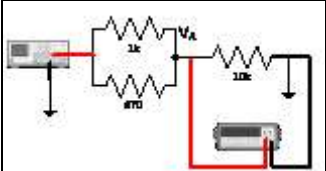
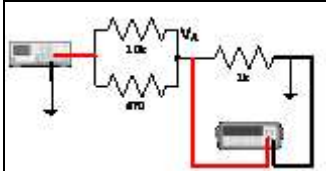
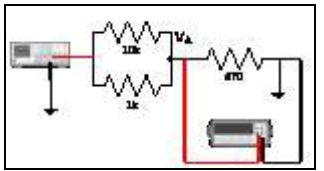
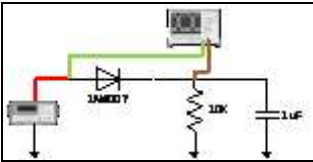
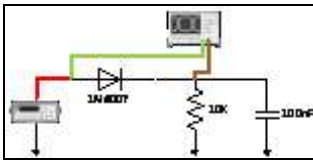
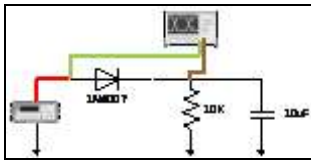
Objetivo Ejercicio	Modelo 1	Modelo 2	Modelo 3
Ejercicio 1. Medidas simples con osciloscopio digital	 Señal senoidal de frecuencia 1kHz y 10Vpp	 Señal diente de sierra de 60kHz y 8Vpp	 Señal cuadrada de 800Hz y 3Vpp
Ejercicio 2. Medidas de resistencias en serie			
Ejercicio 3. Medidas de resistencias en paralelo			
Ejercicio 4. Medidas de tensiones			
Ejercicio 5. Medidas de circuitos en frecuencia	 Señal de entrada triangular de 50Hz y 7Vpp	 Señal de entrada senoidal de 100Hz y 10Vpp	 Señal de entrada triangular de 75Hz y 5Vpp + 1Voffset

Tabla 4.2: Juego de ejercicios utilizados con usuarios reales

Además a la hora de llevar a cabo la prueba, el orden de ejercicios en cada uno de los modelos fue alterado, de manera que cuando los usuarios del modelo 1 estaban realizando el ejercicio 1, los usuarios del modelo 2 estaban realizando el ejercicio 2 y los usuarios del modelo 3 el ejercicio 3.

La razón de desordenar los ejercicios es obtener información sobre la capacidad del sistema para devolver respuestas válidas a las peticiones de

hasta 15 usuarios que están accediendo simultáneamente al laboratorio y que están construyendo físicamente circuitos con diferentes componentes en distintas configuraciones y estimulando y midiendo señales sobre el circuito utilizando diferentes instrumentos configurados también con distintos valores.

Esta prueba así planteada ha servido para obtener resultados sobre la exactitud y dispersión de las medidas realizadas respecto a los valores teóricos que se deberían obtener. En una prueba posterior compararemos estos resultados con los obtenidos utilizando los mismos componentes pero empleando el laboratorio VISIR y un laboratorio manual.

Los resultados de las medidas obtenidas en los diferentes ejercicios y modelos, aparecen reflejadas en las tablas que mostramos a continuación, donde se ha incluido la siguiente información:

- Para cada uno de los cinco usuarios que realizan medidas sobre cada uno de los tres modelos de ejercicios, hemos indicado el valor medio de las diez medidas realizadas consecutivamente por cada uno de estos usuarios. Representamos este valor medio, ya que debemos tener en cuenta que en este ensayo tenemos hasta 15 usuarios accediendo a la vez en el sistema, por lo que para eliminar posibles medidas erróneas, si las hubiera, trabajaremos en base a este valor medio
- A continuación, calcularemos la media aritmética de las medias de los cinco usuarios que realizan cada modelo de ejercicio. De este modo estaremos obteniendo el valor medio de las medidas obtenidas sobre un mismo ejercicio, por los cinco usuarios que están experimentando sobre él.
- A continuación calculamos la desviación típica estándar de las medias obtenidas para cada grupo de 5 usuarios, obteniendo mediante esta fórmula información sobre la dispersión obtenida en base a los valores medios utilizados en los cálculos.
- Este valor indica la desviación del valor medio de las medias obtenidas, respecto al valor teórico que se debería obtener, es decir, error medio cometido. Este valor nos da información sobre la proximidad de los valores medios empleados con respecto al valor

teórico. Un valor cercano a 0 nos indicará que el valor medio de las medidas obtenidas por los cinco usuarios que realizan el mismo circuito, está próximo al valor teórico, siendo así el error cometido prácticamente nulo.

$$Error = 1 - \left(\frac{\bar{U}_T}{Valor\ teórico} \right) * 100 \quad E. 1$$

El primer ejercicio realizado en esta prueba, consiste en la conexión directa del generador de funciones con uno de los canales de entrada del osciloscopio. En la Tabla 4.3 se puede ver como a la hora de efectuar medidas con el osciloscopio sobre la señal capturada, el valor de frecuencia medida es prácticamente igual al introducido por el usuario en el generador de funciones, obteniendo una desviación cercana a 0Hz respecto al valor teórico.

En cambio, en el valor de tensión medido existen pequeñas variaciones, las cuales pueden ser debidas a las pequeñas resistencias introducidas en el circuito por el propio sistema de medida: sondas de medida, así como las resistencias parásitas introducidas por los relés y los módulos de conexión del sistema de control de experimentos.

Aún así, estos errores pueden ser considerados propios del sistema de medida, ya que como veremos en una prueba posterior, estos errores se mantienen incluso cuando se realizan las mismas medidas sobre el circuito montado en una placa de prototipado.

Ejercicio 1		Usuarios					Valor medio U_T	Desv. típica δ_{media}	Error (%)
		U_1	U_2	U_3	U_4	U_5			
Modelo 1	Frecuencia 1kHz	999,00Hz	1,00kHz	998,00Hz	1,00kHz	1,00kHz	999,400Hz	0,800	0,06
	Tensión 10Vpp	20,00V	20,00V	20,00V	20,00V	19,99V	19,998V	0,004	0,01
Modelo 2	Frecuencia 60Hz	60,10Hz	59,80Hz	60,20Hz	60,06Hz	60,20Hz	60,090Hz	0,120	0,16
	Tensión 8Vpp	16,15V	16,25V	16,25V	15,94V	16,09V	16,136Hz	0,110	0,85
Modelo 3	Frecuencia 800Hz	798,00Hz	800,00Hz	800,00Hz	800,00Hz	798,00Hz	799,200Hz	0,970	0,10
	Tensión 3Vpp	6,05V	6,12V	6,10V	6,11V	6,15V	6,108V	0,030	1,80

Tabla 4.3: Resultados obtenidos sobre el ejercicio 1, medidas con osciloscopio

En este ejercicio el osciloscopio se conecta directamente a la salida del generador, por lo que debemos tener en cuenta la impedancia de cada uno de estos terminales para interpretar los resultados. La Z_{out} del generador son 50Ω y la Z_{in} del osciloscopio $1M\Omega$.

Tanto el ejercicio 2 como el ejercicio 3 tienen como objetivo la medida de resistencias en configuración serie y configuración paralelo. Teniendo en cuenta que las resistencias presentan una tolerancia del 10% respecto a su valor teórico, los valores medidos e indicados en la Tabla 4.4, no se encuentran fuera del rango de tolerancia para cada una de las configuraciones de resistencias realizadas, por lo que consideramos los valores obtenidos como válidos en todos los casos.

El ejercicio 4 tiene como objetivo utilizar la fuente de alimentación y comprobar la caída de tensión en uno de los puntos del circuito. A tenor de los resultados mostrados en la Tabla 4.5, la estabilidad del sistema de control para llevar a cabo este tipo de medidas es elevada ya que la desviación típica obtenida en las medidas es prácticamente 0, así como la desviación respecto al valor teórico.

Ejercicio 2	Usuarios					Valor medio U_T	Desv. típica δ_{media}	Error (%)
	U_1	U_2	U_3	U_4	U_5			
Modelo 1 Valor teórico: 1470Ω	1.461Ω	1.463Ω	1.457Ω	1.465Ω	1.464Ω	$1.462,4\Omega$	2,92	0,51
Modelo 2 Valor teórico: $11k\Omega$	10.887Ω	10.868Ω	10.909Ω	10.868Ω	10.885Ω	$10.883,4\Omega$	15,13	1,06
Modelo 3 Valor teórico: $10,470k\Omega$	10.320Ω	10.359Ω	10.396Ω	10.347Ω	10.365Ω	$10.357,4\Omega$	24,72	1,07

Ejercicio 3	Usuarios					Valor medio U_T	Desv. típica δ_{media}	Error (%)
	U_1	U_2	U_3	U_4	U_5			
Modelo 1 Valor teórico: 909Ω	$901,4\Omega$	$904,8\Omega$	$900,4\Omega$	$902,4\Omega$	$899,8\Omega$	$901,80\Omega$	1,74	0,79
Modelo 2 Valor teórico: 448Ω	$446,0\Omega$	$448,5\Omega$	$447,8\Omega$	$446,4\Omega$	$447,1\Omega$	$447,20\Omega$	0,90	0,17
Modelo 3 Valor teórico: 319Ω	$319,5\Omega$	$321,1\Omega$	$318,5\Omega$	$318,3\Omega$	$321,2\Omega$	$319,75\Omega$	1,22	0,23

Tabla 4.4: Resultados obtenidos en ejercicios 2 y 3 sobre medidas de resistencias

Ejercicio 4	Usuarios					Valor medio U_T	Desv. típica δ_{media}	Error (%)
	U_1	U_2	U_3	U_4	U_5			
Modelo 1 Valor teórico: 12,59V	12,59V	12,59V	12,58V	12,59V	12,59V	12,58V	0,0010	0,004
Modelo 2 Valor teórico: 4,83V	4,82V	4,82V	4,82V	4,82V	4,82V	4,82V	0,0001	0,160
Modelo 3 Valor teórico: 4,43V	4,44V	4,44V	4,44V	4,44V	4,44V	4,44V	0,0005	0,280

Tabla 4.5: Resultados obtenidos en el ejercicio 4 sobre medidas de tensiones

En el ejercicio 5 se realizan medidas sobre un rectificador de media onda basado en un diodo y con un filtro paso bajo a la salida para filtrar la señal rectificada. En este ejercicio, hemos representado en la Tabla 4.6 los valores obtenidos al medir la señal de salida del circuito, pues las medidas obtenidas a la entrada nos dan unos datos similares a los recogidos en el ejercicio 1. En los modelos estudiados, aún obteniendo una señal rectificada a la salida, su frecuencia debe ser la misma que la señal de entrada. En cambio su amplitud dependerá del filtrado realizado. En este análisis hemos tomado como valores de referencia para calcular la desviación teórica los valores de tensión obtenidos al medir el mismo circuito sobre una placa de prototipos en un laboratorio manual.

Al igual que sucedía en el ejercicio 1, el valor de la frecuencia de la señal es prácticamente igual al valor de la señal original, presentando en todos los modelos una desviación típica y desviación respecto al valor teórico cercanas a 0. En el caso de las medidas de tensión, las desviaciones típicas siguen una progresión cercana al cero, y una desviación respecto al valor teórico por debajo del 1%.

Debemos tener en cuenta, aparte de las posibles resistencias parásitas inherentes al sistema de control y medida, que en los casos en los que las señales a medir son pequeñas debemos considerar la precisión de los propios equipos empleados. Esta información puede ser consultada en la web del fabricante de los equipos empleados.

Ejercicio 5		Usuarios					Valor medio U_T	Desv. típica δ_{media}	Error (%)
		U_1	U_2	U_3	U_4	U_5			
Modelo 1	Frecuencia 50Hz	50,00Hz	50,00Hz	49,99Hz	50,00Hz	49,99Hz	49,990Hz	0,004	0,008
	Tensión 5,2Vpp	5,13V	5,15V	5,15V	5,23	5,13	5,150V	0,039	0,870
Modelo 2	Frecuencia 100Hz	100,00Hz	100,00Hz	100,00Hz	100,00Hz	99,99Hz	99,998Hz	0,004	0,002
	Tensión 9,87Vpp	9,84V	9,83V	9,84V	20,10V	9,80V	9,820V	0,014	0,440
Modelo 3	Frecuencia 75Hz	74,91Hz	75,00Hz	75,00Hz	75,00Hz	74,91Hz	74,960Hz	0,041	0,053
	Tensión 0,871Vpp	0,875V	0,865V	0,872V	0,875V	0,875V	0,872V	0,003	0,160

Tabla 4.6: Resultados obtenidos en el ejercicio 5 sobre medidas de frecuencia

Si representamos gráficamente el error obtenido en las medidas realizadas en los cinco ejercicios anteriores (Figura 4.4), se puede observar como los valores de tensión en continua y frecuencia son los que menor desviación presentan respecto al valor teórico, mientras que las medidas de resistencias y de tensión alterna (valores pico a pico de señales periódicas), conllevan un mayor error. En todos los casos se observa que el error no supera el 2%.

En resumen, para llevar a cabo estos ensayos hemos definido ejercicios basados en circuitos de diferente complejidad, partiendo del ejercicio más simple (ejercicio 1) consistente únicamente en una conexión directa entre dos equipos, a uno más complejo (ejercicio 5) en el que entran en juego más componentes y por tanto más conexiones a realizar sobre la matriz de conmutación. Hemos realizado estos ensayos con 15 usuarios reales interactuando a la vez sobre el laboratorio, por lo que el método de control debía ser capaz de configurar el circuito de cada usuario y devolverle a cada uno de ellos las medidas obtenidas sobre su circuito sin interferir en los ejercicios de los demás usuarios.

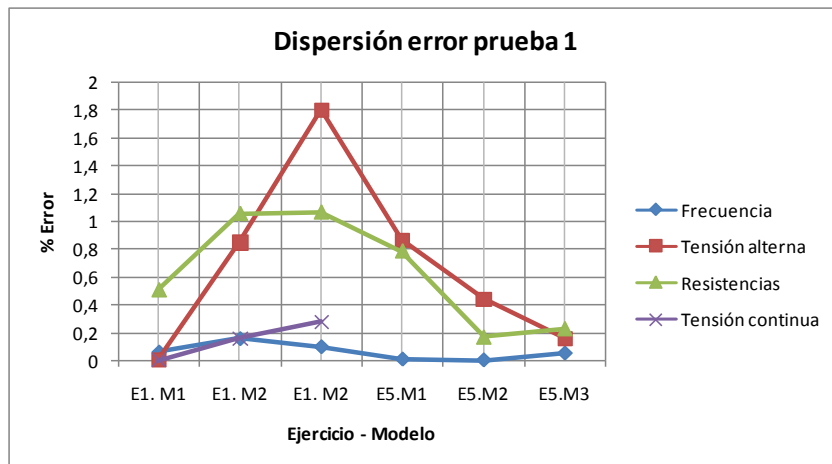


Figura 4.4: Dispersión de error en medidas prueba 1

A tenor de los resultados obtenidos durante los ensayos, vemos que mediante el sistema de control y medida propuesto obtenemos repetitividad en las medidas y valores muy cercanos a los valores teóricos calculados para cada modelo de ejercicio. Podemos determinar por tanto, que el método de control diseñado proporciona un sistema remoto de construcción y medida de circuitos electrónicos independiente de su tipo y complejidad y hábil para ser utilizado con usuarios reales que accedan de manera concurrente al laboratorio. Quedan por tanto de esta manera satisfechos los objetivos marcados, así como validadas las soluciones propuestas a los problemas detectados a este respecto.

4.2.3 Tiempos de respuesta del método de control

El objetivo de este ensayo es estimar el tiempo de respuesta del sistema en un escenario en el cual se producen un número determinado de peticiones simultáneas sobre el mismo. Es decir, queremos obtener información sobre la rapidez con la que el sistema puede devolver datos al usuario, cuando existe acceso concurrente de usuarios en el laboratorio.

Las pruebas que vamos a llevar a cabo están centradas en validar el interfaz de control empleado en el método propuesto. Como recordaremos, el interfaz utilizado para comunicar la aplicación de control con los instrumentos es LXI-C, el cual está basado en Ethernet. Tal y como ya apuntábamos debemos tener en consideración la latencia o tiempo de retardo introducido por este interfaz en el sistema. Los resultados obtenidos nos

servirán para analizar el rendimiento del método propuesto respecto al tiempo de respuesta del sistema, es decir, el tiempo que necesita el sistema para devolver al usuario las medidas realizadas sobre el circuito previamente construido.

Para realizar estos ensayos se ha desarrollado un programa en lenguaje Java que nos permite emular un número determinado de usuarios enviando peticiones al laboratorio. El objetivo de este programa es generar de 1 a 60 peticiones simultáneas sobre el laboratorio y analizar el tiempo de respuesta del sistema. Es decir, el programa actúa como si hubiera hasta 60 usuarios enviando simultáneamente comandos de configuración al laboratorio.

Obviamente, el tiempo de respuesta del sistema depende de diversos factores: tiempo necesario para que los equipos se configuren, tiempo requerido por los instrumentos para alcanzar los valores introducidos (caso de la fuente de alimentación y generador de funciones), tiempo de apertura/cierre de los relés de la matriz de conmutación, latencia del interfaz de comunicación y velocidad de procesamiento del ordenador en el cual se está ejecutando la aplicación de control. Para esta prueba hemos considerado que los tiempos que vamos a obtener son el sumatorio de todos ellos, sin diferenciar cuánto retardo introduce cada uno de estos elementos al tiempo de respuesta total del sistema.

Para tratar de obtener conclusiones más detalladas sobre la capacidad de respuesta del sistema, hemos realizado 3 pruebas diferentes, en las cuales enviamos al laboratorio circuitos con más o menos conexiones, y con diferentes configuraciones de instrumentos:

- Prueba 1: experimento sencillo, basado en el ejercicio 1 descrito en el ensayo anterior.
- Prueba 2: experimento de complejidad media, basado en el ejercicio 3 descrito en el ensayo anterior.
- Prueba 3: experimento difícil, basado en el ejercicio 5 del ensayo anterior.

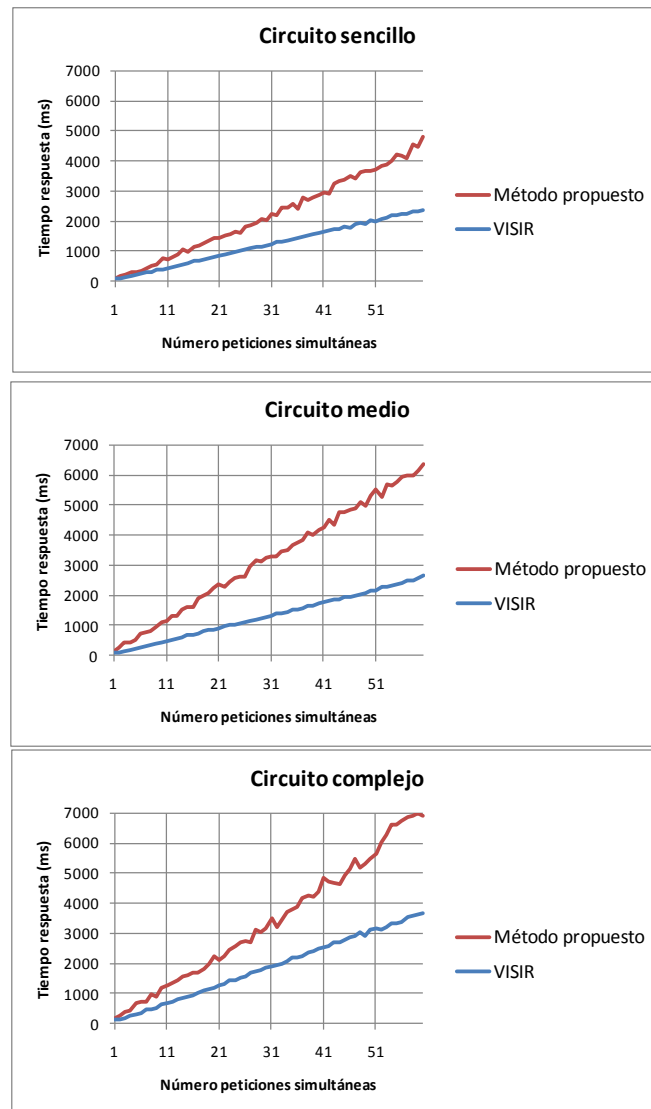


Figura 4.5: Tiempos de respuesta del sistema de control respecto a diferentes tipos de circuitos

Igualmente hemos llevado a cabo estas mismas pruebas sobre la plataforma VISIR con el objetivo de comparar ambos sistemas y cotejar las diferencias entre ambos métodos de control. Los resultados obtenidos se muestran en la Figura 4.5. De esta gráfica se extraen los valores expuestos en la Tabla 4.7, en la cual se indica el tiempo de respuesta medido en milisegundos, tanto del sistema propuesto como del sistema del laboratorio VISIR, respecto a un número determinado de usuarios enviando peticiones simultáneamente al laboratorio.

Núm. Usuarios	Circuito Fácil		Circuito Medio		Circuito Difícil	
	VISIR	Propuesto	VISIR	Propuesto	VISIR	Propuesto
10	408,5ms	777,1ms	444,7ms	1129,4ms	631,7ms	1189,4ms
30	1195,0ms	2019,6ms	1287,4ms	3235,6ms	1874,4ms	3554,3ms
60	2347,9ms	4803,5ms	2662,2ms	6341ms	3679,9ms	6919,3ms

Tabla 4.7: Tiempos de respuesta del sistema de control propuesto y el sistema VISIR

Se observa cómo a medida que aumenta la complejidad del circuito, el sistema necesita más tiempo para proporcionar una respuesta, ya que se tienen que accionar más relés, y las configuraciones de los instrumentos pueden llevar más tiempo. Como es obvio, igual sucede al aumentar el número de peticiones simultáneas sobre el sistema: al recibir 60 peticiones, el usuario situado en la última posición de la cola, tarda más tiempo en recibir la respuesta del sistema.

Además observamos como en las tres pruebas el sistema de control del laboratorio VISIR es más rápido que el propuesto en este trabajo. Esto es debido a los siguientes motivos:

- Según el fabricante de los relés utilizados en la matriz de conmutación del sistema VISIR, su tiempo de conmutación es de 2ms. En cambio los relés de las matrices de conmutación empleados en el sistema propuesto tienen un tiempo de conmutación de 4ms.
- La red de instrumentos creada con nuestra solución, basándonos en el interfaz LXI, aún estando a una distancia inferior a 2 metros del equipo en el cual se ejecuta la aplicación de control, presenta una velocidad de respuesta inferior a la que dispone el sistema VISIR. En la solución del proyecto VISIR todos los equipos, incluido el ordenador que ejecuta esta aplicación, se encuentran integrados en un chasis PXI cuyo bus de datos presenta una tasa de transmisión de 132Mb/s, frente a los 100Mb/s de la red de instrumentos LXI utilizada en el escenario utilizado en nuestras pruebas.

- El equipo en el cual actualmente estamos ejecutando nuestra aplicación de control dispone de un procesador a 1,66GHz, mientras que el controlador embebido en el chasis PXI utilizado en el VISIR tiene un procesador de 2GHz. Esto hace que si utilizamos la utilidad que proporciona LabVIEW denominada *LabVIEW Profile performance*, el tiempo de ejecución del algoritmo principal del sistema de control sea de 31,250 milisegundos en el caso del sistema VISIR, y de 46,875 milisegundos en el caso del método propuesto.

De manera gráfica podemos observar en la Figura 4.6 que ambos sistemas presentan la misma progresión en el tiempo de respuesta respecto al número de peticiones simultáneas realizadas. Se advierte también cómo el sistema VISIR es considerablemente más rápido que el sistema propuesto, ya que incluso ante un circuito difícil, su tiempo de respuesta es menor que el del sistema propuesto ante un circuito considerado como fácil. Vemos además como la proporcionalidad entre el tiempo de respuesta de un sistema y otro en función del tiempo de circuito se mantiene en torno al 50%, es decir, en prácticamente todos los casos el sistema VISIR es el doble de rápido que el sistema propuesto.

Aún así, los tiempos de respuesta del sistema son válidos, ya que en ningún momento el usuario estará más de 7 segundos esperando su respuesta, instante a partir del cual perderá interés en la misma [BKB00]. Debemos tener en cuenta que en esta prueba se está forzando al sistema a dar respuesta a 60 peticiones realizadas simultáneamente sobre el laboratorio. Es decir, se está recreando la situación en la que en el mismo instante de tiempo, 60 usuarios pulsan el botón de ejecución en el cliente web. En el transcurso normal de uso del laboratorio, este escenario puede tener pocas probabilidades de producirse bien porque no hay tantos usuarios conectados o bien porque aun habiendo más de 60 usuarios, la probabilidad de que pulsen sobre el botón de ejecución del laboratorio simultáneamente es baja.

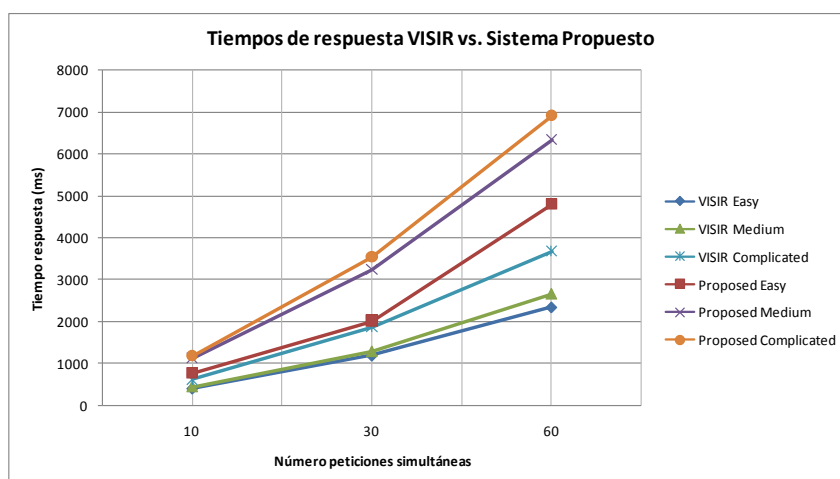


Figura 4.6: Tiempo de respuesta sistema VISIR vs. Sistema propuesto

Por tanto, observando los resultados podemos decir que el sistema de control empleado ofrece una velocidad de respuesta adecuada en el supuesto de que hasta 60 alumnos envíen simultáneamente sus peticiones al laboratorio. Podríamos además mejorar este tiempo de respuesta, aumentando la velocidad del procesador en el cual se ejecuta la aplicación de control y reduciendo al máximo la distancia entre ésta y los instrumentos del laboratorio.

4.2.4 Comparativa de medidas respecto a escenario manual y laboratorio VISIR

El objetivo de esta prueba es comprobar si mediante el método de control que proponemos en este documento, obtenemos medidas diferentes a las que obtenemos utilizando otros escenarios. En concreto, vamos a comparar las medidas obtenidas mediante el laboratorio que utiliza el método propuesto, con las obtenidas utilizando el laboratorio VISIR, así como las obtenidas realizando las mismas medidas pero en un laboratorio manual utilizando para ello los mismos componentes e instrumentos que empleamos en las pruebas remotas. De este modo, podremos evaluar si el sistema planteado introduce errores significativos en las medidas adquiridas a través de él.

Los escenarios sobre los que se plantean las siguientes pruebas son:

- Emplearemos el laboratorio desplegado usando el método de control propuesto, pero considerando que hay varios usuarios accediendo simultáneamente. Utilizaremos para este análisis los resultados

obtenidos anteriormente en la segunda prueba de esta fase de experimentación.

- Repetiremos las medidas sobre los mismos circuitos, pero accediendo únicamente un usuario al sistema. Podremos analizar de este modo si la concurrencia de accesos influye sobre la exactitud de las medidas tomadas.
- Colocaremos exactamente los mismos componentes utilizados en estos circuitos en la plataforma VISIR y repetiremos las medidas llevadas a cabo en el punto anterior.
- Utilizando ahora los mismos componentes, realizaremos el circuito utilizando una placa de prototipos y repetiremos las medidas anteriores empleando para ello los instrumentos utilizados para validar la solución propuesta y repetiremos las medidas, pero utilizando los instrumentos del laboratorio VISIR. Es decir, repetiremos las medidas tal y como si se tratara de una práctica en un laboratorio manual. De este modo se comprueba cómo si el sistema de control y medida remoto interfiere en las medidas adquiridas, al compararlas con las medidas tomadas a mano sobre los mismos circuitos y componentes.

En la Tabla 4.8 se muestran los resultados obtenidos al realizar todas estas medidas sobre los modelos de circuitos contemplados en el ejercicio 1 (Tabla 4.2). En primer lugar podemos advertir cómo apenas hay diferencia entre las medidas obtenidas mediante la solución propuesta en el caso de tener un único alumno accediendo al laboratorio en comparación con los valores medios obtenidos al tener 15 alumnos accediendo simultáneamente sobre el laboratorio. Es decir, la concurrencia de accesos no está modificando los valores obtenidos sobre los circuitos.

Igual sucede si comparamos las medidas obtenidas por un único usuario con las medidas obtenidas sobre los mismos circuitos, pero realizando la experiencia tal y como si estuviéramos en un laboratorio manual y midiendo el circuito con los mismos instrumentos que disponemos en nuestra solución.

Ejercicio 1		Solución propuesta			Plataforma VISIR	
		Valor medio 5 usuarios	1 único usuario	Medida lab. manual	1 único usuario	Medida lab. manual
Modelo 1	Frecuencia 1kHz	999,4Hz	1000Hz	1000Hz	1000Hz	999Hz
	Tensión 10Vpp	19,998V	19,98V	20,8V	19,94V	20,7V
Modelo 2	Frecuencia 60Hz	60,096Hz	60Hz	60Hz	59,88Hz	60Hz
	Tensión 8Vpp	16,136V	16,11V	16,3V	16,02V	15,89V
Modelo 3	Frecuencia 800Hz	799,2Hz	800Hz	800Hz	800Hz	800Hz
	Tensión 3Vpp	6,108V	6,01V	6,7V	6,05V	6,1V

Tabla 4.8: Comparativa realizada respecto a la plataforma VISIR en el ejercicio 1

Ejercicio 5		Solución propuesta			Plataforma VISIR	
		Valor medio 5 usuarios	1 único usuario	Medida lab. manual	1 único usuario	Medida lab. manual
Modelo 1	Frecuencia 50Hz	49,99Hz	50Hz	50Hz	50Hz	49,98Hz
	Tensión 5,6Vpp	5,1544V	5,125V	5,6V	5,01V	5,5V
Modelo 2	Frecuencia 100Hz	100Hz	100Hz	100Hz	100Hz	100Hz
	Tensión 10Vpp	9,82V	10V	10V	9,61V	9,76V
Modelo 3	Frecuencia 75Hz	74,96Hz	74,91Hz	75,8Hz	75,12Hz	75,12Hz
	Tensión 0,81Vpp	0,8724V	0,875V	0,81V	0,87V	0,865V

Tabla 4.9: Comparativa realizada respecto a la plataforma VISIR en el ejercicio 5

Se llega a la misma conclusión si se comparan estas medidas con las obtenidas utilizando la plataforma remota VISIR o los instrumentos de este laboratorio pero de manera local sobre los circuitos montados sobre una placa de prototipos, usando para ello siempre los mismos componentes.

Las medidas mostradas en la Tabla 4.9 se corresponden con los valores adquiridos sobre la señal de salida del circuito una vez rectificadora y filtrada del ejercicio 5. Podemos ver en primer lugar cómo apenas hay diferencias en las medidas adquiridas cuando solo hay un usuario accediendo al laboratorio o varios de manera concurrente. De igual modo, no existen diferencias

reseñables cuando se realiza el ejercicio de manera local con cualquiera de los dos laboratorios.

Si comparamos los resultados obtenidos de manera remota o manual, se observa cómo las medidas de frecuencia apenas sufren variación mientras que en las medidas de tensión hay algunas diferencias, aunque no modifican en ningún caso los resultados obtenidos de manera significativa.

En la Figura 4.7 aparecen representados los errores en las medidas de frecuencia y tensión pico a pico de la señal periódica en cada uno de los tres escenarios y en los ejercicios 1 y 5 para cada uno de los seis modelos de ejercicios analizados.

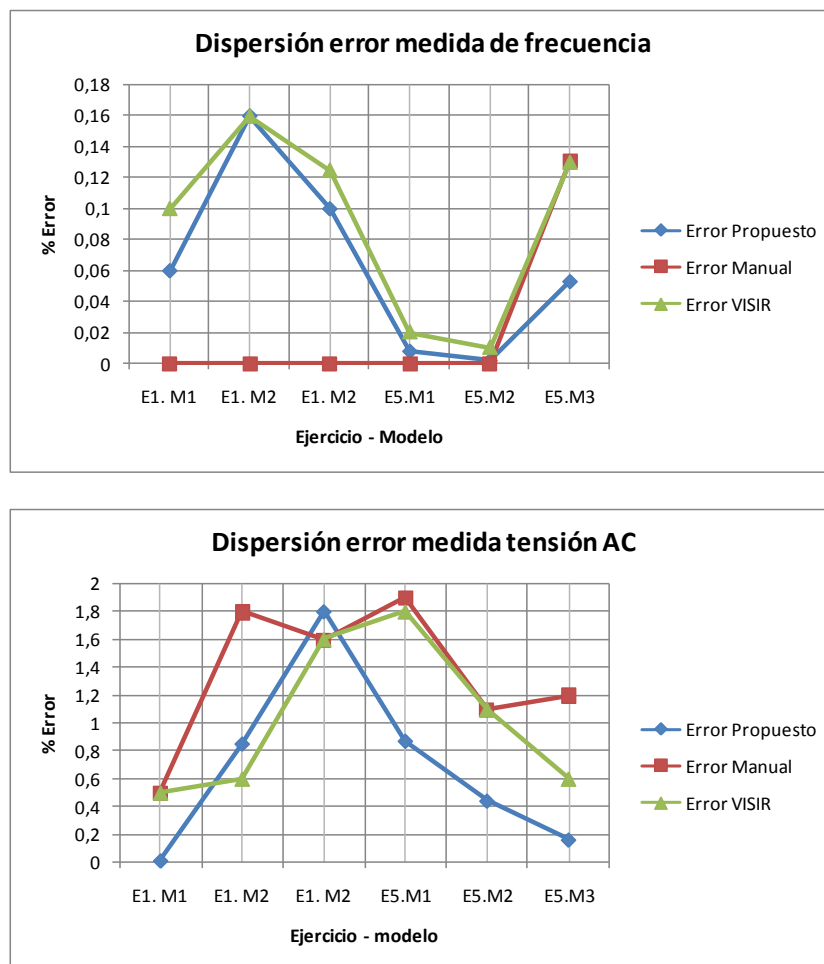


Figura 4.7: Dispersión error en medida de frecuencia y tensión AC

Se puede observar cómo en la medida de frecuencia, el porcentaje de error tanto en el laboratorio VISIR como en el laboratorio que despliega el sistema de control propuesto es prácticamente el mismo. En las medidas realizadas en un laboratorio manual el error es prácticamente nulo en todas las medidas, obteniéndose únicamente una medida en la que el error es del 0,13%.

En el caso de las medidas de tensión de una señal periódica, los tres escenarios presentan errores muy similares en cada uno de los seis modelos de circuitos analizados, si bien el sistema VISIR es el que más semejanzas presenta al laboratorio manual y es en este caso el sistema propuesto el que introduce un menor error en las medidas adquiridas.

Si analizamos ahora los resultados obtenidos en los ejercicios 2 y 3 y que aparecen reflejados en la Tabla 4.10, observamos cómo al igual que en el caso anterior no existen diferencias entre las medidas adquiridas cuando hay varios usuarios accediendo concurrentemente al laboratorio que hace uso de la solución propuesta, o cuando las medidas son realizadas por un único usuario.

Ejercicio 2	Solución propuesta			Plataforma VISIR	
	Valor medio 5 usuarios	1 único usuario	Medida lab. manual	1 único usuario	Medida lab. manual
Modelo 1 Valor teórico: 1470 Ω	1462,4 Ω	1468,1 Ω	1456 Ω	1463,6 Ω	1456 Ω
Modelo 2 Valor teórico: 11k Ω	10883,4 Ω	10965,5 Ω	10880 Ω	10913 Ω	10881 Ω
Modelo 3 Valor teórico: 10,470k Ω	10468,8 Ω	10456,5 Ω	10356 Ω	10357 Ω	10357 Ω

Ejercicio 3	Solución propuesta			Plataforma VISIR	
	Valor medio 5 usuarios	1 único usuario	Medida lab. manual	1 único usuario	Medida lab. manual
Modelo 1 Valor teórico: 909 Ω	901,80 Ω	904,08 Ω	899,73 Ω	903,38 Ω	898,7 Ω
Modelo 2 Valor teórico: 448 Ω	447,20 Ω	448,14 Ω	445,6 Ω	447,95 Ω	445,3 Ω
Modelo 3 Valor teórico: 319 Ω	319,75 Ω	318,94 Ω	317,05 Ω	320,71 Ω	317,13 Ω

Tabla 4.10: Comparativa realizada respecto a la plataforma VISIR en los ejercicios 2 y 3

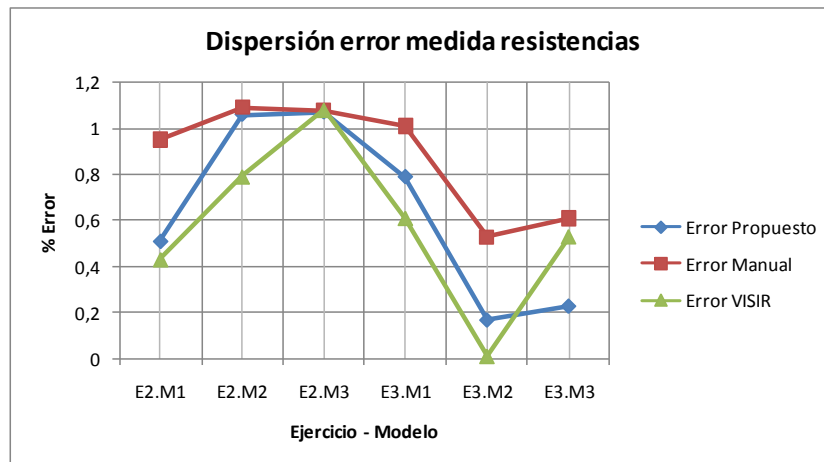


Figura 4.8: Dispersión error en medida de resistencias

Vemos en estos resultados cómo sí que existen pequeñas diferencias cuando comparamos las medidas adquiridas a través del laboratorio remoto con las medidas obtenidas directamente en el laboratorio manual. Vemos que estas diferencias también existen cuando comparamos los resultados obtenidos utilizando el laboratorio VISIR o los instrumentos del laboratorio VISIR de manera local.

Estas diferencias pueden ser debidas a las resistencias introducidas en el sistema de medida por las matrices de conmutación empleadas en ambos casos, ya que si comparamos los resultados obtenidos cuando un único usuario realiza las medidas de manera remota con ambas soluciones, las medidas obtenidas son prácticamente iguales.

En la Figura 4.8 se ha representando el porcentaje de error que aparece en las medidas de resistencias en los tres escenarios y empleando los seis modelos diferentes de circuitos. Podemos observar como en cada uno de los modelos, el error es muy similar en los tres escenarios evaluados durante esta prueba.

Podemos concluir por tanto que el método de control que desarrollado nos permite realizar medidas de resistencias sobre circuitos construidos remotamente sin introducir errores significativos en las mismas, o al menos no mayores que los introducidos por el sistema de control y medida de referencia como es el laboratorio VISIR.

Ejercicio 4	Solución propuesta			Plataforma VISIR	
	Valor medio 5 usuarios	1 único usuario	Medida lab. manual	1 único usuario	Medida lab. manual
Modelo 1 Valor teórico: 12,59V	12,58V	12,59V	12,59V	12,59V	12,59V
Modelo 2 Valor teórico: 4,83V	4,82V	4,82V	4,83V	4,82V	4,83V
Modelo 3 Valor teórico: 4,43V	4,44V	4,44V	4,41V	4,44V	4,41V

Tabla 4.11: Comparativa realizada respecto a la plataforma VISIR en el ejercicio 4

En lo que respecta a la medida de tensiones llevadas a cabo en el ejercicio 4 cuyos resultados aparecen en la Tabla 4.11, se observa que bien utilizando el laboratorio que emplea nuestra solución o bien el laboratorio VISIR, las medidas obtenidas son las mismas sin apenas desviación y diferencias entre las mismas. Existen unas diferencias mínimas (debemos recurrir para ello hasta valores con 3 y 4 decimales) respecto a los valores obtenidos en los laboratorios manuales, las cuales pueden ser debidas a las matrices de conmutación empleadas en ambos escenarios. Podemos establecer por tanto que el método de control propuesto ofrece gran estabilidad en las medidas y configuración remota en un experimento como el planteado en el ejercicio 4.

Si al igual que en las pruebas anteriores se representa el porcentaje de error en cada uno de los 3 escenarios para cada uno de los tres modelos analizados se obtiene la Figura 4.9. Se observa como en todos ellos el error es menor del 0,3% y cómo para cada uno de los tres modelos, el error de las medidas en los tres escenarios es muy similar entre sí.

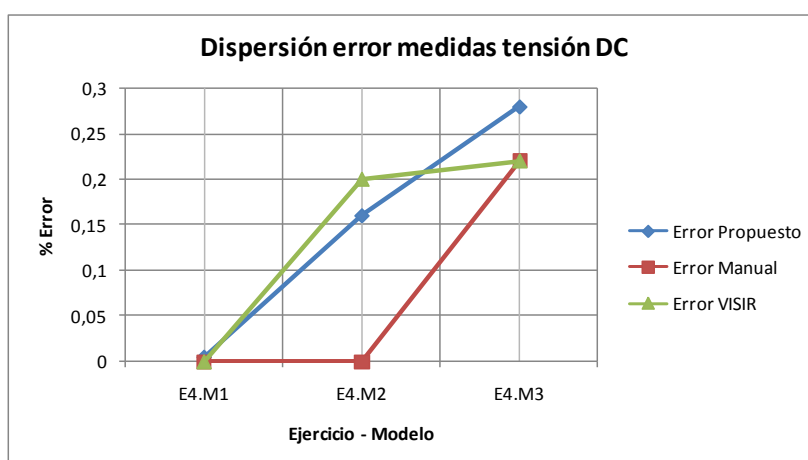


Figura 4.9: Dispersión error en medida tensión DC

A modo de conclusión de este conjunto de pruebas, en lo que respecta a medidas de resistencias observamos cierta variabilidad en las mismas, tanto utilizando el laboratorio que emplea nuestra solución como en el caso de usar el laboratorio VISIR. Estas variaciones pueden ser debidas a las tolerancias de las resistencias (recordemos que se ha utilizado siempre los mismos componentes en los diferentes escenarios) y a que los sistemas de medida remotos empleados en ambos laboratorios pueden favorecer estas diferencias en los valores obtenidos. Los dos sistemas realizan las medidas durante instantes de tiempo relativamente cortos, en los cuales se ejecutan las órdenes de configuración de los instrumentos así como la apertura y cierre de los relés correspondientes de las matrices empleadas. Es posible que este lapso de tiempo sea demasiado corto y el sistema de medida no llegue a alcanzar un estado permanente estable. La solución sería ampliar este tiempo de captura de la medida, pero estaríamos ralentizando el funcionamiento del sistema global. Aún así, los valores obtenidos se pueden considerar válidos, por lo que el sistema desarrollado puede ser utilizado para realizar estas medidas.

En lo que respecta a las medidas de tensiones en continua, los valores obtenidos son muy precisos y prácticamente iguales en todos los escenarios. En cambio, si medimos tensiones sobre señales de una determinada frecuencia, si observamos cierta variabilidad en las medidas. Al igual que en el caso anterior, puede ser debido al instante de tiempo en cual se efectúa el disparo de captura en el osciloscopio, así como a las resistencias de error introducidas por las matrices y módulos de expansión empleados. De cualquier modo, los valores obtenidos mediante el laboratorio que hace uso del método de control propuesto, son totalmente válidos si los comparamos con los obtenidos en la plataforma VISIR y con los obtenidos si realizamos las mismas medidas en un escenario de laboratorio manual.

Gracias a este conjunto de ensayos hemos comprobado que el método propuesto puede ser desplegado en un laboratorio remoto, ya que proporciona prácticamente las mismas respuestas que las adquiridas de manera manual y las obtenidas empleando un laboratorio remoto ampliamente empleado y verificado como es el caso del laboratorio VISIR.

4.2.5 Caracterización del sistema de control de experimentos

El objetivo de esta prueba es caracterizar el sistema de control de experimentos incluido en el método propuesto, el cual está formado por las matrices de conmutación y los módulos de expansión. Como veíamos en las pruebas anteriores, en algunos casos las medidas de tensiones pueden ser diferentes respecto a las obtenidas en un laboratorio manual sobre el mismo circuito debido a las resistencias inherentes a la circuitería de las matrices y los módulos de expansión.

Hemos realizado dos ensayos por separado. El primero de ellos está destinado a obtener el valor de la resistencia introducida en el circuito al cerrar uno de los relés de la matriz de conmutación. De esta manera conoceremos el valor de esta resistencia, considerando únicamente los circuitos propios de la matriz. En el segundo ensayo realizaremos las mismas medidas, pero considerando los módulos de expansión, de modo que así podremos calcular qué resistencia parásita o de error introducen estos módulos en el sistema de medida. Ambos ensayos se han realizado sobre distintos puntos de la matriz de conmutación y de los módulos de expansión.

Al tratarse de resistencias muy pequeñas las que vamos a medir se ha empleado el método de medida de resistencia de 4 hilos. Básicamente el concepto de medida está basado en la ley de Ohm y para ello necesitamos un multímetro digital que pueda realizar estas medidas. Mediante este método, presentado en la Figura 4.10, separamos la señal de medida generada de la señal capturada y empleada por el instrumento para mostrar al usuario el resultado de la resistencia medida.

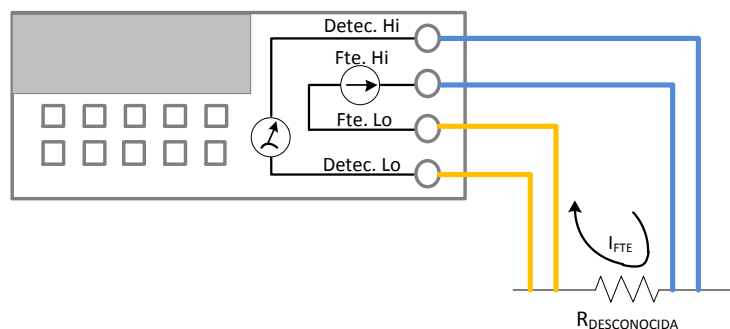


Figura 4.10: Esquema de medida para caracterización de resistencias

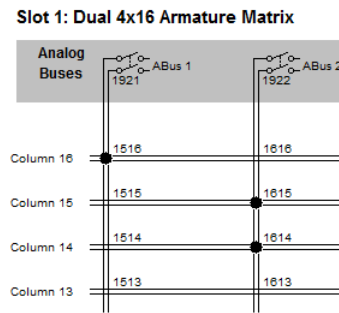


Figura 4.11: Ejemplo de relé cerrado mediante servicio web del mainframe 34480^a

En el ejemplo de la Figura 4.11, el punto negro representa el clic ejecutado a través del interfaz web proporcionado por el equipo y que nos permite cerrar el relé existente entre la columna 16 y la fila 1 de la matriz 2 situada en el slot 5 del mainframe. De este modo si conectamos el sistema de medida entre la columna 16 y la fila 1 de dicha matriz, estaremos obteniendo en el multímetro el valor de la resistencia comprendida entre ambos puntos.

Para el segundo ensayo se ha utilizado el mismo sistema de medida pero conectando ahora a la matriz de conmutación los módulos de expansión. Mediante este ensayo se quiere confrontar las diferencias en los valores de resistencias medidas en los mismos puntos pero teniendo en cuenta los cables de conexión empleados entre las matrices y los módulos así como la circuitería propia de éstos.

Las medidas han sido tomadas en puntos elegidos al azar de entre todas las combinaciones posibles. Los valores mostrados en la Tabla 4.12 son los obtenidos en ambos ensayos, y están expresados en ohmios.

Columna – Fila	Slot	Matriz	Resistencia matriz	Resistencia módulos
C4 – R7	5	2	0,281Ω	0,311Ω
C12 – R6	5	2	0,318Ω	0,320Ω
C9 – R5	5	2	0,317Ω	0,324Ω
C16 – R1	1	1	0,315Ω	0,321Ω
C2 – R3	1	1	0,317Ω	0,322Ω
C14 – R4	1	1	0,316Ω	0,327Ω

Tabla 4.12: Caracterización del sistema de control de experimentos

En relación a los resultados obtenidos, vemos como el valor medio de las resistencias medidas sobre la matriz de conmutación es de $0,310\Omega$ y de $0,320\Omega$ en el caso de considerar los módulos de expansión. Como primera conclusión al respecto, podemos decir que los módulos de expansión utilizados apenas introducen resistencias parásitas en las medidas. Recordemos que estos módulos consistían únicamente en pistas o cables utilizados para interconectar las matrices entre sí, así como los componentes de los circuitos a la matriz 8×32 resultante, por lo que los únicos elementos que pueden introducir cierto error en las medidas son estos cables y conectores.

Además, gracias al sistema de configuración de la matriz resultante, así como el modelo de conexionado de los componentes e instrumentos utilizados, estas resistencias parásitas o de error se encuentran conectadas en serie respecto a los componentes y los instrumentos. De esto modo, si consideramos el circuito de la Figura 4.12 en el cual queremos medir una resistencia de valor 100Ω , a la incertidumbre introducida en la medida por la tolerancia del 10% incluida por defecto en el valor de fábrica del componente, debemos añadir las resistencias de los cuatro relés necesarios para la medida, así como las de los puntos de conexión del instrumento y el componente en el módulo de expansión. Considerando el valor de cada una de estas resistencias $0,320\Omega$, el valor de resistencia parásita introducida en la medida es de $1,24\Omega$, un valor suficientemente pequeño como para no variar de manera significativa la medida de la resistencia bajo prueba.

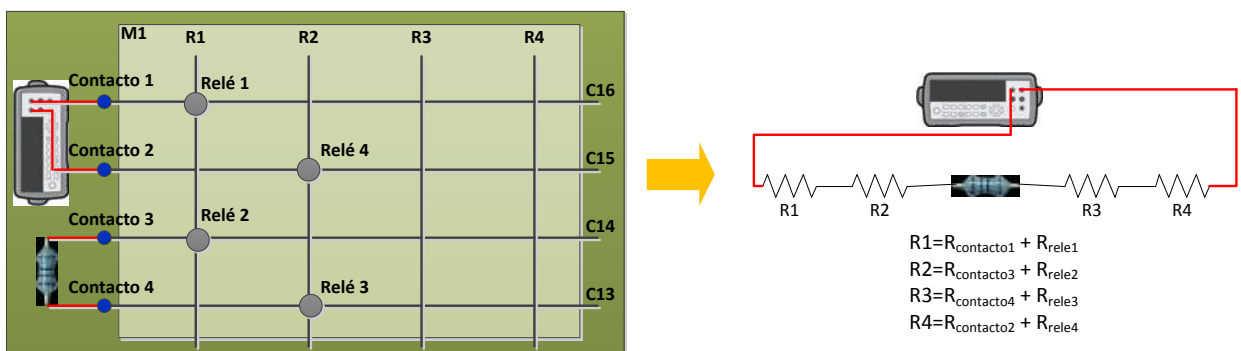


Figura 4.12: Ejemplo de interferencia de resistencias parásitas en medida de resistencias

En el caso de estar realizando medidas de tensión, al ser valores de resistencia de miliohmios, la caída de tensión en cada una de ellas es muy pequeña, aunque como hemos visto en las pruebas anteriores, sí puede que modifique los resultados obtenidos en algunas décimas, pero en todo caso estas variaciones pueden ser asumibles y considerar los valores obtenidos como válidos.

Por tanto podemos finalizar diciendo, que si bien el conjunto formado por la matriz de conmutación y los módulos de expansión, introducen ciertas resistencias parásitas en las medidas, los efectos que éstas producen en las medidas son mínimos y perfectamente asumibles, por lo que el sistema de construcción y medida de circuitos remoto puede ser considerado como válido.

4.2.6 Rendimiento del sistema de control respecto al sistema empleado en el laboratorio VISIR

El objetivo de esta prueba es comprobar cómo el sistema de control de experimentos empleado en el método propuesto, presenta un rendimiento mejor respecto al empleado en el proyecto VISIR, en relación al número de componentes necesarios para la implementación física de circuitos. Este mayor rendimiento permitirá que con el mismo número de componentes, podremos realizar más circuitos que utilizando la plataforma VISIR obteniendo los mismos resultados en las medidas, tal y como veíamos en pruebas anteriores.

Ya en el capítulo anterior veíamos un sencillo ejemplo en el que para realizar distintas configuración con tres resistencias, utilizando este método necesitábamos únicamente 3 resistencias, y en cambio utilizando el sistema VISIR eran necesarias al menos 6 resistencias.

Aún así vamos a analizar un circuito más complejo para obtener más datos comparativos respecto a los dos sistemas. Supongamos que queremos proporcionar a los usuarios un circuito como el mostrado en la Figura 4.13 con el objetivo de que puedan medir las tensiones e intensidades mostradas, puenteando o no la resistencia R1 a la entrada del circuito.

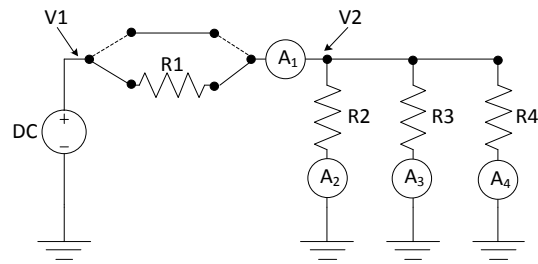


Figura 4.13: Ejercicio de rendimiento del sistema de control propuesto

Aparte de las intensidades mostradas en la Figura 4.13, para dar más opciones de configuración y medida al usuario, queremos que éste pueda medir la corriente que circula por cada combinación en paralelo posible de entre las resistencias R2, R3 y R4 del circuito, es decir, que pueda realizar los montajes y medidas adicionales mostradas en la Figura 4.14.

Para implementar estos circuitos y poder realizar estas medidas mediante el laboratorio VISIR, es necesario utilizar 8 resistencias y hasta 14 cortocircuitos en el caso de que se quiera dar la opción al usuario de realizar las medidas de intensidad A1, A2, A3, y A4, antes o después de las resistencias de las resistencias correspondientes. Es decir, de manera gráfica, se debe montar sobre la matriz de conmutación del laboratorio VISIR el esquema de la Figura 4.15.

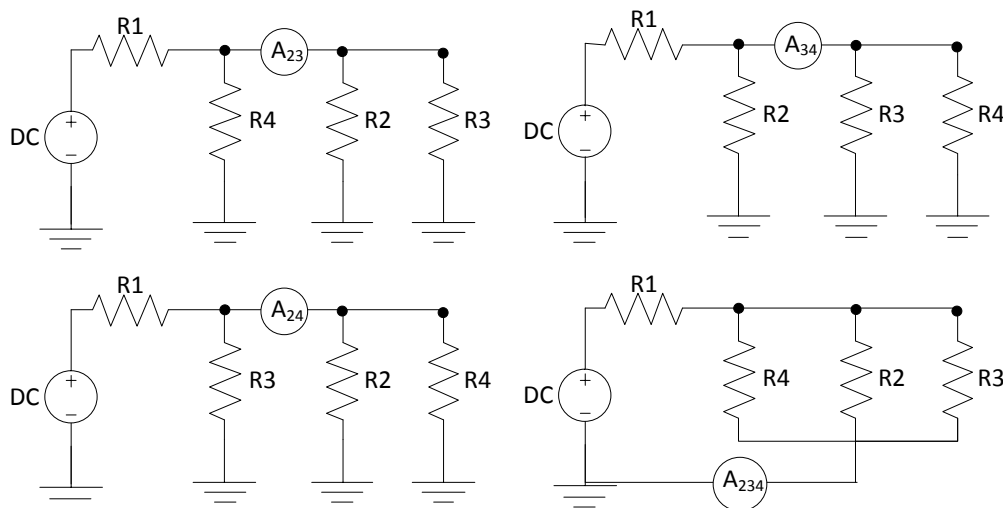


Figura 4.14: Medidas de intensidad adicionales en el test de rendimiento

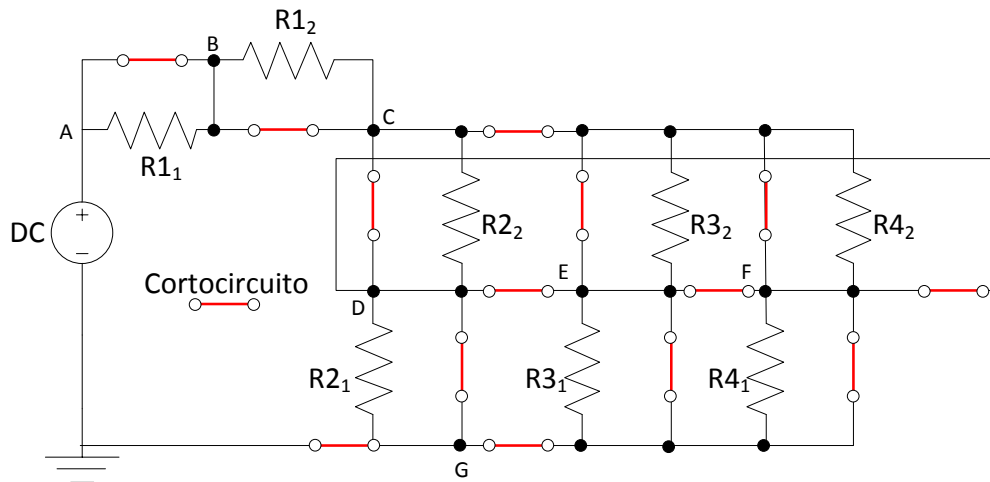


Figura 4.15: Componentes y cortocircuitos necesarios en el laboratorio VISIR

Esto implica la necesidad de utilizar 2 tarjetas completas de la matriz de conmutación, así como realizar todos los cableados necesarios para conectar cada componente y cada cortocircuito a su nodo correspondiente tal y como se puede ver en la Figura 4.16. No olvidemos que en la solución adoptada en el VISIR, un cortocircuito también ocupa un lugar en la matriz de conmutación, ya que se trata de un componente más.

Vemos además cómo en este sistema cada componente únicamente puede estar cableado entre dos nodos de la matriz, por lo que para implementar el ejemplo expuesto necesitamos un par de resistencias de cada valor para poder completar el circuito.

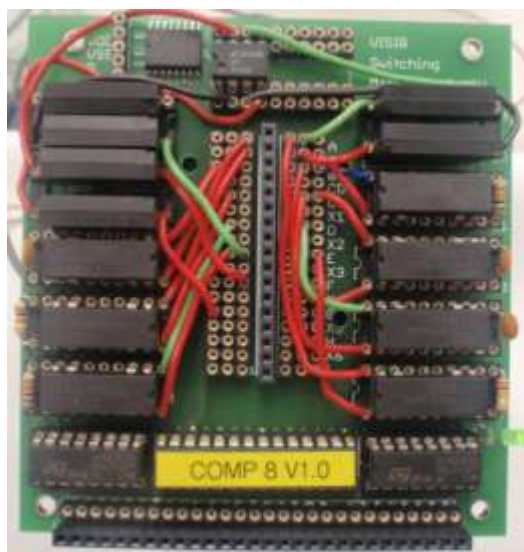


Figura 4.16: Ejemplo de conexionado de componentes de la plataforma VISIR

La razón de necesitar tantos cortocircuitos en la plataforma VISIR (Figura 4.15) radica en que para poder efectuar medidas de intensidad, al igual que un laboratorio manual, es necesario introducir el multímetro en serie en la rama en la cual se desea medir la intensidad.

Ahora bien, si queremos realizar las mismas medidas y configuraciones utilizando el sistema aquí propuesto, el esquema que utilizaríamos sería el indicado en la Figura 4.15, pero eliminando todos los cortocircuitos y los componentes duplicados. Es decir, para implementar estas medidas mediante el sistema planteado, se necesitan únicamente 4 resistencias y ningún cortocircuito incluido como componente en la matriz. Recordemos que en este sistema tanto las conexiones de los componentes a los nodos, como los cortocircuitos se crean dinámicamente abriendo o cerrando los relés de la matriz de conmutación en función del circuito descrito por el usuario y enviado desde el interfaz web.

De este modo eliminamos completamente la necesidad de realizar ningún cableado físico entre los componentes y los nodos, así como incluir cortocircuitos como componentes, ahorrando de esta manera espacio en la matriz. Aún así, cuando definamos el fichero .max utilizado por el instructor virtual para validar que el circuito se puede construir sobre nuestra matriz, sí debemos especificar las ramas del circuito en el cual se permiten hacer medidas de intensidad. Si consideramos que $R1=1k\Omega$, $R2=10k\Omega$, $R3=100k\Omega$ y $R4=470\Omega$, el fichero .max utilizado en el sistema propuesto sería:

```
VDC+25V_1  F    max:20    imax:0.5
VDCCOM_1   0
R_R2       C 0 1K
R_R3       B 0 10K
R_R4       B 0 1K
R_R5       C 0 10K
R_R6       A B 1k
R_R7       C 0 470
R_R8       B 0 470
R_R10      B 0 100k
R_R11      C 0 100k
R_R12      B D 470
R_R13      B D 100k
R_R14      B D 10k
SHORTCUT_S1  B C
SHORTCUT_S2  F A
SHORTCUT_S3  D 0
```

Medida	Valor teórico	Valor medido	Error cometido (%)
V_2	3,089V	3,094V	0,16%
IR_1 (1k)	6,911mA	6,947mA	0,52%
IR_2 (10k)	308,9uA	307,100uA	0,32
IR_3 (100k)	30,89uA	30,590uA	0,97%
IR_4 (470)	6,572mA	6,599mA	0,41%
IR_{34} (100k 470)	6,602mA	6,625mA	0,34
IR_{23} (10k 100k)	339,8uA	339,400uA	0,11%
IR_{24} (10k 470)	6,88mA	6,892mA	0,05%
IR_{1234}	6,911mA	6,975mA	0,92%

Tabla 4.13: Valores obtenidos para la validación de la prueba de rendimiento del sistema

Vemos que en la descripción de este fichero existen componentes repetidos y cortocircuitos, pero esto es debido a que un mismo componente se puede conectar a todos los nodos permitidos de la matriz de manera dinámica en función de la descripción del circuito realizada en el fichero .max. Es decir, la resistencia R_2 y la resistencia R_4 son físicamente el mismo componente, solamente que mediante el fichero .max de configuración del instructor virtual que describe los circuitos permitidos en cada ejercicio, permitimos que la resistencia de $1k\Omega$ se conecte bien entre los nodos C-0 y B-0 en base al circuito recibido desde el cliente web. Obviamente, si en un mismo circuito hay dos componentes iguales, necesitaremos disponer de dos componentes iguales también en nuestra matriz.

Hemos comprobado también esta configuración utilizando el sistema propuesto para verificar que se pueden construir remotamente todos estos circuitos y que los valores medidos son válidos, para lo cual los hemos comparado con los valores teóricos obtenidos aplicando en dichos circuitos (Figura 4.14) las leyes de Kirchhoff.

Los resultados obtenidos al realizar estas medidas mediante un laboratorio que despliegue el método de control propuesto, considerando la tensión de alimentación de 10 Voltios aparecen reflejados en la Tabla 4.13. Se puede advertir que las medidas adquiridas apenas difieren de los valores

teóricos calculados, manteniéndose en todos los casos un error respecto al valor teórico por debajo del 1%. Por consiguiente, al igual que en las medidas realizadas en las pruebas anteriores, podemos afirmar también que el sistema es válido para realizar medidas de intensidad sobre circuitos construidos remotamente.

Además hemos comprobado cómo para realizar estas configuraciones el rendimiento respecto al número de componentes necesarios es mejor en el sistema propuesto respecto al sistema VISIR, ya que si en éste son necesarias 8 resistencias y 14 cortocircuitos para poder realizar todas las configuraciones posibles mostradas en la Figura 4.14, mediante el método de control presentado únicamente se necesitan 4 resistencias y 0 cortocircuitos.

Esta misma prueba de rendimiento ha sido realizada con otro conjunto de circuitos, cuyos resultados aparecen expuestos en la Tabla 4.14. Se puede observar como la tendencia en todos ellos es a emplear la mitad de componentes si se emplea el sistema propuesto. En algunas situaciones, como en el caso de querer realizar circuitos con el mismo conjunto de resistencias, el sistema VISIR permite utilizar cortocircuitos para evitar tener que emplear componentes repetidos, pero recordemos que en dicha plataforma un cortocircuito ocupa un lugar en la matriz como un componente más, por lo que la suma total de componentes y relés necesarios para crear el circuito bajo prueba sería la misma.

En esta prueba, en el caso de los circuitos en los que se emplean diodos, es muy interesante observar cómo en la plataforma VISIR es necesario tener en cuenta la polaridad del diodo, situación que obliga a disponer de dos componentes iguales en la matriz para poder realizar experimentos con distintas polarizaciones, tanto si se trata de un diodo convencional como de un diodo Zener.

En el último set de circuitos en los cuales se analizan distintas configuraciones de amplificador operacional, la solución más sencilla implica que los terminales de este componente estén siempre conectados en los mismos nodos de la matriz (B, D, C y G y H para su alimentación). Esta consideración conlleva que en la matriz del sistema VISIR sean necesarias 5 resistencias de $10\text{k}\Omega$ conectadas entre los nodos A_B, B_C, 0_B, A_D y D_0, mientras que empleando el sistema de control propuesto, para este conjunto de circuitos únicamente es necesario disponer de 2 resistencias de $10\text{k}\Omega$, así

como un solo condensador de 2.2nF y no dos como en el caso del VISIR (conectados físicamente a los nodos E_B y B_C).

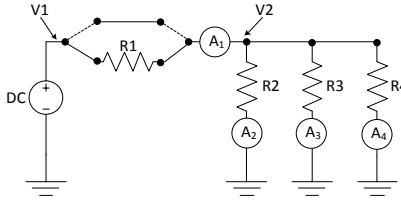
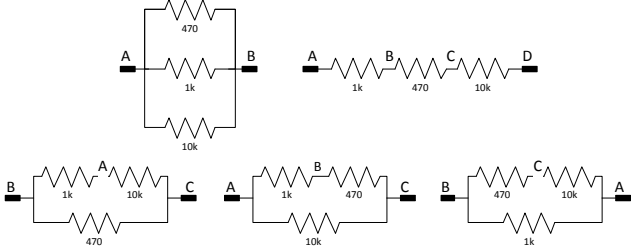
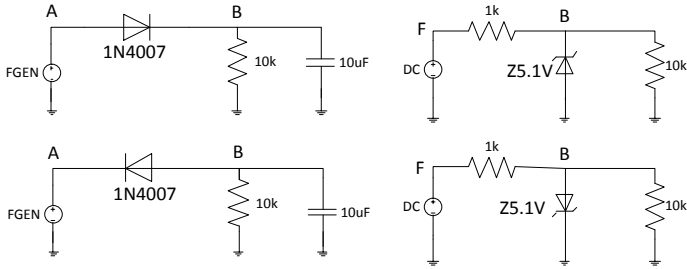
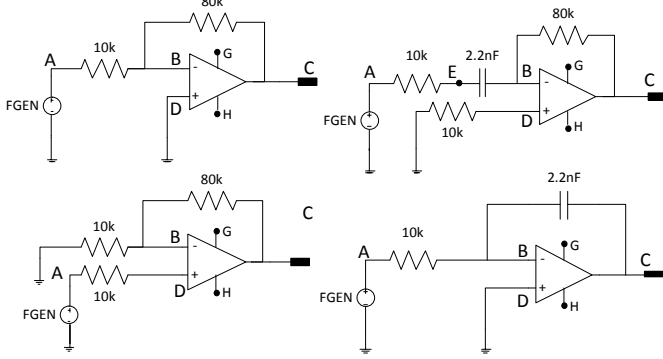
Circuitos	VISIR	Sistema Propuesto
	<p>8 Resistencias 14 Cortocircuitos</p>	<p>4 Resistencias 0 Cortocircuitos</p>
	<p>6 Resistencias</p>	<p>3 Resistencias</p>
	<p>2 Diodos 1N4007 2 Zener 5.1V 2 Resistencias</p>	<p>1 Diodo 1N4007 1 Zener 5.1V 2 Resistencias</p>
	<p>1 A.O. 6 Resistencias 2 Condensadores 1 Cortocircuito</p>	<p>1 A.O. 3 Resistencias 1 Condensador 0 Cortocircuito</p>

Tabla 4.14: Set de circuitos para prueba de rendimiento de la matriz de conmutación

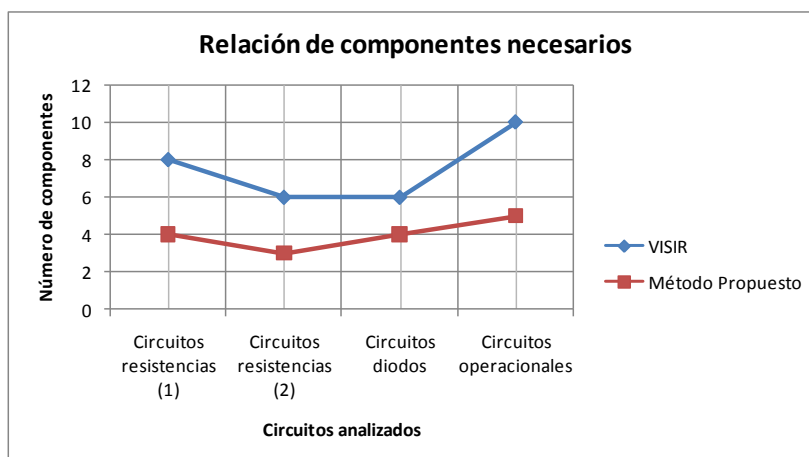


Figura 4.17: Relación componentes necesarios para la realización de circuitos

Por consiguiente, en base las explicaciones realizadas podemos concluir diciendo que el sistema de control propuesto presenta un mejor rendimiento en cuando a número de componentes necesarios respecto al sistema del laboratorio VISIR, ya que como se puede observar en la Figura 4.17 la tendencia en la relación entre ambos sistemas es de 2:1, por lo que se mejora considerablemente el uso del espacio destinado a componentes en la matriz de conmutación.

Como se puede ver en la Figura 4.18, estas diferencias son aún más apreciables si se consideran los cortocircuitos necesarios en la plataforma VISIR, ya que cada uno de ellos ocupa un lugar en la matriz y necesita la acción de un relé para ser incluidos en el circuito.

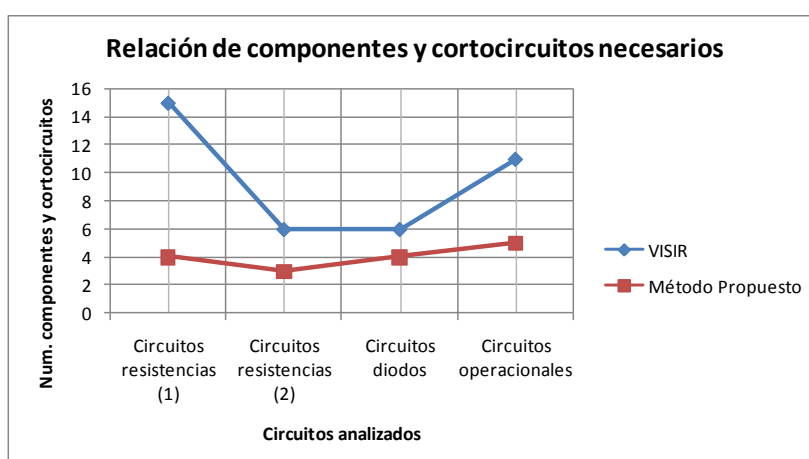


Figura 4.18: Componentes y cortocircuitos necesarios para la realización de circuitos

En la plataforma VISIR el número de componentes y cortocircuitos está directamente relacionado, ya que en los ejemplos analizados se podría reducir el número de componentes a costa de aumentar el número de cortocircuitos, pero aún así se estaría consumiendo espacio físico en la matriz para poder albergar los cortocircuitos necesarios.

Además mediante este sistema, al eliminar las conexiones físicas entre componentes y nodos en la matriz de conmutación, añadir nuevos componentes a los experimentos del laboratorio es más sencillo y rápido que en el caso del sistema VISIR.

4.2.7 Medida de valores de tensión media en señales periódicas

El objetivo de esta prueba es comprobar cómo mediante el método de control y medida propuesto en este trabajo conseguimos solventar el error que aparece en el laboratorio VISIR al conectar el multímetro digital al generador de funciones con el propósito de obtener el valor de tensión media de la señal periódica proporcionada por este equipo. Como veíamos en el capítulo anterior, este error se produce porque el sistema de medida del laboratorio VISIR únicamente toma una muestra de la señal, por lo que dependiendo del instante de tiempo en el que se realice, el valor de tensión capturado es uno u otro.

Para eliminar este error se debe tener en cuenta que la expresión teórica del cálculo del valor medio de la tensión de una señal periódica considera un periodo completo de dicha señal, por lo que es necesario obtener muestras de sus valores de tensión durante dicho tiempo.

Esta prueba ha consistido en dos experimentos realizados tanto mediante la solución propuesta, como con el laboratorio VISIR y en un laboratorio manual en el que se ha empleado el mismo multímetro que en el escenario propuesto para validar el método desarrollado. En el primero de ellos, hemos medido el valor medio de tensión de una señal senoidal de $5V_{pp}$ y 100Hz sin ningún offset de tensión, obteniendo los resultados de la Tabla 4.15.

Experimento 1	Orden de Medida	Solución propuesta (V)	Plataforma VISIR (V)	Medida Lab. Manual (V)
Frecuencia 100Hz Tipo señal: senoidal Amplitud: 5Vpp Offset: 0V	1	0,008414	2,480	0,003
	2	0,028410	-1,219	0,003
	3	-0,021110	-2,146	0,003
	4	-0,001699	-1,557	0,003
	5	0,004073	1,170	0,003
	6	0,001090	-2,414	0,003
	7	0,003940	-1,141	0,003
	8	0,004685	2,427	0,003
	9	0,002067	2,397	0,003
	10	0,001030	-2,164	0,003

Tabla 4.15: Medida de tensión media de señal periódica con Offset 0V

En el segundo experimento, se ha repetido la medida pero sobre una señal cuadrada de $6V_{pp}$ y 300Hz a la que hemos introducido un offset de 2V, siendo las medidas adquiridas las mostradas en la Tabla 4.16. En ambas pruebas, las señales son periódicas y con un ciclo de trabajo del 50%.

En cuanto a los resultados obtenidos, en el primer experimento realizado en esta prueba, el valor medio teórico de la señal periódica es 0V. Podemos ver que las medidas adquiridas en el laboratorio manual, aparte de presentar una desviación típica igual a 0, ofrecen un valor medio de tensión de 0,003V. En las medidas obtenidas mediante el método desarrollado, aún presentando una desviación típica de 19,39V, los valores son del orden de mV, muy distintos a las obtenidas en el laboratorio VISIR, que en ningún caso se acercan a un valor de 0V.

En el segundo experimento hemos introducido una señal cuadrada que presenta un offset de 2V. En este caso vemos cómo las medidas tomadas mediante la solución propuesta son muy cercanas a los valores resultantes del escenario manual y muy próximas al valor teórico de 2V. En cambio, observamos cómo las tomadas en la plataforma VISIR son muy dispares y en ningún caso se acercan al valor de 2V ofrecido en los otros dos escenarios de pruebas.

Experimento 2	Orden de Medida	Solución propuesta (V)	Plataforma VISIR (V)	Medida Lab. Manual (V)
Frecuencia 300Hz Tipo señal: cuadrada Amplitud: 6Vpp Offset: 2V	1	1,966	0,0414	1,992
	2	1,971	-0,0422	1,992
	3	2,000	0,0230	1,991
	4	1,973	0,0273	1,991
	5	1,973	-2,4300	1,991
	6	1,929	-3,4330	1,991
	7	1,967	-0,00500	1,991
	8	2,000	-1,2450	1,992
	9	1,968	0,0041	1,991
	10	2,010	1,1410	1,991

Tabla 4.16: Medida de tensión media de señal periódica con Offset 1V

Podemos concluir por tanto diciendo que al realizar medidas manuales y mediante la solución propuesta sobre señales periódicas sin offset de tensión, es muy difícil obtener un valor medio medido de 0V, ya que los circuitos y equipos presentan ruidos y perturbaciones que afectan, aunque sea levemente, a los resultados leídos por los equipos. En cambio, cuando el valor a medir no es el 0 teórico, la solución propuesta ofrece unos resultados muy próximos a los tomados en el laboratorio manual, por lo que podemos considerar que el método desarrollado permite realizar medidas del valor medio de tensión de señales periódicas con valores más cercanos a los obtenidos en un laboratorio manual que los ofrecidos en el laboratorio VISIR, el cual en todos los experimentos realizados muestra al usuario medidas erróneas.

4.3 Conclusiones de los resultados obtenidos

Si volvemos sobre los resultados adquiridos en las pruebas que hemos llevado a cabo durante estas fases de experimentación y validación de la solución propuesta, llegamos a las siguientes conclusiones:

- El método de control propuesto es totalmente independiente de los instrumentos y circuitos bajo prueba que pueden formar parte de los experimentos de un laboratorio remoto.

- Un laboratorio que emplee el método presentado puede ser utilizado por usuarios reales que lleven a cabo experimentos reales en los que sea necesaria la construcción y medida remota de circuitos electrónicos analógicos.
- El conjunto de soluciones propuestas permiten a los usuarios remotos obtener medidas válidas de los circuitos construidos. Los resultados obtenidos no muestran diferencias reseñables significativas respecto a las obtenidas en un laboratorio manual o mediante otro laboratorio remoto de referencia como el VISIR, por lo que podemos confirmar esta afirmación.
- Por tanto, el método de control, y en particular el sistema de control de experimentos formado por la matriz de conmutación y los módulos de expansión diseñados y desarrollados, no introduce errores significativos durante la adquisición de medidas en comparación con los valores obtenidos mediante un laboratorio remoto de referencia y las medidas realizadas de manera local, por lo que el sistema es fiable y válido.
- Se ha comprobado que la velocidad de respuesta del sistema, aún siendo prácticamente el doble que la del sistema empleado en el laboratorio VISIR, es suficiente proporcionar una buena calidad de servicio y que el usuario no pierda interés en la aplicación y siga realizando experimentos. Aún así, debemos recordar que estas medidas se han tomado forzando al sistema a dar respuesta a 60 peticiones enviadas simultáneamente, circunstancia que es difícil de reproducir durante el uso normal de un laboratorio remoto.
- Hemos verificado cómo mediante el método descrito el rendimiento del laboratorio respecto al número de componentes necesarios para llevar a cabo circuitos electrónicos es mejor que el presentado en el laboratorio VISIR, ya que para un conjunto de circuitos que emplean los mismos componentes, se requiere la mitad para realizar todas las configuraciones posibles. Es decir, se ha mejorado el aprovechamiento de las matrices de conmutación empleadas, ya que un mismo componente se conecta dinámicamente a los nodos de un circuito, por lo que no es necesario duplicar componentes en la matriz, y se elimina

la necesidad de incluir cortocircuitos como un elemento más conectado en la matriz de conmutación.

- Finalmente hemos comprobado que la solución aportada para poder realizar medidas de valor medio de tensión sobre señales periódicas, es válida y supone una mejora sobre el sistema desarrollado en el laboratorio VISIR, que hasta el momento ofrece medidas erróneas en este escenario de evaluación.

En base a estas conclusiones podemos afirmar que el conjunto de desarrollos y tecnologías empleadas en la realización y despliegue del método de control que proponemos en este trabajo de investigación son válidos para llevar a cabo tareas de configuración, control y medida de los experimentos incluidos en un laboratorio remoto de electrónica analógica.

Capítulo**5****Conclusiones y líneas futuras de trabajo**

Las conclusiones alcanzadas tras finalizar el trabajo desarrollado y presentado en este documento pueden ser establecidas desde diferentes perspectivas, partiendo para ello de la hipótesis que planteábamos en el primer capítulo y que recogemos aquí de nuevo:

Es posible definir una metodología de control para ser desplegada en laboratorios remotos de electrónica analógica, que con independencia de los instrumentos y experimentos disponibles proporcione herramientas para la configuración de circuitos y la adquisición de medidas, obteniendo resultados lo más cercanos posibles a los adquiridos en un laboratorio manual.

Revisados y evaluados los resultados obtenidos mediante los diferentes ensayos y pruebas llevadas a cabo en el capítulo anterior, podemos considerar validada la hipótesis enunciada, ya que:

- Como resultado de nuestra investigación y desarrollos se ha obtenido un método de control independiente de los instrumentos, de manera que se pueden reemplazar y añadir instrumentos al laboratorio sin tener que realizar ninguna modificación sobre el software de control ya codificado.

- Este método permite a los usuarios configurar, controlar y testear remotamente experimentos por lotes de electrónica analógica independientemente de su tipo obteniendo unos resultados y valores en las medidas totalmente válidos y muy próximos a los obtenidos sobre los mismos experimentos de manera manual en un laboratorio local.
- El método de control diseñado y los desarrollos implementados en base a él pueden ser desplegados en un laboratorio remoto orientado a la electrónica analógica, en el cual todos los instrumentos necesarios para la configuración, control y medida remota de los experimentos están basados en plataformas comerciales, eliminando la necesidad de disponer complejas soluciones propietarias.

En cuanto a los objetivos más específicos planteados junto con la hipótesis que ha marcado este trabajo podemos reseñar las siguientes aportaciones:

- Se ha obtenido una aplicación de control de experimentos que puede ser utilizada con instrumentos de cualquier fabricante, cumpliendo únicamente un requisito: que los equipos puedan ser controlados mediante drivers IVI.
- Se ha validado que LXI puede ser utilizado como interfaz de control entre la aplicación software y los instrumentos del laboratorio. Utilizar esta tecnología permite que el despliegue de la red de instrumentos en el laboratorio sea sencilla, rápida y fácilmente ampliable desde el punto de vista de número de equipos a incorporar a los experimentos.
- Aún siendo LXI el interfaz de comunicaciones seleccionado como parte del método propuesto, la aplicación software de control de experimentos es totalmente transparente al interfaz de control, por lo que puede ser desplegada en un laboratorio remoto en el que los equipos se comuniquen con ella mediante cualquier otra tecnología de comunicaciones.
- Se ha implementado y validado un sistema de control de experimentos basado fundamentalmente en una matriz de conmutación comercial a la cual únicamente hemos necesitado incorporar unos módulos de expansión diseñados para aumentar su

capacidad desde el punto de vista de la cantidad de componentes y conexiones a utilizar en los circuitos bajo prueba.

- Gracias a la versatilidad y funcionalidades del laboratorio remoto WebLab-Deusto, se ha incorporado a su estructura el método de control propuesto y el cliente web proporcionado por el proyecto VISIR. De este modo hemos comprobado cómo el conjunto de soluciones aportadas por este trabajo pueden ser desplegadas en un laboratorio remoto y puede ser utilizado por usuarios reales, obteniendo resultados y medidas válidas, las cuales son muy cercanas a los obtenidos en un laboratorio manual.

La parte experimental de este trabajo doctoral ha estado basada en el despliegue del conjunto de desarrollos que hemos implementado en un escenario real de aplicación. Este escenario ha consistido en un prototipo de laboratorio remoto en el cual hemos desplegado la aplicación software de control, el interfaz de comunicaciones y el sistema de control de experimentos desarrollados durante este trabajo.

Posteriormente y con el objetivo de validar tanto la hipótesis como los objetivos planteados al comienzo de este trabajo, así como para satisfacer las motivaciones que han dado pie al mismo, se ha evaluado la respuesta de este escenario con usuarios reales, los cuales han realizado actividades de configuración, control y medida sobre los experimentos reales contenidos en el prototipo de laboratorio empleado durante la fase de experimentación. Los resultados obtenidos nos han permitido confirmar que el método de control es válido pues permite remotamente crear circuitos y tomar medidas sobre ellos sin introducir errores significativos en las mismas.

En cuanto a las aportaciones ofrecidas por el presente trabajo en relación a la posibilidad de obtener un método de control independiente de los instrumentos y los experimentos contenidos en un laboratorio remoto, podemos apuntar las siguientes ideas:

- El método de control propuesto está basado en los desarrollos y tecnologías utilizadas en la aplicación de control, el interfaz de comunicaciones y el sistema de control de experimentos para ser desplegado total o parcialmente en un laboratorio remoto de electrónica analógica. Como hemos visto, la aplicación software es

independiente de los instrumentos, de los experimentos e incluso de la tecnología empleada en el interfaz de comunicaciones, por lo que se podrán utilizar otras herramientas que satisfagan otras especificaciones en función de las necesidades del laboratorio remoto en el cual quiera ser incluido esta metodología.

- Hemos eliminado la necesidad de utilizar soluciones propietarias para poder construir y medir circuitos electrónicos mediante un laboratorio remoto. Los módulos de expansión diseñados y utilizados como parte del método propuesto, pueden ser sustituidos por otro mecanismo que permita conexionar las matrices de conmutación de otro modo. De este modo si mantenemos la estructura de configuración utilizada para describir la situación de los componentes en la matriz resultante, la aplicación de control sigue siendo totalmente válida y no es necesario realizar ninguna recodificación.

Como se ha indicado a lo largo de este trabajo, la metodología de control empleada en la plataforma VISIR ha sido una de las referencias más importantes tenidas en cuenta para el desarrollo de la solución propuesta. Como resultado se han obtenido las siguientes mejoras e innovaciones respecto a dicha plataforma:

- Se han simplificado las tareas de añadir nuevos componentes a los experimentos al eliminar los cableados requeridos en la matriz del sistema VISIR. Se ha aumentado también la capacidad del sistema respecto al número de componentes necesarios para llevar a cabo circuitos en los que intervengan los mismos componentes, así como la necesidad de incorporar cortocircuitos como componentes en la matriz de conmutación, aumentando también los puntos de medida de intensidad en los circuitos bajo prueba.
- Se ha solucionado el problema que aparece en el laboratorio VISIR al tratar de realizar mediante el multímetro medidas de tensión sobre señales periódicas, obteniendo resultados válidos que son comparables con los adquiridos sobre el mismo experimento en un escenario manual.

5.1 Publicaciones realizadas durante el trabajo de investigación

Durante el proceso de investigación se ha tratado de recoger información y opiniones de los investigadores y grupos relevantes en nuestra área de trabajo, así como de llevar a cabo acciones de disseminación de nuestras actividades y resultados en diferentes foros y congresos.

Para ello hemos acudido a conferencias para presentar los avances en la implementación del método de control que hemos diseñado y obtener así impresiones sobre el mismo por parte de la comunidad investigadora. Un hito fundamental en este proceso, fue el *workshop* organizado por el equipo del WebLab-Deusto y realizado en la Universidad de Deusto el año 2006 con los mayores expertos en experimentación remota de Europa. Durante estas reuniones tuvimos la oportunidad de conocer al coordinador del proyecto VISIR, el Profesor Ingvar Gustavsson, quien nos invitó a participar en dicho proyecto. Fruto de esta estrecha colaboración surgieron posteriormente las ideas y motivaciones para la búsqueda de nuevas soluciones relacionadas con el control de los instrumentos y experimentos en laboratorios remotos de electrónica analógica. En concreto, como resultado de posteriores reuniones de trabajo surgió la idea de incorporar las tecnologías IVI y LXI a un laboratorio remoto, y comprobar así sus bondades y diferencias respecto a los métodos empleados en otros laboratorios en general y respecto al VISIR en particular.

El conjunto de publicaciones realizadas desde la incorporación del autor de esta tesis doctoral al grupo de investigación WebLab-Deusto muestra la evolución de este trabajo desde que comenzamos a trabajar en el diseño y desarrollo de laboratorios remotos, partiendo de un primer laboratorio basado en el control de instrumentos mediante GPIB, posteriormente nuestra colaboración en el proyecto VISIR, y finalmente el desarrollo de una nueva solución basada en las tecnologías IVI y LXI:

- Hernández-Jayo, U., García-Zubía, J. "Instruments and Experiments Control Methodology Based on IVI and LXI Technologies", in *International Journal of Online Engineering (iJOE)*, Vol.7, pp. 15-21, 2011
- Hernández-Jayo, U., García-Zubía, J. "A Remote and Reconfigurable Analog Electronics Laboratory based on IVI an LXI Technologies" in

Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV), pp. 71 - 77, Brasov, Rumania, 29 June - 2 July 2011

- García-Zubía, J., Gustavsson, I., Hernández-Jayo, U., Orduña, P., Angulo, I., Ruiz de Garibay, J. "El proyecto VISIR en la Universidad de Deusto: laboratorio remoto para electrónica básica" *actas del IX Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica (TAEE)* Madrid, Abril 2010.
- García Zubia, J., Hernández, U., Angulo, I., Orduña, P., Irurzun, J. "LXI technologies for remote labs: an extension of the VISIR project", *in Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV)*, Stockholm, Sweeden, June 25 - 27, 2010.
- Gustavsson, I., Nilsson, K., Zackrisson, J., Garcia-Zubia, J., Hernández-Jayo, U., Nafalski, A., Nedic, Z., Gol, O., Machotka, J., Pettersson, M.I., Lago, T., Hkansson, L. "On Objectives of Instructional Laboratories, Individual Assessment, and Use of Collaborative Remote Laboratories" *IEEE Transactions on Learning Technologies*, Vol 2, Issue 4, pp. 263-274. 2009
- García Zubia, J., Hernández, U., Angulo, I., Orduña, P. "Remote Laboratories based on LXI" *International Journal of Online Engineering (iJOE)*. Vol. 4. No.3. ISSN: 1861-2121. 2008
- García-Zubía, J., López de Ipiña, D., Hernández, U., Orduña, P., Trueba, I. "WebLab-GPIB at University of Deusto" *in Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV)*, Oporto (Portugal). Jun. 2007.

5.2 Líneas futuras de trabajo

El trabajo presentado en este documento plantea una serie de alternativas a los problemas previamente descritos y relacionados con el control del hardware de un laboratorio remoto de electrónica analógica. En este contexto aún quedan soluciones que aportar y mejoras que realizar respecto a los desarrollos realizados, entre las cuales destacamos las siguientes posibles líneas futuras de investigación:

- Mejorar los tiempos de respuesta del sistema, ya que tal y cómo hemos podido comprobar durante la fase de pruebas, al aumentar el número de alumnos conectados, el rendimiento del sistema respecto a la velocidad de respuesta aún queda alejado respecto al que ofrecen otras alternativas, como por ejemplo el laboratorio VISIR. Para conseguir este objetivo se pueden plantear diferentes líneas de trabajo encaminadas bien a incrementar la velocidad de ejecución de los algoritmos de control, o bien a buscar el método que permita realizar una sincronización óptima del sistema de medida, haciendo uso para ello por ejemplo de equipos LXI clase B o A.
- Aun habiendo diseñado unos módulos de expansión sencillos que permiten obtener la configuración de matriz de conmutación descrita en este trabajo y que posibilita la realización de gran cantidad de circuitos y configuraciones, se puede mejorar el rendimiento de esta matriz resultante respecto a la capacidad de componentes disponibles en los experimentos. Para ello se pueden estudiar diferentes alternativas de configuración, conexionado y control que exploten la característica de doble capa que ofrecen las matrices de conmutación proporcionadas por los fabricantes.

Referencias bibliográficas

- [AA09] Aydogmus, Z., Aydogmus, O. "A Web-Based Remote Access Laboratory Using SCADA", *IEEE Transactions on Education*, Vol. 52, Issue 1, pp. 126, ISSN: 0018-9359. Feb. 2009
- [AAJ12] Azad, A., Auer, M.E., Judson, V. "Internet Accessible Remote Laboratories: Scalable E-Learning Tools for Engineering and Science Disciplines", Ed. IGI Global, ISBN13: 978-1613501-863, 2012
- [AB96] Aktan, B., Bohus, C.A. "Distance Learning Applied to Control Engineering Laboratories", *IEEE Transactions on Education*, Vol 39, No.3, pp. 320-326. 1996
- [ABC+00] Arpaia, P., Baccigalupi, A., Cennamo, F., Daponte, P. "A measurement laboratory on geographic network for remote test experiments," *IEEE Transaction on Instrumentation and Measurement*, Vol. 49, No. 5, pp. 992-996, Oct. 2000.
- [ABM+02] del Alamo, J. A., L. Brooks, C. McLean, J. Hardison, G. Mishuris, V. Chang, and L. Hui, "The MIT Microelectronics WebLab: a Web-Enabled Remote Laboratory for Microelectronic Device Characterization", *World Congress on Networked Learning in a Global Environment*, Berlin (Germany), 2002.
- [ACH+03] del Alamo, J.A., Chang, V., Hardison, J., Zych, D., Hui, L. "An online microelectronics device characterization laboratory with a circuit-like user interface" in *Proceedings of International Conference Engineering Education (ICEE)*, Valencia, Spain, Jul. 2003.
- [AHG+06] Åkesson, H., Håkansson, L., Gustavsson, I., Zackrisson, J., Claesson, I., Lagö, T. "Vibration Analysis of Mechanical Structures over the Internet Integrated into Engineering Education", *Proceedings of the 2006 ASEE Annual Conference*, Chicago, USA, June 18 - 21, 2006.
- [AHM+02] del Alamo, J.A., Hardison, J., Mishuris, G., Brooks, L., McLean, C., Chan, V., Hui, L. "Educational experiments with an online microelectronics characterization laboratory" in *Proceedings of the International Conference of Engineering Education*, Manchester, U.K., 2002

- [Ala⁺03] del Alamo, J.A.d., et al., "MIT Microelectronics WebLab". In *Lab on the Web*. pp. 49-87. 2003
- [Ala04] del Alamo, J. A. "iLabs: Performing Laboratory Experiments Across Continents," in *Learning International Networks Consortium Workshop(LINC)*, MIT, March 24-25, 2004
- [Alv11] Alves, G.R. "Los laboratorios remotos: experimentación a distancia", *Jornadas Educar en Tecnología de la Cátedra Telefónica -Deusto*, 27 de Octubre 2011
- [AMM91] Aburdene, M.F., Mastascusa, E.J., Massengale, R. "A proposal for a remotely shared control systems laboratory", In *Proceedings of the American Society for Engineering Education Annual Conference (ASEE)*, pp. 589-592, 1991.
- [AO04] Asumadu, J.A., Ogunleye, H. "Translation of an Efficient Web-based 3-D Matrix Switching Algorithms for Remote Switching and Control of Electrical and Electronics Instruments," *Proceedings of the 21st IEEE Instrumentation and Technology Conference* .Vol. 3, pp. 1776 - 1779, May 18 - 20, Italy 2004
- [AS02] Amaratunga. K., Sudarshan, R. "A virtual laboratory for real-time monitoring of civil engineering infrastructure",in *Proceedings of International Conference Engineering Education (ICEE)*, Manchester, U.K., Aug. 18-22, 2002.
- [AT01] Asumadu, J.A., Tanner, R., "Remote wiring and measurement lab". In *Proceedings of American Society Engineering Education Annual Conference Albuquerque, NM*. June 24-27, 2001
- [ATF⁺05] Asumadu,J.A., Tanner, R., Fitzmaurice, J., Kelly, M., Ogunleye, H., Belter J., Song Chin Koh, "A Web-Based Electrical and Electronics Remote Wiring and Measurement Laboratory (RwmLAB) Instrument," *IEEE Transaction on Instrumentation and Measurement*, Vol. 54, No. 1, pp. 8-44, February 2005, doi:10.1109/TIM.2004.834597.
- [BCP⁺00] Bagnasco, A., Chirico, M., Parodi G., Scapolla, A.M. "Virtual Laboratory for Remote Engineering Education" in *International Perspective on Tele-education and Virtual Learning Environment*, Ashgate Book, pp 1-14, ISBN 0-7546-1202-3, 2000
- [BCS⁺03] Bagnasco, A., Chirico, M., Scapolla, A.M., Parodi, G. "A model for an open and flexible training platform to encourage companies' learning culture and meet employees' learning needs", *Educational technology & Society* 6(1), ISSN 1436-4522, 2003

- [BCS05a] Bagnasco, A., Chirico, M., Scapolla, A.M. "A New and Open Model to Share Laboratories in the Internet", *IEEE Transaction on Instrumentation and Measurement*, Vol. 54, pp 1111-1117. ISSN 0018-9456, June 2005.
- [BCS05b] Bagnasco, A., Chirico, M., Scapolla, A.M. "XML Technologies to Design Didactical Distributed Measurement Laboratories". *IEEE Transactions on Instrumentation and Measurement*, Vol. 54, No.1, Febrero 2005
- [BI08] Basher, H.A. and Isa, S.A. "Laboratory experiments using NI ELVIS". *Proceedings of IEEE SOUTHEASTCON 2008*. Vol. 1-2, pp. 378-383. Huntsville, USA. Abril 03-06, 2008
- [BKB00] Bouch, A., Kuchinsky, A., and Bhatti, N. "Quality is in the Eye of the Beholder: Meeting Users' Requirements for Internet Quality of Service," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 297-304, The Hague, The Netherlands, April 1-6, 2000.
- [BM06] Brezovec, B., Matko, V., "Remote measurement of temperature and humidity in climate chamber with LabVIEW software" in *Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV)*, Brasov (Rumanía), Julio 2006
- [BPP+07] Bagnasco, A., Poggi, A., Parodi, G., Scapolla, A.M. "A Configurable Remote Laboratory for the Flexible Setup of Experiments in Electronics", in *Advances on remote laboratories and e-learning experiences*, Ed. by L. Gomes and J. Garcia Zubia, University of Deusto, Bilbao, 2007, pp. 271-283. ISBN: 978-84-9830-077-2
- [Bro09] Browne, J. "Networking RF Tests With LXI", *MICROWAVES & RF journal*, Vol. 48, Issue 5, pp.54-55, ISSN:0745-2993, May 2009
- [BS09] Buschiazzo, P., Scapolla, A. M. "A service oriented framework to set up flexible and distributed data acquisition and control systems", *GRIT Workshop @ ACM MEDES 2009*, October 27-30, 2009, Lyon, France
- [BSV+03] Bagnasco, A., Scapolla, A.M., Vassileva, T., Vasilev, V., Furnadziej, I. "Authoring and Automtic Generation of Virtual Panel for Controlling Instrumentation in a Remote Laboratory Conext". *Proceedings of 2nd International Conferenceon Multimedia ICT's in Education - m-ICTE*. ISBN 84-96212-09-02. Badajoz, España. Diciembre 2003

- [CAM+11] Costa Lobo, M.C., Alves, G.R., Marques, M.A., et al "Using Remote Experimentation in a Large Undergraduate Course: Initial Findings", in *Proceedings of 41st ASEE/IEEE Frontiers in Education Conference*, pp. S4G1-S4G7, Rapid City (USA), October 12-15, 2011
- [CAP05] Coito, F., Almeida, P., Palma, L.B. "SMCRVI - a LabVIEW/Matlab based tool for remote monitoring and control", in *Proceedings of 10th IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1084-1090, Sept. 2005
- [CCA02] Cooper, M., Colwell, C., Amaral, T. "Accessibility and usability in complex web based learning applications: Lessons from the PEARL project," in *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, pp. 1358-1365. 2002
- [CDF02] Cooper, M., Donnelly, A., Ferreira, J.M. "Remote controlled experiments for teaching over the internet: a comparison of approaches developed in the PEARL project". In *proceedings of 19th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE)*. pp 119-128. Auckland, New Zealand 2002
- [CDK01] Coulouris, G., Dollimore, J., Kindberg, T. "Distributed systems concepts and design", *Pearson Education Limited editor*, 3rd Edition, 2001
- [CJB+06] Chang, T., Jaroonsiriphan, P., Bernhardt, M., Ludden, P. "Web-based command shaping of cobra 600 robot with a swinging load" *IEEE Transactions on Industrial Informatics*, Vol. 2, Issue 1, pp. 59-69, ISSN: 1551-3203, Feb. 2006
- [CNE+04] Corter, J.E., Nickerson, J.V., Esche, S.K., Chassapis, C. "Remote versus hands-on labs: a comparative study" in *Proceedings of 34th Annual Frontiers in Education Conference*, F1G - 17-21 Vol. 2, ISSN: 0190-5848, 2004
- [CPA+00] Choy, G., Parker, D.R., d'Amour, J.N., Spencer, J.L. "Remote experimentation: a Web-operable two-phase flow experiment" in *Proceeding of the 2000 American Control Conference*, Vol. 4, pp. 2939-2943. ISBN: 0-7803-5519-9. Chicago, USA. Jun. 2000
- [CPV03] Casini, M., Prattichizzo, D., Vicino, A. "The automatic control telelab: a user-friendly interface for distance learning", *IEEE Transactions on in Education*, Vol. 46, Issue 2, pp. 252. ISSN: 0018-9359. May 2003

- [CSS+06] Cedazo, R., Sánchez, F.M., Sebastián, J.M., Martínez, A., Pinazo, A., Barros, B., Read, T. "Ciclope chemical: a remote laboratory to control a spectrograph", in *Proceedings of Advances in Control Education Conference (ACE06)*, Madrid, España, Jun. 2006
- [Dor02] Dormido, S. "Control learning: present and future". In *Proceedings of 15th World Conference of the International Federation of Automatic Control (IFAC'02)*. Plenary Lecture. Barcelona 2002
- [DP07] Donzellini, G., Ponta, D. " The electronic laboratory: traditional, simulated or remote?", in *Advances on remote laboratories and e-learning experiences*, Ed. by L. Gomes and J. Garcia Zubia, University of Deusto, Bilbao, 2007, pp. 223-245. ISBN: 978-84-9830-077-2
- [DSM00] Dormido, S., Sánchez, J., Morilla, F. "Laboratorios virtuales y remotos para la práctica a distancia de la Automática", *Proceedings del Congreso Online Educa Madrid*, Madrid, Junio 2000
- [ED11] El Helou, S., Gillet, D. "The Four Elements of a viable Personal Learning Enviroments", *1st Workshop on Exploring the Fitness and Evolvability of Personal Learning Environments (EFEPLE'11)*, La Clusaz, France, March 27-31,2011.
- [Edu05] Educación universitaria. Nuevas carreras más universidades. *Boletín informativo del Instituto Nacional de Estadística*. No. 4 Año 2005
- [ESF+03] Eikaas, T. I.; Schmid, C.; Foss, B. A.; Gillet, D. "A global remote laboratory experimentation network and the experiment service provider business model and plans". In *Journal of Modeling, Identification and Control*. Vol. 24, No. 3, pp. 159-168 ISSN 1890-1328. 2003
- [Est08] Estadística de la Enseñanza Universitaria en España. Curso 2006/2007. *Notas de prensa del Instituto Nacional de Estadística*. 23 de Mayo 2008. Disponible en www.ine.es/prensa/prensa.htm
- [Est10] Estadística de la Enseñanza Universitaria en España. Curso 2009/2010. Disponibles en <http://www.ine.es>
- [Exp11] Experimentos iLab [online, cited 1 de Marzo de 2011]. URL: <https://wikis.mit.edu/confluence/display/ILAB2/iLabs>.

- [Fei02] Feisel, L.D. et al., "Learning objectives for engineering education laboratories," presented at the Frontiers in Education (FIE) 32nd Annual Conference, 2002.
- [FGW+01] Feng, H., Gengquan, W., Wei, L., Xiaolin, H., Jianhua, L. "Object request brokers for distributed measurement" *IEEE Computer Applications in Power*, Vol.14, Issue 1, pp. 50-54, ISSN: 0895-0156, Jan. 2001
- [GA11] García-Zubía, J., Alves, G.R "Using remote labs in Education", Ed. Universidad de Deusto. ISBN: 978-84-9830-335-3, 2011
- [GAH08] García-Zubia, J., Angulo, I., Hernandez, U., Orduña, P. "Plug&Play Remote Lab for Microcontrollers: WebLab-DEUSTO-PIC" in *Proceedings of 7th European Workshop on Microelectronics Education*, May 28-30, 2008, BME - Budapest, Hungary.
- [GAO+10] García-Zubia, J., Angulo, I., Orduña, P., Hernández, U., Irurzun, J., Ruiz, J., Castro, M., Sancristobal, E. "Easily Integrable platform for the deployment of a Remote Laboratory for microcontrollers" in *Proceedings of IEEE Education Engineering Conference (EDUCON)*, pp. 327, ISBN: 978-1-4244-6568-2, Abril 2010
- [GB09] Gomes, L., Bogosyan, S. "Current Trends in Remote Laboratories", *IEEE Transactions on Industrial Electronics*. Vo. 56, Issue 12, pp. 4744-4756. ISSN: 0278-0046. Diciembre 2009
- [GCC+07] Gomes, L., Coito, F., Costa, A., Brito, L. "Teaching, learning and remote laboratories", in *Advances on remote laboratories and e-learning experiences*, Ed. by L. Gomes and J. Garcia Zubia, University of Deusto, Bilbao, 2007, pp. 180-204. ISBN: 978-84-9830-077-2
- [GFB+08] Gravier, C., Fayolle, J., Bayard, B., Ates, M. and Lardon, J. "State of the Art About Remote Laboratories Paradigms—Foundations of Ongoing Mutations," *International Journal of Online Engineering (iJOE)*. Vol. 4, No. 1, pp. 19-25, 2008.
- [GGH+10] García-Zubía, J., Gustavsson, I., Hernández-Jayo, U., Orduña, P., Angulo, I., Ruiz de Garibay, J. "El proyecto VISIR en la Universidad de Deusto: laboratorio remoto para electrónica básica" *actas del IX Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica (TAEE)* ISBN:978-84-96737-67-9 Madrid, Abril 2010.

- [GHA+08] García Zubia, J., Hernández, U., Angulo, I., Orduña, P. "Remote Laboratories based on LXI". *International Journal of Online Engineering (iJOE)*. Vol. 4. No.3. ISSN: 1861-2121. 2008
- [GHA+10] García Zubia, J., Hernández, U., Angulo, I., Orduña, P., Irurzun, J. "LXI technologies for remote labs: an extension of the VISIR project", in *Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV)*, Stockholm, Sweeden, June 25 - 27, 2010.
- [Gik06] Gikandi, S. "ELVIS iLab: a flexible platform for online laboratory experiments in electrical engineering", Thesis (M. Eng.) Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 2006
- [GIO+10] Garcia-Zubia, J., Irurzun, J., Orduna, P., Angulo, I., Hernández, U., Ruiz, J., Castro, C., San-Cristobal, E. "SecondLab: A Remote Laboratory under Second Life" *International Journal of Online Engineering (iJOE)*, Vol.6, No. 4. ISSN: 1861-2121. 2010.
- [GLH+07a] García-Zubía, J., López de Ipiña, D., Hernández, U., Orduña, P., Trueba, I. "WebLab-GPIB at University of Deusto" in *Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV)*, Oporto (Portugal). Jun. 2007.
- [GLH+07b] García-Zubía, J., López de Ipiña, D., Hernández, U., Orduña, P., Trueba, I. "An Approach for WebLabs Analysis", *International Journal of Online Engineering (iJOE)*, Vol.3, No. 2. ISSN: 1861-2121. 2007
- [GLO+06a] García-Zubia, J., Lopez-de-Ipiña, D., Orduña, P., Hernández-Jayo, U., Trueba, I. "Questions and Answers for Designing Useful WebLabs" *International Journal of Online Engineering (iJOE)*. Vol. 2, No. 3. ISSN: 1861-2121. 2006
- [GLO+06b] García-Zubia, J., López-de-Ipiña, D., Orduña, P., Hernández, U., Trueba, I. "Evolution of the WebLab at the University of Deusto" in *Proceedings of 6th European Workshop on Microelectronics Education (EWME)*, Stockholm (Sweden). Jun. 2006.
- [GLO05b] García-Zubia, J., López-de-Ipiña, D., Orduña, P. "Towards a Canonical Software Architecture for Multi-Device WebLabs" in *Proceedings of 31st Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pp. 2146-2151, ISBN: 0-7803-9253, November 2005

- [GLO08] García-Zubía, J., López de Ipiña, D., Orduña, P. "Mobile devices and remote labs in engineering education" *Proceedings of 8th IEEE International Conference on Advanced Learning Technologies (ICALT)*. pp.620-622, ISBN: 978-0-7695-3167-0, Santander, 2008
- [GM05] Garcia-Zubia, J., del Moral, A. "Suitability and implementation of a WebLab in engineering" in *Proceedings of 10th IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 56-64, Catania, Sept. 2005
- [GMP+03] Guimaraes, E., Maffeis, A., Pereira, J., Russo, B., Cardozo, E., Bergerman, M., Magalhaes, M.F. "REAL: a virtual laboratory for mobile robot experiments" *IEEE Transactions on Education*, Vol. 46, Issue 1, ISSN: 0018-9359, Feb. 2003
- [GNR05] Gillet, D., Nguyen Ngoc, A.V., Rezik, Y. "Collaborative web-based experimentation in flexible engineering education" *IEEE Transactions on Education*, Vol. 48, Issue. 4. pp. 696-704. ISSN: 0018-9359. 2005
- [GNZ+09] Gustavsson, I., Nilsson, K., Zackrisson, J., Garcia-Zubia, J., Hernández-Jayo, U., Nafalski, A., Nedic, Z., Gol, O., Machotka, J., Pettersson, M.I., Lago, T., Håkansson, L. "On Objectives of Instructional Laboratories, Individual Assessment, and Use of Collaborative Remote Laboratories" *IEEE Transactions on Learning Technologies*, Vol 2, Issue 4, pp. 263-274. 2009
- [GOA+05] Gustavsson, I., Olsson, T., Åkesson, H., Zackrisson, J., Håkansson, L. "A Remote Electronics Laboratory for Physical Experiments using Virtual Breadboards", *Proceedings of the 2005 ASEE Annual Conference*, Portland, USA, June 12 - 15, 2005.
- [GOA08] García-Zubia, J., Orduña, P., Angulo, I., Irurzun, J., Hernández, U. "Towards a Distributed Architecture for Remote Laboratories" *International Journal of Online Engineering (iJOE)*, Vol.4, Special Issue REV 2008, pp. 11-14, 2008
- [GOH+09] García-Zubia, J., Orduña, P., Hernández-Jayo, U., Angulo, I., Irurzun, J. "Students' review of acceptance, usability and usefulness of WebLab-Deusto." *Journal of Digital Information Management (JDIM)*. Vol. 7, No. 3. ISSN: 0972-7272. 2009.
- [GOI+09a] García-Zubia, J., Orduña, P., Irurzun, J., Hernández, U., Sancristobal, E., Martín, S., Castro, M., López-de-Ipiña, D., Angulo, I. "Towards an Extensible WebLab Architecture", *3rd IEEE International Conference on E-Learning in Industrial Electronics (ICELIE)*, pp. 115-120, ISBN: 978-1-4244-4653-7, Porto (Portugal), Nov. 2009

- [GOL⁺07] García-Zubia, J., Orduña, P., López-de-Ipiña, D., Hernández, U., Trueba, I. "Remote laboratories from the software engineering point of view" in *Advances on remote laboratories and e-learning experiences*, Ed. by L. Gomes and J. Garcia Zubia, University of Deusto, Bilbao, 2007, pp. 133-149. ISBN: 978-84-9830-077-
- [GOL⁺09] Garcia-Zubia, J., Orduna, P., Lopez-de-Ipina,D., Alves, G.R. "Addressing Software Impact in the Design of Remote Laboratories" *IEEE Transactions on Industrial Electronics*, Vo. 56 Issue 12, pp. 4757-4767, ISSN: 0278-0046. Diciembre 2009
- [GPO⁺10] García-Zubía, J., Pester, A., Orduña, p., Irurzun, J., González, J.M., Angulo, I., Hernández, U., Rodríguez, L. "One Lesson from TARET: what is expected from a remote lab?" in *Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV)*, Stockholm, June 29 - July 2, 2010
- [GRB10] Rodriguez-Andina, J.J., Gomes, L., Bogosyan, S. "Current Trends in Industrial Electronics Education". *IEEE Transactions on Industrial Electronics*. Vo. 57, Issue 10, pp. 3245-3252. ISSN: 0278-0046. Julio 2010
- [GS08] Gercek, G., Saleem, N. "Transforming Traditional Labs into Virtual Computing Labs for Distance Education". *International Journal of Online Engineering (iJOE)*. Vol. 4. No. 1. ISSN: 1861-2121. 2008
- [Gus02] Gustavsson, I. "A Remote Laboratory for Electrical Experiments", *Proceedings of the 2002 ASEE Annual Conference*, Montreal, Canada, June 16 - 19, 2002.
- [Gus03a] Gustavsson, I. "A Traditional Electronics Laboratory with Internet Access", *Proceedings of the International Conference on Networked e-learning for European Universities*, Granada, Spain, November 23 - 25, 2003.
- [Gus03b] Gustavsson, I. "Remote Laboratory for Electrical Experiments" *Chapter 5 in Lab on the Web: Running Real Electronics Experiments via the Internet*, Edited by Tor A. Fjeldly, Michael S. Shur. October 2003, Wiley-IEEE Press. ISBN: 978-0-471-41375-2
- [Gus04] Gustavsson, I. "Traditional Laboratory Exercises by Remote Experimentations in Electrical Engineering Education" *Chapter 16 in INNOVATIONS 2004: World Innovations in Engineering Education and Research*, Edited by W. Aung et al, iNEER, Arlington (2004), ISBN 0-9741252-1-0

- [Gus06] Gustavsson, I. et al., "An Instructional Electronics Laboratory Opened for Remote Operation and Control", *Proceedings of the ICEE 2006 Conference*, San Juan, Puerto Rico, July 23 - 28, 2006.
- [GV05] Gobbo, F., Vaccari, M. "Open Standards for Higher Education in Robotics by Immersive Telelaboratories" *Learning Technology Newsletter*, Vol. 7, No. 3, pp.30-32, 2005
- [GZH*07] Gustavsson, I., Zackrisson, J., Håkansson, L., Claesson, I., Lagö, T.L., "The VISIR project - an Open Source Software Initiative for Distributed Online Laboratories", in *Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV)*, Porto, Portugal, June 25 - 27, 2007.
- [GZH07] Gustavsson, I., Zackrisson, J., Håkansson, L. "A new concept for distributed laboratories based on open source technologies", in *Advances on remote laboratories and e-learning experiences*, Ed. by L. Gomes and J. Garcia Zubia, University of Deusto, Bilbao, 2007, pp. 247-267. ISBN: 978-84-9830-077-2
- [GZO04] Gustavsson, I., Zackrisson, J., Olsson, T. "Traditional Lab Sessions in a Remote Laboratory for Circuit Analysis", *Proceedings of the 15th EAAEIE Annual Conference on Innovation in Education for Electrical and Information Engineering*, Sofia, Bulgaria, 27th - 29th May 2004.
- [HAE*07] Hine, N.A., Alves, G.R., Erbe, H.H., Müller, D., da Mota Alves, J.B., Pereira, C., Ferreira, J.M., Sucar, L.E., Herrera, O.A., Chiang, L., García-Zubia, J. "Institutional factors governing the deployment of remote experiments: Lessons from the REXNET project," in *Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV)*, Porto (Portugal), Julio 2007
- [HAL*08] Harward, V.J., del Alamo, J.A., Lerman, S.R., Bailey, P.H. et al. "The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories", *Proceedings of the IEEE*. Vol. 96, Issue 6, pp.931, ISSN: 0018-9219. June 2008
- [HDB*08] Hardison, J., DeLong, K., Bailey, P., Harward, V.J. "Deploying Interactive Remote Labs Using the iLab Shared Architecture" in *Proceedings of Frontiers in Education (FIE) Conference*, pp. S2A-1 S2A-6, October 2008
- [HG11] Hernández-Jayo, U., García-Zubía, J. "A Remote and Reconfigurable Analog Electronics Laboratory based on IVI an LXI Technologies" in *Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV)*, pp. 71 - 77, Brasov, Rumania, 29 June - 2 July 2011

- [IEE03] IEEE Std 488.1-2003, IEEE Standard for Higher Performance Protocol for the Standard Digital Interface for Programmable Instrumentation
- [IGR07] Indrusiak, L.S., Glesner, M., Reis, R. "On the evolution of remote laboratories for prototyping digital electronic systems," *IEEE Transactions on Industrial Electronics*, Vol. 54, No. 6, pp. 3069-3077, Dec. 2007.
- [IVI10a] IVI Specification, IVI-3.1: Driver Architecture Specification, IVI Foundation 2010
- [IVI10b] IVI Specification, IVI-4.2: IviDmm Class Specification, IVI Foundation 2010
- [JCT08] Jara, C.A., Candelas, F.A., Torres, F. "Robolab.ejs: a new tool for robotics e-learning", in *Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV)*, Düsseldorf, June 2008.
- [Joh02] Johnson, A. et al., "Nuts and volts: A Web-based hands-on electrical and electronics remote, wiring and measurement laboratory", *Proceedings of the 2002 ASEE Annual Conference*, Montreal, Canada, June 16 - 19, 2002.
- [LCC+10] Leiner, B.M., Cerf, V.G., Clark, D.D., Kahn, R.E. et al. "A brief History of the Internet". Available at the Internet Society web site <http://www.isoc.org/internet/history/brief.shtml> Consulted at 2010
- [LLI07] Lindsay, E., Long, P., and Imbrie, P.K. "Workshop – Remote Laboratories: Approaches for the Future," in *Proceedings of American Society for Engineering Education Annual Conference. (ASEE'07)*, pp. W1C-1-W1C-2, 2007.
- [LMP+08] Lerro, F., Marchisio, S., Plano, M., Protano, M., Von Pamel, O. "A remote lab like a didactic resource in the teaching of the physics of electronic devices" in *Proceedings of 11th International Conference on Interactive Computer aided Learning (ICL)*, Kassel University Press, 2008
- [LMV09] Lerro, F., Marchisio, S., Von Pamel, O. "Exploring didactic possibilities of an electronic devices remote lab with students of Electronic Engineering" in *Proceedings of International conference on Interactive Computer Aided Blended Learning (ICBL)*, Florianopolis, (Brasil), Nov. 2009
- [LP07] Lerro, F.G., Protano, M.D. "Web-based Remote Semiconductors Devices Testing Laboratory" *International Journal of Online Engineering (iJOE)*, Vol.3, No. 3. ISSN: 1861-2121. 2007.

- [LXI08] LXI Version 1.3 Specification, LXI Standard Revision, LXI Consortium 2008
- [Mat99] Matthews, D. "The Origins of Distance Education and Its Use in the United States," *THE J.*, http://www.thejournal.com/articles/14278_1, Sept. 1999.
- [MBG10] Markan, C.M., Bhatnagar, A., Gupta, A. "A Remote Triggered Electronics Laboratory based on LXI methodology". in *Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV)*, Stockholm, June 29 - July 2, 2010
- [ME07] Müller, D., Erbe, H.H. " Collaborative remote laboratories in engineering education: Challenges and visions", in *Advances on remote laboratories and e-learning experiences*, Ed. by L. Gomes and J. Garcia Zubia, University of Deusto, Bilbao, 2007, pp. 35-59. ISBN: 978-84-9830-077-2
- [MF03] Müller, D., Ferreira, J. "MARVEL: A Mixed Reality Learning Environment for Vocational Training in Mechatronics." *International Conference on Technology Enhanced Learning (TEL)*, Nov. 2003, Milano, Italy
- [MMN06] Machotka, J., Mohtar, A. and Nedic, Z. "Incorporation of remote laboratories into on-campus engineering courses". *Proceedings of UNESCO International Centre for Engineering Education UICEE 9th Annual Conference on Engineering Education*, Muscat, Oman, pp. 157-160. February 2006
- [MN02] Machotka, J and Nedic, Z. "Online Remote Laboratory (NetLab)". In *UICEE 5th Annual Conference On Engineering Education*. 2002. Chennai, India.
- [MN03] Machotka, J and Nedic, Z. "The Development of Remote Laboratory (NetLab) at the University of South Australia". In *Proc 7th Baltic Region Seminar on Engineering Education*. St Petersburg, Russia, 2003.
- [MN04] Machotka, J. and Nedic, Z. "Students' telepresence in a remote laboratory: a case study - NetLab", *Proceedings of the International Conference on Engineering Education and Research - Progress Through Partnership*, Farana, R. et al (Ed.). Bouzov Castle, Czech Republic, pp. 1237-1242. 2004
- [MN06] Ma, J., Nickerson, J.V. "Hands-On, Simulated, and Remote Laboratories: A Comparative Literature Review". *ACM Computing Surveys*, Vol. 38, No. 3, Article 7. *ACM Computing Surveys*, Vol. 38, No. 3, Article 7. September 2006
- [MN08] Machotka, J. and Nedic, Z. "From the Collaborative Environment of Remote Laboratories to the Global Collaboration", in *Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV)*, Oporto, Portugal. 2009

- [MNY04] Machotka, J., Nedic, Z. and Yordanova, S. "Collaboration in remote controlled laboratory", *Proceedings of 14th Scientific Symposium with International Participation- Metrology and Metrology Assurance*, Sozopol, Bulgaria, pp. 60-64. Septiembre 2004
- [MOO09] "Moodle Course Management System", *Moodle Community Website*, <http://moodle.org>. 2009
- [MV07] Matute, H., Vadillo, M.A. "Assesing e-learning in weblabs", in *Advances on remote laboratories and e-learning experiences*, Ed. by L. Gomes and J. Garcia Zubia, University of Deusto, Bilbao, 2007, pp. 97-107. ISBN: 978-84-9830-077-2
- [MVM+99] Moure, M.J., Valdes, M.D., Mandado, E., Salverria, A. "Educational application of virtual instruments based on reconfigurable logic." *Proceedings of IEEE International Conference on Microelectronic Systems Education, MSE '99*. pp. 24-25. Arlington,USA. Julio 1999
- [NAT06] National Instruments - NI Educational Laboratory Virtual Instrumentation Suite (NI ELVIS) - *Hardware User Manual*, [En línea], Issue 373363D-01, 2006. [Consult. 25 Oct. 2010]. Disponible en <<http://www.ni.com/pdf/manuals/373363d.pdf>>
- [NIE09] "NI-ELVIS: Educational Design and Prototyping Platform," *National Instruments Corp. Website*, <http://www.ni.com/nielvis>. 2009.
- [NIL10] "NI-LabVIEW Development System," *National Instruments Corp. Website*, <http://www.ni.com/LabVIEW>, 2010.
- [NM07] Nedic, Z and Machotka, J. "Remote Laboratory Netlab for Effective Teaching of 1st Year Engineering Students" *International Journal of Online Engineering (iJOE)*, Vol.3, No. 3. ISSN: 1861-2121. 2007.
- [NMS+05] Nedic, Z., Machotka, J., Sprok, A., Ruud, L. O., Carr, S. "The Circuit Builder for NetLab". *8th UICEE Annual Conference on Engineering Education*. Kingston, Jamaica, 7-11 February 2005.
- [NZP08] Nilsson, K., Zackrisson, J. and Pettersson, M. "Remote Access of Computer Controlled Experiments", *International Journal of Online Engineering (iJOE)*, Vol. 4, No. 4, 2008.

- [OAD*07] Oliver, J., Abaitua, J., Díaz, J., Jacob, I., Buján, D., Garaizar, P., Lázaro, L. "Issues in WebLab development: security, accessibility, collaboration and multilinguality." in *Advances on remote laboratories and e-learning experiences*, Ed. by L. Gomes and J. Garcia Zubia, University of Deusto, Bilbao, 2007, pp. 111-129. ISBN: 978-84-9830-077-2
- [OGL*11] Orduña, P., García-Zubía, J., López de Ipiña, D., Irurzun, J. "Accessing Remote Laboratories from Mobile Devices", Ed. IGI Global, ISBN13: 978-1609606-138, 2011
- [PA07] Pester, A., Alves, G.R. "WebLabs in Education". *Guest Editorial in the International Journal of Online Engineering (iJOE)*.Vol.3, No. 2. ISSN: 1861-2121. 2007
- [Pau05] Paulson, L.D. "Building Rich Web Applications with Ajax" *Journal Computer*, Vol. 38, No. 10, pp.14-17, Oct. 2005
- [PXI05] PXI Express Hardware Specification Revision 0.5, PXI Systems Alliance, July 15 2005
- [REV09] REV 2009, Proceedings of the conference Remote Engineering & Virtual Instrumentation Conference, Bridgeport, USA 2009
- [REV10] REV 2010, Proceedings of the conference Remote Engineering & Virtual Instrumentation Conference, Estocolmo, Suecia 2010
- [Ros01] Rosenberg, M.J. "E-Learning: Strategies for Delivering Knowledge in the Digital Age". McGraw Hill, 2001. ISBN: 0-07-136268-1
- [RZH*12] Rojko, A., Zürcher, T., Hercog, D., Stebler, R. "Implementation of Remote Laboratories for Industrial Education", Ed. IGI Global, ISBN13: 978-1613501-863, 2012
- [SAG10] Sousa, N., Alves, G.R., Gericota, M.G. "An Integrated Reusable Remote Laboratory to Complement Electronics Teaching," *IEEE Transactions on Learning Technologies*, Vol. 3, pp. 265-271. July-Sept. 2010
- [SB07] Sanchez, B., Bragos, R. "Modular workbench for in-situ and remote laboratories", in *Proceedings of IEEE Instrumentation and Measurement Technology Conference (IMTC)*, Vol. 1-5, pp. 2137-2140, ISSN:1091-5281, Warsaw (Poland), May 2007

- [SBP+05] Scapolla, A.M., Bagnasco, A. Ponta, D., Parodi, G. "A Modular and Extensible Remote Electronic Laboratory" *International Journal of Online Engineering (ijOE)*. Vol 1, No 1. ISSN: 1861-2121. 2005
- [Sch08] Schwartz, J. "PXI Express Technology and Mainstream Applications", in *Proceedings of IEEE AUTOTESTCON*, Vo.1-2, pp- 169-172, ISSN:1088-7725, Salt Lake City (USA), 2008
- [Sch98] Schiffer, C.C. "Teaching in the 21st century". *The Internet and Higher Education*. Vol. 1, Issue 4, pp. 281-290. 1998
- [Sch99] Schmid, C. "A remote laboratory using virtual reality on the Web", in *Transactions of the Society for Modeling and Simulation International*, Vol.73, Issue 1, pp.13-21, ISSN: 0037-5497. Jul. 1999
- [SCP99] SCPI Technical Specification, Standard Commands for Programmable Instruments (SCPI), IVI Foundation, 1999
- [SCU07] Samoila, C., Cosh, S.G., Ursutiu, D. "Competences, Remote Labs and Bologna Process" in *Advances on remote laboratories and e-learning experiences*, Ed. by L. Gomes and J. Garcia Zubia, University of Deusto, Bilbao, 2007, pp. 63-96. ISBN: 978-84-9830-077-2
- [SHH+06] Spanik, P., Hargas, L., Hrianka, M., Kozehuba, I. "Application of virtual instrumentation LabVIEW for power electronic system analysis". *Proceedings of 12th International Power Electronics and Motion Control Conference*, Vol. 1-4. pp 457-460. Portoroz, Slovenia, Agosto 30- Septiembre 01, 2006
- [SN05] Sinha, R., Nagurka, M.L. "Analog and LabVIEW-based control of a maglev system with NI-ELVIS". *Proceedings of the ASME Dynamic Systems and Control Division 2005*. pp. 741-746. Orlando, USA. Noviembre 05-11, 2005
- [Soy00] Soysal, O. A. "Computer Integrated Experimentation in Electrical Engineering Education over Distance," in *Proceedings of American Society for Engineering Education Annual Conference*. (ASEE '00). 2000.
- [SPC+01] Safaric, R., Parkin, R.M., Czarnecki, C.A., Calkin, D.W "Virtual environment for telerobotics", in *Transactions on Integrated Computer-Aided Engineering*, Vol. 8, Issue 2, pp. 95-104. ISSN: 1069-2509. 2001

- [SY10] Shu-Hao Ye, Yen-Chu Hung. "The Study of Self-Seamless Teaching Strategy for Ubiquitous Learning Environments" in *Proceedings of 6th IEEE International Conference on Wireless, Mobile and Ubiquitous Technologies in Education (WMUTE)*, pp. 182-186. 2010
- [TSM+11] Tawfik, M., Sancristobal, E., Martín, S., Gil, C. et al. "VISIR Deployment in Undergraduate Engineering Practices" in *Proceedings of the 41st ASEE/IEEE Frontiers in Education Conference*, pp. T1A1 - T1A6, Rapid City (USA), October 12-15, 2011
- [Tza09] Tzafestas, S.G. "Web-Based Control and Robotics Education", *Intelligent Systems, Control and Automation: Science and Engineering*, Vol. 38. Ed. Springer-Verlag, 2009.
- [UJK+07] Ubar, R., Jutman, A., Kruus, M., Orasson, E., Devadze, S., Wuttke, H.D. "Learning Digital Test and Diagnostics via Internet", *International Journal of Online Engineering (iJOE)*, Vol.3, No. 1. ISSN: 1861-2121. 2007
- [Uni10] University of London External Programme Website, "Key Facts" http://www.londonexternal.ac.uk/about_us/facts.shtml, 2010.
- [VIS10] VISA Implementation, VPP-4.3: The VISA Library, IVI Foundation 2010
- [VM10] Vadillo, M.A., Matute, H. "Augmentation in contingency learning under time pressure" in *British Journal of Psychology*, Vol. 101, No.3. pp. 579-589. ISSN: 0007-1269. August 2010
- [VX10] VXI Technical Specification revision 4, VXIBus Organization, May 2010
- [VXI10] VXIPlug&play Specifications, VPP-2: System Frameworks Specification, IVI Foundation 2010
- [Wol06] Wolle, J. "LXI-the new LAN standrad for networking T&M equipment", *News from Rohde&Schwarz*, No. 190, Issue II, 2006
- [ZGH07] Zackrisson,J., Gustavsson, I., Håkansson, L. "An Overview of the VISIR Open Source Software Distribution 2007", in *Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV)*, Porto, Portugal, June 25 - 27, 2007.

- [ZS08] Zackrisson, J., Svahnberg, C. "OpenLabs Security laboratory - The online security experiment platform", in *Proceedings of Remote Engineering & Virtual Instrumentation Conference (REV)*, Düsseldorf, June 2008.

Apéndices

A

Configuración y uso de drivers IVI

Aunque ya introducíamos en capítulos anteriores, qué eran los drivers IVI y porqué los hemos seleccionado como tecnología de acceso a los recursos de los instrumentos del laboratorio, vamos a retomar alguno de estos conceptos para describir el marco en el cual hacemos uso de este tipo de drivers en la metodología de control aquí propuesta.

Como definíamos previamente, un driver es un conjunto de rutinas software que permiten controlar un instrumento programable, sin necesitar conocer el protocolo de programación de cada instrumento, lo que agiliza enormemente el desarrollo de algoritmos de configuración, escritura y lectura de los mismos.

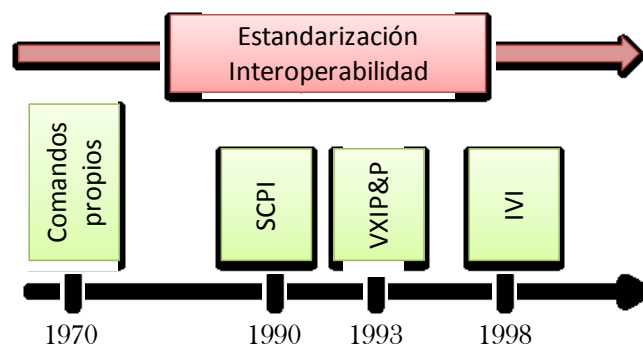


Figura A.1: Evolución de los estándares de control de instrumentos

En la Figura A.1 podemos ver como a lo largo de los años, las tecnologías destinadas a proporcionar acceso a la programación de los instrumentos, han ido evolucionando con el objetivo de conseguir una mayor estandarización y sobre todo una mayor interoperabilidad entre los drivers y los instrumentos, de manera que un mismo algoritmo de control, pueda controlar varios instrumentos de la misma familia.

Ha sido por tanto esa búsqueda de un estándar que propicie que un mismo comando pueda ser utilizado en dos equipos de una misma clase sin importar el fabricante, la que ha favorecido la aparición del estándar IVI allá por el año 1998. Por ejemplo, para cualquier driver IVI desarrollado para un multímetro, el comando de medida *IviDmmMeasurement.Read*, es el mismo y ejecuta las mismas acciones, independientemente del fabricante del multímetro. Además, el funcionamiento y operativa de los drivers de una misma clase son idénticos, lo que facilita el aprendizaje y sobre todo agiliza su utilización.

Esta cualidad ha sido principalmente la que nos ha hecho seleccionar los drivers IVI como el estándar a utilizar en los algoritmos que hemos desarrollado para el control de la instrumentación de laboratorio, y comprobar así la validez del método de control propuesto en este trabajo. Recordemos que uno de los objetivos que planteamos al principio de este documento, consistía en obtener un método válido para cualquier instrumento, independientemente del fabricante del mismo.

Las familias o clases de instrumentos contempladas por IVI, aparecen recogidas en la Tabla A.1, junto con el nombre del driver IVI definido por el estándar para cada uno de ellos.

Clase	IVI Driver
Multímetro digital	IviDmm
Osciloscopio	IviScope
Generador de funciones	IviFgen
Fuente de alimentación DC	IviDCPwr
Matriz de conmutación	IviSwitch
Medidor de potencia	IviPwrMeter
Analizador de espectros	IviSpecAn
Generador de señales RF	IviRFSigGen

Tabla A.1: Clases de instrumentos y drivers definidos en el estándar IVI

Junto a estas clases de drivers, el estándar IVI también define los denominados drivers a medida, los cuales son usados para aquellos instrumentos que no pertenecen a ninguna de las clases de la tabla anterior. Por ejemplo, no hay una clase definida para los analizadores de redes, por lo que el driver de este tipo de instrumentos, será un driver a medida, el cual tendrá toda la consistencia y beneficios definidos de un driver IVI, excepto la interoperabilidad.

Y, ¿cuáles son las ventajas de utilizar drivers IVI? Ya las enunciábamos en el capítulo 2 cuando explicábamos por qué habíamos elegido este tipo de drivers como núcleo principal sobre el que sustente el método de control que aquí proponemos. Pero las podemos agrupar de la siguiente forma:

- Alto rendimiento: ya que estos drivers integran una serie de funcionalidades que permiten conocer el estado del instrumento en cada momento y actualizar aquellas variables que se requiera en cada momento. Recordemos que cuando explicábamos la arquitectura lógica del Servidor de Equipos y Experimentos, comprobar el estado de cada instrumento antes de ejercer ninguna acción sobre él para reducir su número al mínimo, era lo primero que hacíamos, lo cual es posible gracias a los drivers IVI.

- Interoperabilidad: entendida como la capacidad de que un mismo algoritmo destinado a controlar un instrumento, pueda ser utilizado para controlar un instrumento de la misma clase sin importar el fabricante. Esta característica, importantísima como veíamos en este trabajo, facilita tanto el mantenimiento como la reutilización del código desarrollado, así como la propia instrumentación del laboratorio.
- Facilidad de desarrollo: el estándar IVI proporciona una serie de herramientas, accesibles desde la gran mayoría de los entornos de programación, que facilita la configuración de los drivers de los instrumentos. Además, como explicábamos también en el capítulo segundo, los drivers IVI hacen uso de VISA para la comunicación con los instrumentos, tecnología ya madura y probada que contiene una serie de utilidades para hacer posible la comunicación entre el instrumento físico y su algoritmo de control.

Así pues, una vez que sabemos qué son los drivers IVI y por qué los vamos a utilizar, veamos cómo los hemos empleado en la solución aquí propuesta.

A.1 Uso de drivers IVI

Tal y como decíamos en el punto anterior, el estándar IVI proporciona una serie de utilidades que tienen como finalidad ayudar a los desarrolladores a hacer uso de este tipo de drivers. Una de estas utilidades es la denominada IVI Shared Components. Se trata de una serie de componentes software, desarrollados y disponibles en la propia web del estándar (<http://www.ivifoundation.org>), que tienen como objetivo proporcionar una serie de servicios comunes a todos los drivers IVI y que ayudan al desarrollador a utilizarlos desde cualquier aplicación software.

Son servicios que ayudan a gestionar, administrar y configurar los drivers de los diferentes instrumentos conectados en este caso, al Servidor de Equipos y Experimentos. De este modo, el algoritmo de control únicamente se tienen que encargar de definir la lógica que queremos aplicar a cada uno de los equipos, dejando la parte administrativa a estas aplicaciones proporcionadas por el propio estándar y común a todos los drivers IVI.

Así, en el método de control que aquí proponemos hemos separado las funciones de control, administración, gestión y configuración, y comunicación con los equipos, en diferentes niveles dentro del Servidor de Equipos y Medidas. Proponemos esta división para independizar completamente el control de los instrumentos del resto de tareas del Servidor de Equipos y Medidas y además, hacer posible que actualizaciones o modificaciones de los equipos afecten únicamente a su configuración y no a su lógica de control, siendo común a todos ellos. La Figura A.2 muestra estos tres niveles:

1. Nivel lógico: formado por los algoritmos destinados a controlar los instrumentos y experimentos del laboratorio. Estos algoritmos permanecen inalterados en el caso de hacer cambios en los instrumentos, siempre y cuando éstos permitan la implementación de drivers IVI. En este caso, tal y como razonábamos anteriormente, hemos seleccionado LabVIEW como herramienta de desarrollo de estos algoritmos, pero puede utilizarse cualquier otra tecnología (C/C++, Matlab, etc) y seguir aplicando los conceptos que proponemos con el método de control propuesto, dejando la elección de la tecnología al desarrollador. Es decir, la metodología y estructura diseñada en este trabajo doctoral pueden ser aplicadas a otros desarrollos realizados con cualquiera de las tecnologías citadas anteriormente.
2. Nivel de administración, gestión y configuración: se encarga, mediante las aplicaciones proporcionadas por los IVI Shared Components, de la instalación y configuración de los drivers de cada uno de los instrumentos controlados por el Servidor de Equipos y Experimentos. Cada instrumento en el servidor estará referenciado mediante un *Nombre Lógico*, que será el nombre utilizado por el algoritmo de control para hacer las llamadas a dicho equipo. Este nombre permanece inalterado aunque cambie el instrumento, de manera que la referencia a dicho equipo en los algoritmos de control no cambia. Más adelante explicamos cómo se realiza esta configuración.

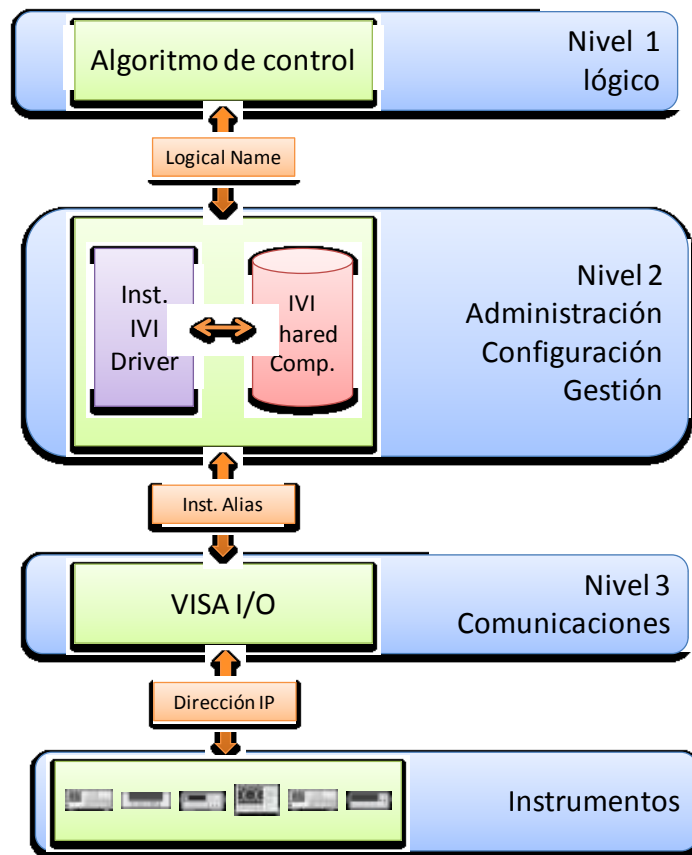


Figura A.2: Arquitectura lógica del método de control propuesto

3. Nivel de comunicación con los equipos: haciendo uso de las funcionalidades proporcionadas por VISA, este nivel permite configurar las direcciones IP de cada uno de los instrumentos, las cuales han sido asignadas durante la configuración de la red LXI del laboratorio, y gestionadas por el Servidor de Equipos y Experimentos, asignando a cada uno de ellos un alias, el cual es usado para indicar a cada uno de los drivers previamente configurados, qué recurso físico es al que deben dirigir los comandos de control solicitados desde el algoritmo del primer nivel. Gracias a este sistema de niveles, si se tuviera que modificar la ubicación de alguno de ellos, únicamente se debería cambiar en este nivel la referencia a su dirección física, sin influir en ninguno de los otros dos niveles.

En los siguientes apartados explicamos cómo se desarrollan los procesos llevados a cabo en los Niveles 2 y 3 para llevar a cabo la configuración de los drivers IVI que se quieren utilizar para controlar un conjunto de

instrumentos y una vez configurados, explicaremos con un sencillo ejemplo, cómo se utiliza uno de estos drivers para controlar un equipo del laboratorio, en lo que sería el desarrollo de un algoritmo desplegado en el denominado Nivel 1 de la Figura A.0.2.

A.2 Configuración de drivers IVI

El servicio principal del IVI Shared Component que facilita la configuración, administración y gestión de los drivers IVI del servidor, es el denominado **IVI Configuration Store**, cuyas funcionalidades son accedidas, en nuestro caso, a través del MAX. Este servicio tiene como misión principal crear las relaciones existentes entre la aplicación de control, los drivers y las referencias de entrada y salida, haciendo posible que el algoritmo de control desarrollado para controlar, por ejemplo, un multímetro, pueda ser usado sobre equipos de diferentes proveedores sin tener que cambiar dicha lógica de control.

Por lo tanto el servicio IVI Configuration Store tiene como funciones:

- Disponer de las referencias físicas de localización de los instrumentos para comunicarse con él.
- Proporcionar las asociaciones existentes entre los módulos software y las localizaciones específicas de los instrumentos, proporcionadas mediante VISA.
- Proveer de los módulos software necesarios y de referencias a los mismos para ser invocados desde las aplicaciones y poder controlar los instrumentos.

Para ello, el IVI Configuration Store está formado por las siguientes entidades lógicas, las cuales aparecen representadas en la Figura A.3:

- Módulo software: contiene los comandos y funciones necesarias para que el algoritmo de control pueda comunicarse con el instrumento. Este módulo contiene el driver IVI específico de cada instrumento y es proporcionado por el fabricante del equipo.
- Recurso hardware: el cual identifica el puerto de comunicación a través del cual está el servidor conectado con el instrumento. En nuestro caso, este recurso hardware apuntará al alias del instrumento proporcionado por VISA.

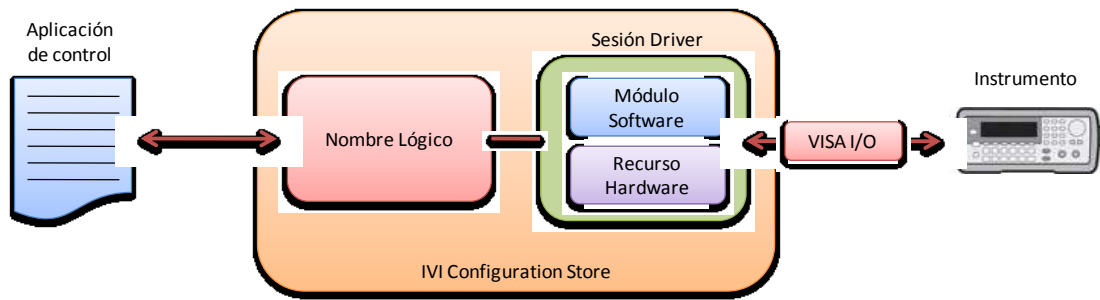


Figura A.3: Arquitectura lógica del IVI Configuration Store

- Sesión del driver: es el encargado de asociar el módulo software con el recurso hardware. Se dispone de una sesión del driver para cada uno de los instrumentos en sus posibles localizaciones. Instrumentos idénticos conectados en diferentes localizaciones, pueden usar el mismo módulo software, pero como su recurso hardware es diferente al estar conectados en diferentes lugares, necesitarán diferentes sesiones del driver.
- Nombre lógico: es la referencia proporcionada por el IVI Configuration Store para acceder a la sesión del driver configurada. Podemos entenderlo como un puntero a la sesión del driver. Es el nombre lógico al que se accede desde el algoritmo de control del nivel 1 (Figura A.2), independizando completamente la lógica del algoritmo del driver y de la dirección física del instrumento.

Tal y como introducíamos en un apartado anterior, para poder configurar, administrar y gestionar los drivers IVI de los instrumentos controlados por el Servidor de Equipos y Experimentos, National Instruments proporciona junto con LabVIEW la herramienta Measurement & Automation (MAX) que sirve para acceder a los servicios proporcionados por los IVI Shared Components. Otras plataformas de desarrollo proporcionan sus propias herramientas, como es el caso del *Test & Measurement Tool* de MatLab o el *Instrument Manager* de AgilentVEE, por lo que si el desarrollador prefiere utilizar cualquier otra tecnología de desarrollo para el algoritmo de control del nivel 1 (Figura A.2), dispondrá de otras herramientas para la configuración de los drivers llevada a cabo en los niveles 2 y 3.

Así pues, haciendo uso del MAX podemos configurar tanto el Nombre Lógico como la Sesión del Driver de nuestro instrumento. Para ello el primer paso consiste en asignar al instrumento que queremos controlar, una

dirección IP y asociar a esa dirección IP un alias. En el ejemplo que hemos seguido y que podemos ver en la Figura A.4, hemos asignado al instrumento la dirección IP 192.168.55.93 y el alias *FGenerator* ya que se trata de un generador de funciones.

Gracias a las funcionalidades VISA incluidas dentro de la aplicación MAX, la relación alias-dirección IP queda automáticamente establecida, de modo que cualquier petición realizada a la referencia *FGenerator*, será automáticamente enviada al instrumento localizado en esta dirección IP.

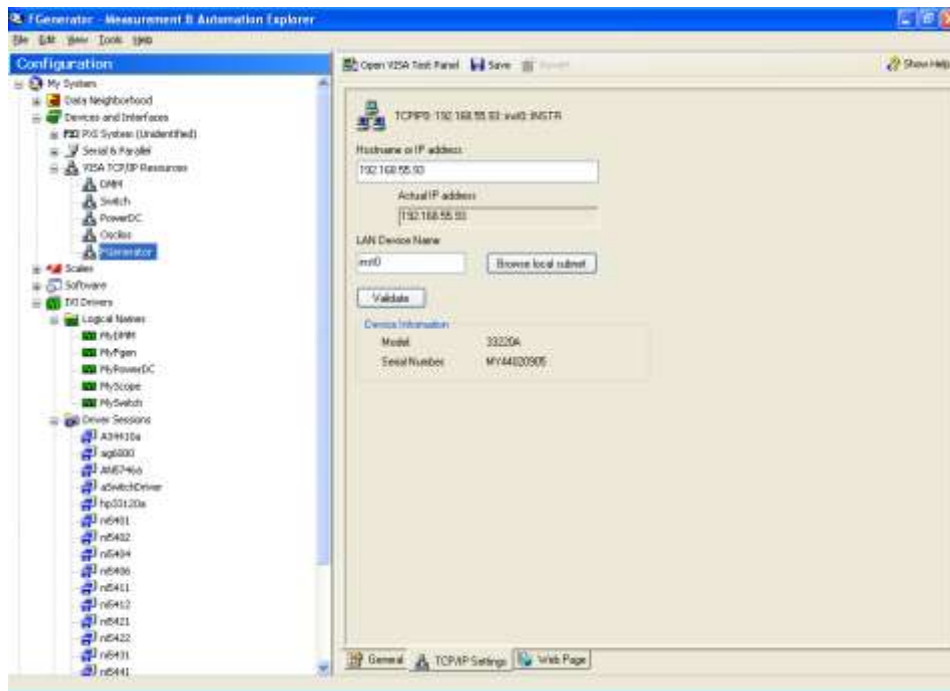


Figura A.4: Configuración del alias VISA del instrumento

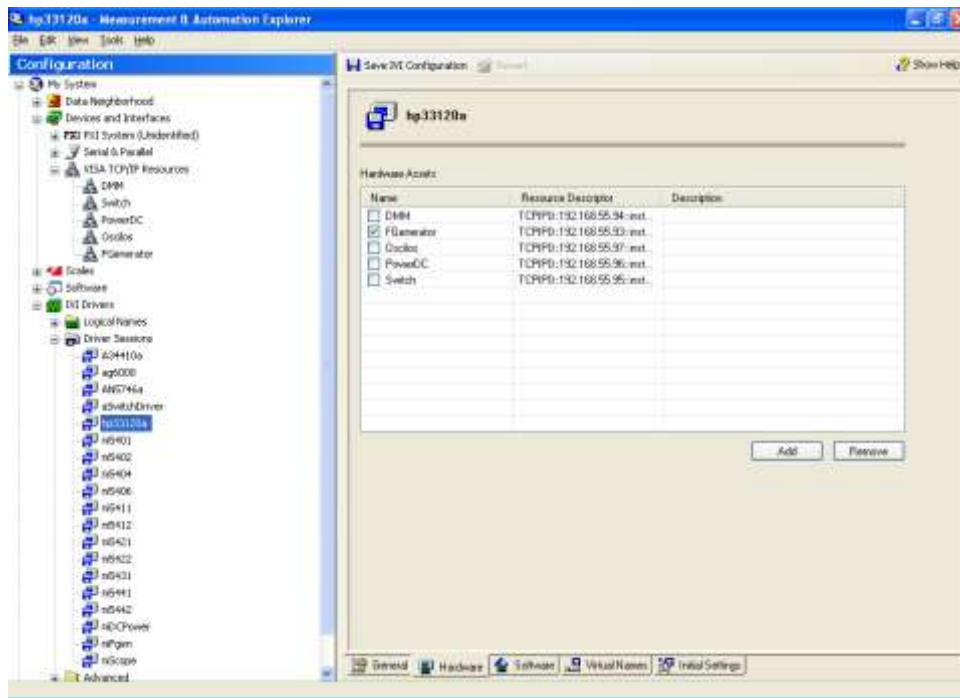


Figura A.5: Configuración del Recurso Hardware del driver

Una vez configurado la dirección física del instrumento, podemos proceder a configurar la Sesión del Driver que vamos a utilizar para controlar el equipo. Para ello, vamos a configurar primero el Recurso Hardware del mismo. Para ello, una vez que hemos instalado el driver IVI del instrumento proporcionado por el fabricante, nos aparecerá bajo el epígrafe “*Driver Session*” en el MAX. En nuestro ejemplo, se trata del equipo HP33120A. Si hacemos click en la pestaña “*Hardware*”, nos aparecerá la pantalla vista en la Figura A.5, en donde podremos asociar al Recurso Hardware, el alias VISA que hemos configurado en el paso anterior.

De este modo, queda asociado el recurso con dirección IP 192.168.55.93 y alias *Fgenerator*, a la sesión del driver del instrumento HP33120A. Al mismo tiempo, en la pestaña “*Software*” aparecerá tras la instalación del driver del instrumento la versión del driver IVI de dicho instrumento y los equipos del mismo fabricante que pueden ser controlados por el mismo driver (Figura A.6).

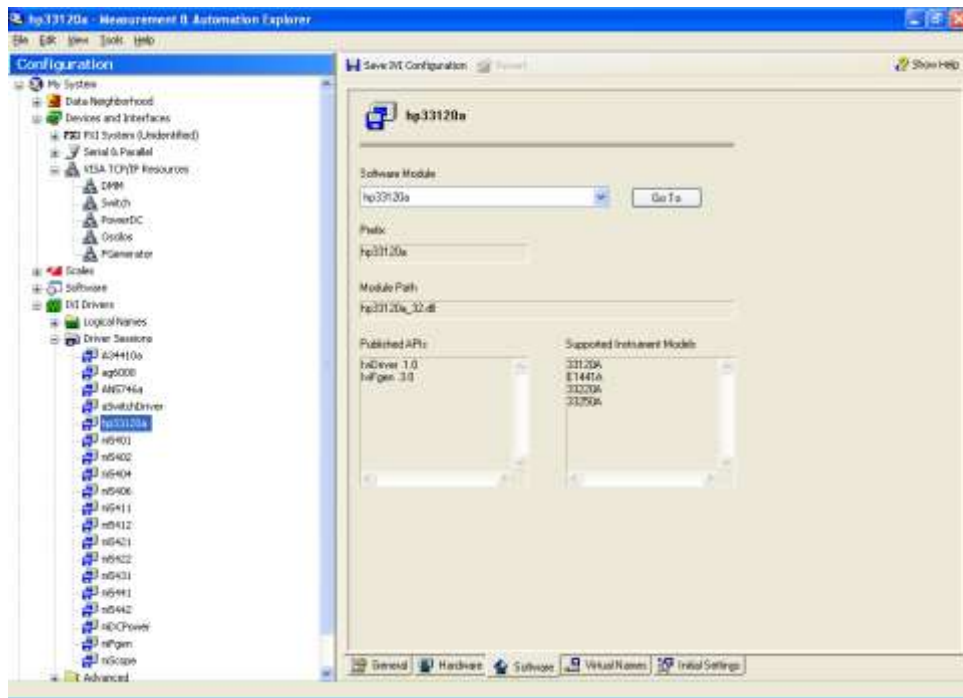


Figura A.6: Configuración del Módulo Software del driver

Es en este Módulo Software en donde se encuentra el driver IVI del instrumento, de manera que cuando desde la aplicación de control se solicita una acción sobre el equipo, este Módulo Software se encarga de interpretarla y enviarla al instrumento con la dirección IP configurada mediante VISA.

Si ahora creamos un Nombre Lógico que apunte a esta Sesión del Driver (Figura A.7), cada vez que queramos hacer una llamada desde el algoritmo de control a dicho instrumento lo haremos mediante este nombre, ya que al configurarlo le indicamos a que Sesión de Driver queremos hacer referencia, o lo que es lo mismo, a que equipo queremos dirigir nuestras peticiones de control. En el ejemplo, hemos creado el Nombre Lógico *MyFgen*, que vemos cómo apunta a la Sesión del Driver HP33120A.

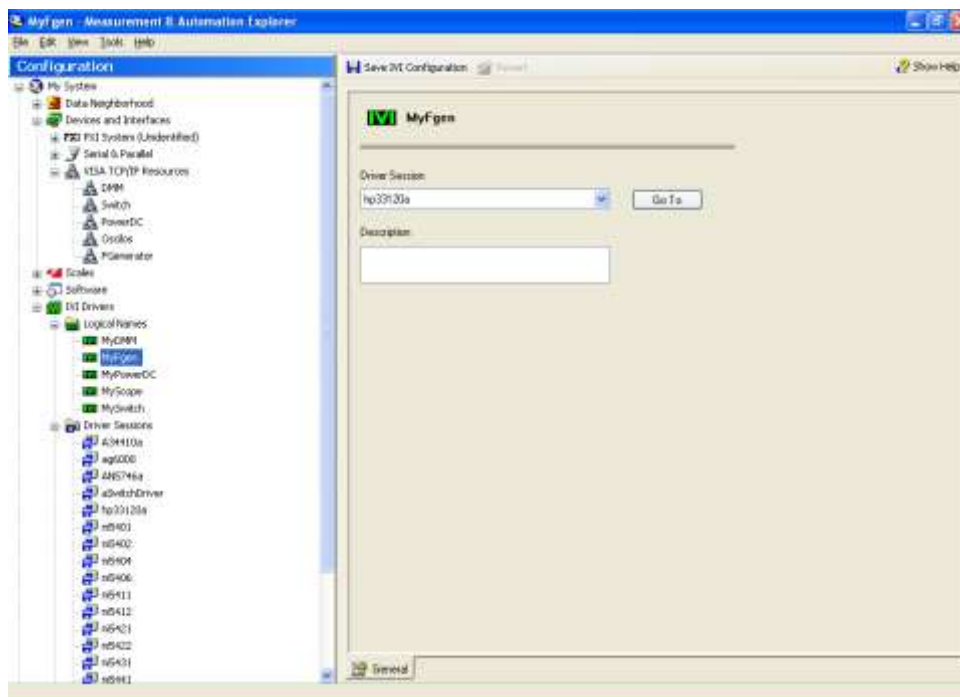


Figura A.7: Configuración del Nombre Lógico del driver

Esta configuración permite fácilmente la sustitución de un instrumento por otro de la misma clase sin la necesidad de cambiar ninguna parte de la aplicación de control. Únicamente modificando la Sesión del Driver mediante la instalación del driver del nuevo equipo, y haciendo que el Recurso Hardware de esta sesión apunte a la IP de nuevo instrumento, y que el Nombre Lógico apunte a esta nueva Sesión de Driver, es suficiente. No necesitamos ni siquiera cambiar al Nombre Lógico al que hacemos referencia en el algoritmo de control pues únicamente estamos accediendo a *MyFgen*, no estamos definiendo un instrumento en concreto en la aplicación. Estos cambios en la configuración aparecen representados en la Figura A.8, en donde tenemos dos generadores de funciones de dos fabricantes diferentes, pero referenciados mediante el mismo Nombre Lógico *MyFgen*.

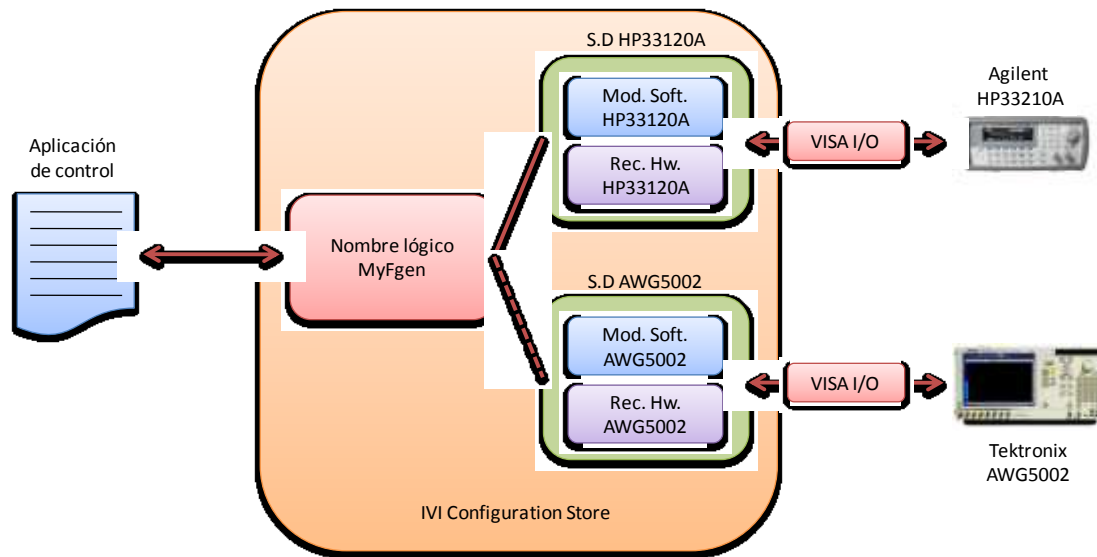


Figura A.8: Configuración de dos instrumentos con mismo Nombre Lógico

Así pues, haciendo uso de las características tanto de los drivers IVI como de sus complementos, podemos definir un método de control que permite el intercambio de unos instrumentos por otros, de modo sencillo y rápido. Por tanto, el algoritmo de control que desarrollemos siguiendo este método, se puede adaptar a los instrumentos que cada institución disponga en su laboratorio remoto, siendo de este modo una aplicación portable y desplegable para cualquier laboratorio. La única restricción es por tanto que los instrumentos a utilizar en el laboratorio puedan ser controlados mediante drivers IVI.

A.3 Ejemplo de uso de drivers IVI para el control de un instrumento

Si bien el punto anterior describe cómo se realiza la configuración de los niveles 2 y 3 mostrados en la arquitectura de la Figura A.0.2, en este apartado tratamos de explicar con un ejemplo cómo se desarrollaría una aplicación de control situada en el nivel 1 de dicha arquitectura.

La Figura A.9 muestra una aplicación sencilla para el control de un generador de funciones. Es un algoritmo que hemos desarrollado en LabVIEW a modo de sencillo ejemplo y en el que en ningún momento se realizan llamadas a funciones o se utilizan drivers o comandos propios de un generador de funciones específico. Es más, podemos apreciar como a simple

vista, no sabemos decir qué modelo de generador de funciones estamos controlando.

Observamos en la figura anterior como todos los bloques de funciones hacen referencia a la clase de driver IVIFgen, las cuales proporcionan acceso a todos los atributos definidos en el estándar para esta clase de instrumento y que describíamos a modo de ejemplo en el capítulo tercero, los cuales son comunes a todos los generadores de funciones que implementan el estándar IVI. De este modo, este código de control puede ser utilizado para controlar cualquier instrumento independientemente de su fabricante, pues está basado en el uso de las funciones genéricas proporcionadas por los drivers IVI.

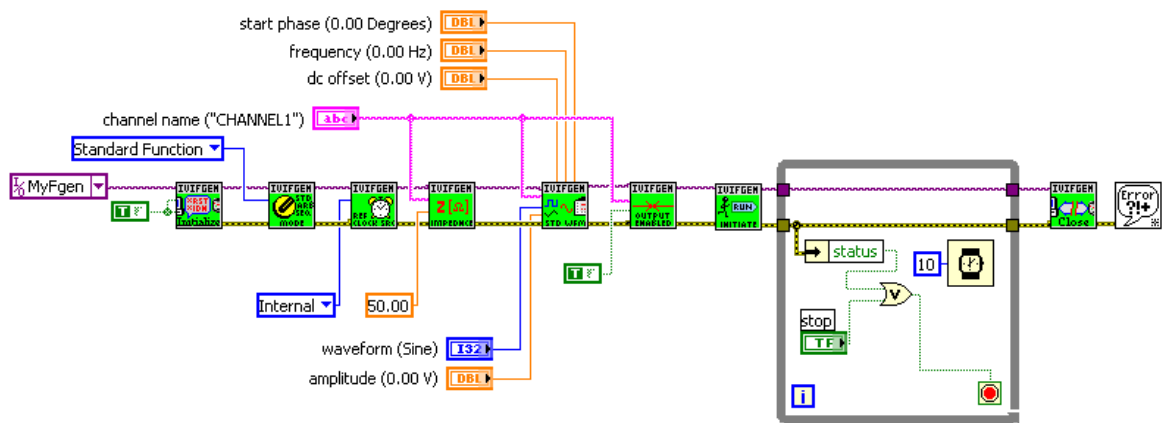


Figura A.9: Ejemplo de uso de driver IVI

B

Configuración de la red de instrumentos mediante LXI

Al estar LXI basado en el estándar Ethernet, permite definir una implementación uniforme e interoperable de una Red de Área Local (LAN) de modo que es muy sencillo y rápido integrar los instrumentos del laboratorio dentro de un sistema de test y medida modular. Además y muy importante en el caso del método propuesto, al estar LXI basado en Ethernet, las nuevas versiones que puedan aparecer en un futuro, deben ser compatibles con versiones anteriores, lo que garantiza el funcionamiento del sistema ante nuevas y posteriores actualizaciones del sistema de comunicaciones empleado.

Así pues, veíamos en el capítulo tercero cómo hemos creado una red instrumentos de laboratorio, basada en una topología de red en estrella, en donde el nodo central es el Servidor de Equipos y Experimentos, el cual gestiona, controla y centraliza a través de un concentrador o *hub*, las comunicaciones con los instrumentos del laboratorio.

Cada uno de los instrumentos, así como el servidor, dispone de una dirección IP local que le identifica en la red creada, de manera que el direccionamiento de los mensajes se realiza de manera inequívoca entre los nodos de la red. Además, si recordamos como se configuraban los drivers de cada uno de los instrumentos, esta dirección IP es la que utilizamos para referenciar cada uno de los instrumentos mediante su correspondiente alias en la arquitectura VISA, al que posteriormente llamamos desde la

configuración de los drivers IVI de cada instrumento. La configuración de cada una de las direcciones de red de los instrumentos se realiza de manera automática y sencilla. Esto es gracias al uso del estándar LXI. Recordando el análisis que realizábamos en el capítulo 2, el estándar LXI define tres clases de instrumentos, denominadas A, B y C, de modo que las funcionalidades de cada una de las clases están jerarquizadas y basadas cada una de ellas en la anterior tal.

La clase C contiene la extensión del estándar LXI más sencilla, en donde los instrumentos conectados a la red, se comportan como cualquier otro dispositivo que implemente el estándar Ethernet. Esta característica incluye la capacidad de detectar de manera automática cualquier instrumento LXI conectado a una red LAN, gracias a la implementación de esta clase de instrumentos, del protocolo VXI-11 y que ya mencionábamos en el capítulo segundo de este trabajo. Así, las funcionalidades incluidas en este protocolo permiten una fácil y rápida integración de instrumentos LXI en una red LAN, mediante la asignación dinámica de una dirección IP a través del protocolo DHCP (Dynamic Host Configuration Protocol) o una IP estática, la cual puede ser configurada a través de un navegador web. Esto es, los instrumentos LXI de clase C (recordemos que es la más básica y que el resto de clases superiores heredan sus características y funcionalidades), implementan también un servidor web el cual permite acceder a ciertas configuraciones de control y configuración del instrumento. A modo de ejemplo, en la Figura B.1 se muestra la pantalla de configuración de la red de un instrumento LXI a la cual hemos accedido a través de un navegador web. En la parte superior, vemos como podemos asignar una IP al instrumento mediante DHCP o bien asignarle una dirección manualmente, en cuyo caso deberemos rellenar en consecuencia los campos que aparecen en la parte inferior.

Por otro lado, y tal y como analizábamos en el capítulo segundo, los instrumentos LXI-clase C, deben ser totalmente compatibles con el estándar IVI, es decir, que todas sus funcionalidades y parámetros controlable deben ser accesibles mediante aplicaciones que hagan uso de los drivers definidos por este estándar. Esta funcionalidad proporcionada por los instrumentos LXI-clase C, hace que la utilización en el laboratorio de instrumentos que implementen este estándar sea recomendable, ya que se adaptan perfectamente al método de control propuesto.

Parameter	Configured Value	Edit Configuration
IP Settings may be obtained automatically using the following:		
DHCP: *	OFF	<input type="radio"/> OFF <input checked="" type="radio"/> ON
Automatic Private IP: *	OFF	<input type="radio"/> OFF <input checked="" type="radio"/> ON
IP Settings to use if automatic modes are off or servers are unavailable:		
IP Address: *	192.168.55.93	192.168.55.93
Subnet Mask: *	255.255.255.0	255.255.255.0
Default Gateway: *	192.168.55.254	192.168.55.254
DNS Server: *	0.0.0.0	0.0.0.0
Hostname: *	A-33220A-20905	A-33220A-20905
Domain: *		
Net BIOS: *	ON	<input type="radio"/> OFF <input checked="" type="radio"/> ON
Description:	Agilent 33220A (MY44020905)	Agilent 33220A (MY44020905)
LAN Keepalive Timeout (sec):	1800	1800
Ethernet Connection Monitoring: *	ON	<input type="radio"/> OFF <input checked="" type="radio"/> ON
GPIO Address:	10	10
Telnet Prompt:	33220A>	33220A>
Telnet Welcome Message:	Welcome to Agilent's 33220A Waveform Generator	Welcome to Agilent's 33220A Waveform Gene
Change Password:	<input type="text"/> (Enter Old)	<input type="text"/> (Enter New) <input type="text"/> (Confirm New)

Figura B.1: Ejemplo pantalla de configuración de red en un equipo LXI

Podemos por lo tanto afirmar que si queremos controlar una serie de instrumentos mediante el método aquí propuesto, la única restricción existente en cuanto a las especificaciones de los mismos, es que deben implementar el estándar LXI, ya que en su clase más sencilla, la denominada clase C, ya llevan desarrolladas intrínsecamente las funcionalidades necesarias para ser controlados mediante drivers IVI, base tecnológica en la que se sustenta el método de control definido.

Finalmente y gracias a la estructura jerárquica de las clases definidas en el estándar LXI, si hiciéramos uso de instrumentos que implementasen bien la clase B o bien la clase A del estándar, el método de control es completamente válido y no sería necesario ninguna modificación de los algoritmos de control.