UNIVERSIDAD DE DEUSTO

MEJORA DE LA FIABILIDAD Y SOSTENIBILIDAD DE LOS LABORATORIOS REMOTOS A TRAVÉS DE NUEVAS ARQUITECTURAS ORIENTADAS A LA CONCURRENCIA Y LA DETECCIÓN DE FALLOS

Tesis doctoral presentada por Aitor Villar Martínez

Dirigida por el Dr. Javier García Zubía y el Dr. Ignacio Angulo Martínez

Bilbao, septiembre de 2022

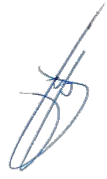
UNIVERSIDAD DE DEUSTO

MEJORA DE LA FIABILIDAD Y SOSTENIBILIDAD DE LOS LABORATORIOS REMOTOS A TRAVÉS DE NUEVAS ARQUITECTURAS ORIENTADAS A LA CONCURRENCIA Y LA DETECCIÓN DE FALLOS

Tesis doctoral presentada por Aitor Villar Martínez
dentro del programa de doctorado en *Ingeniería para la Sociedad de la Información y el Desarrollo Sostenible*

Dirigida por el Dr. Javier García Zubía y el Dr. Ignacio Angulo Martínez

El doctorando



Los directores

GARCIA ZUBIA, JAVIER (AUTENTICACIÓN)
Firmado digitalmente por GARCIA ZUBIA, JAVIER (AUTENTICACIÓN)
Nombre de reconocimiento (DN): c=ES, serialNumber=30562497M, sn=GARCIA, givenName=JAVIER, cn=GARCIA ZUBIA, JAVIER (AUTENTICACIÓN)
Fecha: 2022.09.14 09:46:41 +02'00'

ANGULO MARTINEZ, IGNACIO (AUTENTICACIÓN)
Firmado digitalmente por ANGULO MARTINEZ, IGNACIO (AUTENTICACIÓN)
Nombre de reconocimiento (DN): c=ES, serialNumber=30671306R, sn=ANGULO, givenName=IGNACIO, cn=ANGULO MARTINEZ, IGNACIO (AUTENTICACIÓN)
Fecha: 2022.09.14 09:45:34 +02'00'

Bilbao, septiembre de 2022

Autor: Aitor Villar Martínez
Directores: Dr. Javier García Zubía y Dr. Ignacio Angulo Martínez

Tesis impresa en Bilbao
Primera edición, septiembre de 2022

A mi familia y a Rakel

Resumen

Los laboratorios educativos a distancia, cuando están correctamente diseñados, han demostrado ser muy eficaces desde el punto de vista pedagógico. A lo largo de los años su tecnología ha evolucionado y ofrecen grandes capacidades, que en ocasiones son incluso superiores a los laboratorios presenciales. Sin embargo, a pesar de su potencial, todavía no son una herramienta extendida en los centros de enseñanza. Una de las principales razones es que muchos laboratorios remotos han estado tradicionalmente orientados a la investigación, es porque no han sido capaces de garantizar una calidad de servicio (QoS) suficientemente alta en un entorno educativo real. Dicha calidad de servicio requiere que un número relativamente alto de estudiantes pueda acceder a los laboratorios de forma simultánea, ya que las clases pueden incluir docenas de estudiantes. Asimismo, para lograr una calidad de servicio notable, se requiere fiabilidad: el laboratorio debe funcionar, estar disponible y proporcionar resultados correctos. Este trabajo trata de dar solución a estas dos cuestiones ya mencionadas, escalabilidad y fiabilidad, a través de una arquitectura para el desarrollo de laboratorios remotos. La primera parte de esta tesis mejora la calidad de servicio de los laboratorios remotos desde un punto de vista de escalabilidad. La solución de escalabilidad propuesta en este trabajo permite el desarrollo de laboratorios remotos con múltiples instancias de experimentación, con un coste contenido y una notable capacidad de escalabilidad. La segunda parte de esta tesis busca la mejora de la calidad de servicio de los laboratorios remotos desde un enfoque de fiabilidad. La solución de fiabilidad propuesta en este trabajo permite dotar a los laboratorios remotos, mediante técnicas de detección de fallos, de una calidad de servicio mejorada. La parte tercera y final de este trabajo aúna en una arquitectura para el desarrollo de laboratorios remotos ambas soluciones tecnológicas previamente introducidas. Esta arquitectura, denominada *WebLabPRO*, permi-

te la creación de laboratorios remotos multi-instancia, escalables y confiables, y con una alta calidad de servicio y percepción de calidad por parte del usuario. Cada una de las dos primeras partes de este trabajo han sido evaluadas por separado. Además, la arquitectura WebLabPRO también ha sido evaluada como solución global, durante un periodo que abarca más de dos años de uso, y mas de 70.000 sesiones de experimentación por parte de usuarios en un ámbito global, con múltiples instituciones e instancias de experimentación. Dicho análisis se ha realizado en un contexto no controlado, con el fin de obtener datos en situaciones de uso real. Estos análisis se han realizado en diferentes laboratorios remotos alojados en la red global *LabsLand* de laboratorios remotos. Entre ellos se pueden encontrar laboratorios para experimentación con microcontroladores, con FPGAs o con robots programables. Los resultados muestran que la calidad del servicio de los laboratorios remotos desplegados sobre la arquitectura WebLabPRO es adecuada y que por tanto, este enfoque puede derivar en una adopción generalizada de los laboratorios remotos como herramientas de enseñanza en el ámbito educativo.

Abstract

Distance learning educational laboratories, when well designed, have proven to be very effective from a pedagogical point of view. Over the years, their technology has evolved and they offer great capabilities, sometimes even superior to face-to-face laboratories. However, despite their potential, they are still not a widespread tool in schools. One of the main reasons is that many remote laboratories have traditionally been research-oriented, is because they have not been able to guarantee a sufficiently high quality of service (QoS) in a real educational environment. Such QoS requires that a relatively large number of students can access the labs simultaneously, as classes may include dozens of students. Also, to achieve a remarkable QoS, reliability is required: the laboratory must be functional, available and provide correct results. This work attempts to provide a solution to these two aforementioned issues, scalability and reliability, through an architecture for the development of remote laboratories. The first part of this thesis improves the quality of service of remote laboratories from a scalability point of view. The scalability solution proposed in this work allows the development of remote laboratories with multiple experimentation instances, with a contained cost and a remarkable scalability capacity. The second part of this thesis seeks to improve the quality of service of remote laboratories from a reliability approach. The reliability solution proposed in this work allows to provide remote laboratories, through fault detection techniques, with an improved quality of service. The third and final part of this work combines in an architecture for the development of remote laboratories both technological solutions previously introduced. This architecture, called *WebLabPRO*, allows the creation of multi-instance, scalable and reliable remote

laboratories with a high quality of service and user perception of quality. Each of the first two parts of this work have been evaluated separately. In addition, the WebLabPRO architecture has also been evaluated as a global solution, over a period spanning more than two years of use, and more than 70,000 user experimentation sessions in a global, multi-institution, multi-user environment. This analysis has been performed in an uncontrolled context, in order to obtain data in real use situations. These analyses have been performed in different remote laboratories hosted in the *LabsLand* global network of remote laboratories. These include laboratories for experimentation with microcontrollers, FPGAs or programmable robots. The results show that the quality of service of the remote labs deployed on the WebLabPRO architecture is adequate, and that this approach may lead to a widespread adoption of remote labs as teaching tools in education.

Agradecimientos

Este es el resultado de cuantiosas horas de trabajo. Una montaña rusa emocional, mi mayor proyecto. Es algo que comenzó a cuajarse allí en 2017 cuando comencé mis prácticas del Máster en LabsLand y descubrí la capacidad que los laboratorios remotos tienen de democratizar la tecnología en el mundo.

Una vez llegados a este momento, es necesario volver la vista atrás y dar gracias a todas aquellas personas que han compartido mi día a día y que han ayudado de una manera u otra a llevar a buen puerto este trabajo.

En primer lugar, debo agradecer sin medida a mis abuelos, Manuel y Mari Carmen, por compartir un trocito de su hogar conmigo durante mi periodo universitario en Bilbao, a cambio de nada.

Sigo por mi familia más cercana, las personas más importantes de mi vida. Gracias Alberto y Lourdes por acompañarme, no solo en esta etapa, sino en la vida. Gracias por ese apoyo y cariño incesante, tanto en los momentos de alegría como en los de flaqueza. Sin vuestro ímpetu, esto no hubiera sucedido. Rakel, hogar, sinónimos. Gracias por aguantar y estar ahí. Este trabajo es tan mío como tuyo.

En lo profesional, he gozado de suerte similar. Debo mostrar mi agradecimiento a todos los compañeros de LabsLand, en especial a Pablo y Luis. Sois culpables de que la curiosidad ganara, y decidiera inmiscuirme en esta etapa investigadora. Me he sentido a hombros de gigantes. Gracias por vuestros valiosos consejos e incesantes horas de *proof-reading*. Lucas, gracias por ser mis manos en LabsLand durante este último periodo.

No me olvido de todos aquellos compañeros que he conocido en la Universidad de Deusto, desde mis compañeros del Grado

y Máster, como los encontrados durante mi época investigadora. Gracias Iñigo Q., Iñigo G., Eduardo V., Ignacio F. y al resto que por brevedad, no incluyo de manera explícita aquí.

Finalmente, agradezco a mis primero compañeros, y luego directores, Javier e Ignacio. Gracias por volcaros en mí y confiarme este trabajo. Son muchas horas las que me habéis regalado, y con las que estaré siempre en deuda. Este camino que hemos andado, no termina aquí.

Aitor Villar Martínez

Septiembre de 2022

Índice general

Índice de figuras

Índice de tablas

Glosario

API Application Programming Interface

CAN Controller Area Network

CMOS Complementary Metal-Oxide-Semiconductor

CPLD Complex Programmable Logic Device

FIFO First Input - First Output

FPGA Field Programmable Gate Array

FSM Finite States Machine

GPIO General Purpose Input Output

H.264/AVC H.264 Advanced Video Coding

HLS HTTP Live Streaming

HTTP Hyper-Text Transfer Protocol

HTTPS Hyper-Text Transfer Protocol Secure

I2C Inter-Integrated Circuit

ID Identifier

IDE Integrated Development Environment

IEEE Institute of Electrical and Electronics Engineers

IP	Internet Protocol
JSON	JavaScript Object Notation
JTAG	Joint Test Action Group
LED	Light-Emitting Diode
LMS	Learning Management Systems
LTI	Learning Tools Interoperability
MOOC	Massively Open Online Course
OLED	Organic Light-Emitting Diode
PC	Personal Computer
PCB	Printed Circuit Board
PWM	Pulse-Width Modulation
QoS	Quality of Service
RE	Remote Experiment
REST	Representational State Transfer
RL	Remote Laboratory
RLMS	Remote Laboratory Management System
SoC	System on Chip
SPI	Serial Peripheral Interface
STEM	Science, Technology, Engineering and Mathematics
TCP	Transmission Control Protocol
TTL	Transistor-Transistor Logic
UDP	User Datagram Protocol

URL Uniform Resource Locator

USB Universal Serial Bus

VHDL VHSIC Hardware Description Language

VISIR Virtual Instrument Systems in Reality

WILSP Web-based Interactive Live-Streaming Platform

XML eXtensible Markup Language

CAPÍTULO

1

Introducción

1.1 Motivación

El diseño y desarrollo de laboratorios remotos educacionales exige la integración de una amalgama de tecnologías para permitir a los usuarios acceder a equipamientos hardware distantes, de manera remota, a través de Internet. El equipamiento que los usuarios manejan es real, y a través de él pueden aprender de manera similar a como lo harían en una situación tradicional en un laboratorio presencial convencional.

Para una institución educativa, como un colegio o universidad, contar con laboratorios remotos aporta numerosas ventaja técnicas. La más importante en este ámbito, es quizás, la comodidad. El hecho de que los recursos hardware sean accesibles a través de Internet permite a los estudiantes no tener que desplazarse físicamente para realizar sus clases prácticas y sus sesiones de experimentación. Esto cobra importancia en la actual situación post-pandemia, donde forzosamente nos hemos familiarizado con el teletrabajo y la enseñanza a distancia. Además, el hecho de que los recursos estén accesibles *24/7*, dota a los usuarios de mayor libertad temporal, permitiendo un mejor aprovechamiento de su tiempo personal.

Esta cuestión anterior es además de vital importancia para aquellas instituciones que buscan impartir sus conocimientos mediante formación a distancia. Éstas, tradicionalmente han encontrado limitaciones a la hora de impartir conocimientos prácticos, obligando tomar soluciones de alto impacto económico y medioambiental como la fabricación, compra y envío de recursos hardware (tarjetas de desarrollo hardware, por ejemplo) a cada uno de los alumnos matriculados en cursos de formación a distancia. Esta problemática puede verse solucionada de forma parcial o total gracias al uso de laboratorios remotos que permiten desarrollar la experimentación requerida en dichos cursos.

Además, en los últimos años, han surgido plataformas y entidades que permiten compartir los laboratorios remotos desarrollados entre diferentes instituciones educativas. Esto permite compartir recursos entre instituciones y estudiantes, lo cual propicia un ahorro de costes al reducir el gasto en adquisición y mantenimiento de equipamientos tecnológicos.

Finalmente, ampliando el foco, a nivel global la aparición de estas redes de compartición de recursos remotos hace que las ventajas aportadas por los laboratorios remotos no sean solo técnicas o económicas, sino también sociales. En línea con el objetivo para el desarrollo sostenible *ODS4* establecido en 2015 por la Asamblea General de las Naciones Unidas y que persigue garantizar para el año 2030 una educación inclusiva, equitativa y de calidad y promover oportunidades de aprendizaje durante toda la vida para todos, los laboratorios remotos permiten, inherentemente, romper barreras geográficas y socioeconómicas. La experimentación remota permite democratizar el acceso a la tecnología y hacerla accesible de manera global.

Por tanto, los laboratorios remotos pueden ser protagonistas de las mejoras educativas que los anteriores escenarios plantean. Para que esto sea realidad es necesario que el diseño de laboratorios remotos afronte la situación como un problema global y tenga en cuenta tres elementos fundamentales: la escalabilidad, la fiabilidad y el coste. Las dos primeras permiten que la experimentación remota sea útil a profesores, alumnos e instituciones en un escenario global y la tercera, que esta sea sostenible en el tiempo. Esta tesis se centra en la experimentación remota, y más concretamente en el diseño de laboratorios remotos útiles y sostenibles.

1.2 Hipótesis y objetivos específicos

Una vez expuesta la motivación de la presente tesis, se ha formulado la siguiente hipótesis de trabajo:

> *Es posible crear una arquitectura para el desarrollo de laboratorios remotos que facilite el escalado y dé soporte a un elevado número de usuarios concurrentes, con coste contenido, y que permita garantizar la fiabilidad del proceso de experimentación remota a través de técnicas de detección automática de fallos.*

La validez de la hipótesis es el reto principal de esta tesis, que de forma operativa se sustancia en una meta y en cuatro objetivos específicos. La meta queda expresada a continuación:

> *Diseñar e implementar una arquitectura (WebLabPRO) capaz de permitir el despliegue efectivo de laboratorios remotos escalables y fiables en un entorno real de uso con fines comerciales e industriales bajo un enfoque profesional.*

Mientras que los objetivos específicos (OE) se enuncian a continuación:

- OE1: Analizar las arquitecturas de hardware de los laboratorios remotos en el estado del arte en cuanto a su escalabilidad, potencial de concurrencia, eficiencia de costes y fiabilidad.
- OE2: Diseñar, implementar y validar una arquitectura para el desarrollo de laboratorios remotos que facilite la escalabilidad de los mismos.
- OE3: Diseñar, implementar y validar una arquitectura para el desarrollo de laboratorios remotos que facilite la fiabilidad de los mismos.
- OE4: Integrar las soluciones de escalabilidad y fiabilidad en una única arquitectura para su despliegue en un entorno industrial y comercial de experimentación remota.

1.3 Metodología

La consecución de los objetivos específicos anteriores exige el diseño y seguimiento de una metodología firme en su planteamiento y versatil en su uso.

Las fases principales se describen a continuación y se muestran en la Figura 1.1.

- Planteamiento de la hipótesis, desarrollo de los objetivos espécificos y planteamiento de las preguntas de investigación del campo de investigación de experimentación remota. La hipótesis de esta tesis se centra en que es posible mejorar la escalabilidad y la fiabilidad de los laboratorios remotos.

- Elaboración del estado del arte para situar la hipotesis y los objetivos específicos en el contexto de los resultados obtenidos por otros investigadores. El estado del arte se elabora por separado para cada una de las dos características objeto de estudio: escalabilidad y fiabilidad.

- Diseño, implementación y validación de la arquitectura propuesta, WebLabPRO. De nuevo, este proceso se presenta por separado para sus dos facetas, escalabilidad y fiabilidad, ya que tecnológica y funcionalmente son dos retos independientes. Es en esta fase donde se presenta un ciclo iterativo típico de investigación, de manera que cada nueva solución es la anterior, refinada.

- Integración y validación de las dos soluciones de escalabilidad y fiabilidad en una única arquitectura, WebLabPRO. De nuevo, este proceso será iterativo, y su evolución puede afectar a la anterior fase. Esto quiere decir que en la Figura 1.1 las fases en azul claro se completarán secuencialmente para los requisitos de escalabilidad y de fiabilidad.

- Validación de la hipótesis original mediante los resultados obtenidos en las distintas validaciones. La validación de la hipótesis pondrá especial énfasis en observar la consecución de los objetivos específicos.

- Publicación y divulgación de resultados en diferentes revistas y congresos de impacto internacional. De este modo, la nueva arquitectura

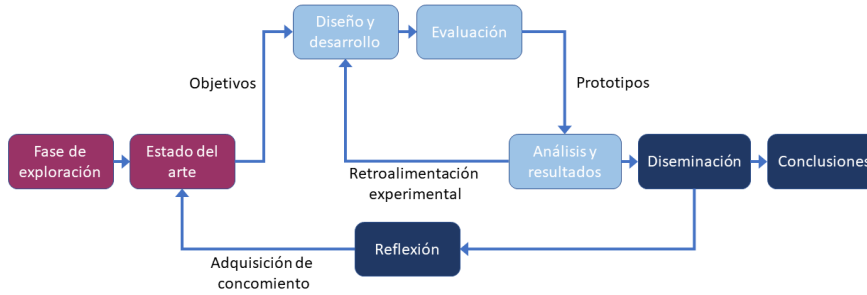


Figura 1.1: Metodología de la tesis.

WebLabPRO y su potencial quedarán compartidas con la comunidad investigadora en el ámbito de la experimentación remota. Durante la elaboración de la tesis, estas publicaciones ayudan a contrastar los avances de la misma con dicha comunidad investigadora.

Metodológicamente, la validación ha consistido en desplegar laboratorios remotos de la empresa LabsLand utilizando la arquitectura WebLabPRO, de esta forma la validación no se da en un entorno controlado por el doctorando, como en la clásica prueba de concepto, sino que se hace en un entorno de producción real involucrando alumnos y profesores que desconocen ser parte de esta validación. Este proceso de evaluación es exhaustivo y suele ser más propio de una tesis industrial, aunque no sea el caso.

1.4 Contribuciones científico-técnicas

La consecución de los cuatro objetivos específicos anteriores, ofrece tres contribuciones científico-técnicas (CCT). Esta doble vertiente de cada contribución, se explica porque la arquitectura resultante de la tesis exige proporcionar una aplicación directa y real para la empresa LabsLand. Por tanto, el caracter

principal de las contribuciones es técnico, siendo soportada cada una de ellas por una base científica.

- CCT1: Estado del arte en términos de escalabilidad y fiabilidad en la experimentación remota
- CCT2: Arquitectura WebLabPRO para el aporte de escalabilidad y fiabilidad en experimentación remota.
- CCT3: Laboratorios remotos escalables, fiables y con potencial para uso en producción basados en WebLabPRO.

1.5 Estructura

Antes de describir como se organiza esta tesis en sus capítulos, es importante explicar la estructura general utilizada. Los dos retos técnicos fundamentales de esta tesis son la escalabilidad y la fiabilidad, y cada uno de ellos se aborda por separado en los Capítulos 3 y 4, respectivamente.

Cada uno de estos capítulos mantienen un esquema similar, incluyendo un estado del arte y una validación por separado. Esto es así para facilitar la comprensión, análisis y validación de cada uno de los dos retos técnicos. El Capítulo 5 aúna en una misma arquitectura, denominada WeblabPRO, ambas soluciones tecnológicas y es en este capítulo donde la validación es completa.

Esta tesis se organiza en seis capítulos:

- El primer capítulo es un capítulo introductorio, e incluye la hipótesis, objetivos y contribuciones obtenidas en la investigación doctoral. Además incluye la metodología seguida.
- El segundo capítulo extiende la motivación presentada en el Capítulo 1 para describir de forma genérica la experimentación remota, poniendo especial interés en la escalabilidad y la fiabilidad. El objetivo de este segundo capítulo es describir el escenario de la investigación doctoral, sin profundizar en aspectos técnicos.
- El tercer capítulo aborda desde un punto de vista técnico el diseño, implementación y validación de la arquitectura WebLabPRO en su característica de escalabilidad. Este capítulo cuenta con un estado del arte en términos de escalabilidad aplicada a los laboratorios remotos.

- El cuarto capítulo aborda desde un punto de vista técnico el diseño, implementación y validación de la arquitectura WebLabPRO en su característica de fiabilidad y su relación con la calidad de servicio. Este capítulo cuenta con un estado del arte en términos de fiabilidad aplicada a los laboratorios remotos.

- El quinto capítulo tiene como objetivo la validación de la arquitectura WebLabPRO en el entorno industrial y comercial de la empresa proveedora de laboratorios remotos LabsLand. La arquitectura WebLabPRO se despliega en multiples laboratorios remotos de LabsLand, que son usados en entornos académicos no controlados por el doctorando. La validación es fundamentalmente cuantitativa respecto de la escalabilidad y fiabilidad, aunque también cuenta con un estudio cualitativo de satisfacción de la experiencia de uso por parte del alumno.

- El sexto y último capítulo aporta las conclusiones y las líneas de investigación futuras que suscitan este trabajo doctoral. Además se incluye una descripción de las publicaciones que divulgan los resultados de esta tesis, tres de ellas publicadas en revistas JCR con impacto Q1 y Q2.

CAPÍTULO

2

La experimentación remota y los laboratorios remotos

La experimentación remota es un campo activo desde hace más de 25 años. Durante este tiempo, los laboratorios remotos obtenidos han ido evolucionando de una forma clara y muy condicionados por las tecnologías. Así, los primeros laboratorios remotos comenzaron en al año 1995, en pleno despliegue de Internet. Es fácil pensar que toda la evolución tecnológica de esta red de comunicación ha afectado al desarrollo de los laboratorios remotos. De esta forma, se ha evolucionado desde laboratorios remotos que eran simples pruebas de concepto, difícilmente utilizables, a laboratorios remotos que son útiles y utilizables por la comunidad educativa sin restricciones de ningún tipo. Los profesores y alumnos, los usuarios, deben ser agnósticos frente a la tecnología que soporta cada laboratorio remoto.

2.1 Introducción

Estamos viviendo un proceso global de digitalización que afecta a la comunicación, la industria, las relaciones sociales y, por supuesto, a la educación. El campo de la educación a distancia u online está experimentando un enorme auge y su evolución depende de los avances en las tecnologías educativas. Los laboratorios remotos garantizan un proceso de experimentación real sin disponer de equipamiento y sin estar delante de los mismos, y por tanto son una herramienta indispensable en disciplinas STEM online o no.

En cierto sentido la experimentación remota tiene como objetivo superior el democratizar el acceso a la ciencia y la tecnología, en línea con el Objetivo 4 ODS: Garantizar una educación inclusiva, equitativa y de calidad y promover oportunidades de aprendizaje durante toda la vida para todos [1].

La experimentación remota ha avanzado significativamente en los últimos años y se ha convertido en una de las principales líneas de investigación en el campo del Technology Enhanced Learning (Froyd et al., 2012), que a su vez arranca con (Ramo, 1957).

Un experimento remoto ofrece al usuario una experiencia real, aunque el equipo no esté delante de él. Esto es, un experimento remoto es en general un experimento real mediado por Internet, donde las manos son el ratón y la interfaz, y los ojos son una webcam. Existen muchos tipos de experimentos remotos lo que da lugar a formas muy distintas de mediación.

La experimentación remota es un campo de investigación activo desde 1995 (Bower and Christensen, 1995; Bohus et al., 1996). La principal diferencia entre un experimento remoto y uno tradicional radica en la proximidad del estudiante. En un laboratorio clásico, el estudiante está físicamente frente al experimento y puede verlo y manipularlo directamente. Mientras tanto, en un experimento remoto el estudiante accede, controla y ve el proceso a través de Internet. La experimentación es real, no es una simulación.

Hay muchos experimentos remotos, especialmente en el campo de la tecnología y las ciencias experimentales (García-Zubía and Alves, 2012; Garciazubia, 2021). A los experimentos remotos hay que unir los experimentos virtuales, aquellos basados en un motor de simulación y que si bien tienen elementos en común con los experimentos remotos, también tienen obviamente diferencias notables: los remotos son reales mientras que los virtuales son si-

[1] https://www.un.org/sustainabledevelopment/es/education/

mulados. Una ventaja significativa que tienen los laboratorios remotos sobre los virtuales es que ofrecen un mayor realismo (Nedic et al., 2003): proporcionan acceso a hardware y equipos reales y permiten a los usuarios tener interacciones reales y obtener resultados reales. Este mayor realismo es expresado por otros investigadores como inmersión, de manera que a mayor inmersión mayor aprendizaje (Corter et al., 2007).

Frente a esto, y hasta ahora, la adopción de laboratorios virtuales en cursos formales es significativamente más común que la de laboratorios remotos (Potkonjak et al., 2016).

Además, cabe decir que en ningún caso los laboratorios remotos, los virtuales y los clásicos o hands-on son opuestos entre sí, son claramente complementarios y así son usados por los profesores. Esta tesis se centra en la mejora de la experimentación remota con el fin de facilitar y mejorar el trabajo del profesor STEM.

Los profesores pueden tener varias razones diferentes para integrar un laboratorio remoto en su enseñanza y estas se explican con más detalle en (De La Torre et al., 2020; Garcia-zubia, 2021). Los laboratorios remotos pueden utilizarse tanto como complemento de los laboratorios prácticos tradicionales como en sustitución de los mismos (Brinson, 2015; Corter et al., 2007; Ma and Nickerson, 2006; Sauter et al., 2013). En ambos casos, facilitan el trabajo del profesor y de los alumnos.

También y por ejemplo, durante la pandemia de COVID-19, que ha hecho obligatoria la formación online o a distancia, el uso de laboratorios remotos ha permitido a los alumnos de algunos centros educativos continuar con sus prácticas de laboratorio casi con normalidad, experimentos STEM incluidos. Diferentes organismos públicos han incluido los laboratorios remotos en sus recomendaciones a los profesores (de Vries et al., 2017), posibilitando la continuidad de la enseñanza a distancia. Ejemplos de laboratorios remotos utilizados durante la situación de COVID-19 se pueden encontrar en EEUU (Pennisi, 2020), en España (Flaño, 2020; García-Zubía and Hernández-Jayo, 2020), en Alemania (Pretz, 2020) o en África (Linden, 2020; Sawahel, 2020).

La eficacia de los laboratorios remotos se ha comprobado repetidamente desde el punto de vista pedagógico (Ma and Nickerson, 2006; Brinson, 2015; Corter et al., 2007; De Jong et al., 2013; Hussein and Wilson, 2021; Nickerson et al., 2007). Se ha demostrado que, cuando se diseñan y usan adecuadamente, los laboratorios remotos pueden ofrecer resultados similares o incluso

superiores a los de los laboratorios convencionales. Además, ofrecen un amplio abanico de posibilidades a los estudiantes, como una mayor flexibilidad en sus horarios o la eliminación de las barreras geográficas (Batanero et al., 2019; Garcia-zubia, 2021). Esto ha provocado un aumento en el número de matriculaciones en las titulaciones que ofrecen metodologías de aprendizaje a distancia (Muzaffar et al., 2021; Nuci et al., 2021).

Los estudios anteriores han sido posibles gracias al uso extensivo de la experimentación remota durante los últimos años. Por ejemplo, el proyecto GoLab/NextLab (www.golabz.eu) ha desarrollado durante 8 años materiales educativos STEM donde la experimentación era siempre mediante laboratorios online. En la misma línea, el proyecto VISIR+ ha desplegado copias del laboratorio remoto VISIR en distintos países del mundo, ha desarrollado materiales educativos y ha comprobado el efecto positivo de VISIR en el aprendizaje de los alumnos.

En este momento, y no solo por la pandemia COVID-19, muchas universidades están comenzando a impartir títulos oficiales de grado online. Por ejemplo, la Universidad de Deusto comenzará en breve a impartir el Grado en Informática en modo online y por tanto son pertinentes dos preguntas: ¿es posible aprender utilizando un laboratorio remoto o un simulador? ¿qué características debe tener un laboratorio remoto para que facilite una respuesta positiva a la anterior pregunta? La primera ya ha sido respondida por las investigaciones ya nombradas, mientras que la segunda es objeto de esta tesis.

Ante esta situación de necesidad y oportunidad destaca el hecho de que los laboratorios remotos no son muy usados en educación. Tanto (Lahoud and Krichen, 2010; Orduña et al., 2016) analizaron la baja tasa de utilización de los laboratorios remotos en educación. (Orduña et al., 2016) afirmó: “Sin embargo, aunque el número de iniciativas de laboratorios remotos es elevado, el impacto global de estos laboratorios es bastante limitado más allá del ámbito de la institución anfitriona o del alcance (y duración) de los proyectos en los que esta participa. Hay casos en los que los laboratorios son utilizados regularmente por otras instituciones, pero siguen siendo excepciones y los laboratorios remotos aún no se utilizan de forma generalizada”. Es decir, a pesar de la utilidad de la experimentación remota, su uso real en las aulas no está actualmente muy extendido. Un hecho objetivo es que a menudo los experimentos solo son utilizados por los profesores que trabajaron en su desarrollo como investigadores, a pesar de que se ofrecen a otros profesores e instituciones.

(Lahoud and Krichen, 2010) analiza qué característica del laboratorio remoto es fundamental para un profesor y concluyen que es la fiabilidad. Los profesores, en general, no están dispuestos a planificar e integrar en su clase un recurso que falla o puede fallar, especialmente cuando no pueden estar seguros de saberlo de antemano y no pueden planificar fácilmente una alternativa en caso de que el laboratorio falle. Esta tesis pone en el centro la fiabilidad como medio para que la experimentación remota alcance un mayor reconocimiento y un uso real en educación.

La fiabilidad es una característica que es eminentemente tecnológica. Técnicamente, los laboratorios remotos hoy en día están muy desarrollados ya que pueden estar totalmente basados en la web, y por lo tanto pueden ser fácilmente accesibles para la mayoría de los usuarios (Aydogmus and Aydogmus, 2008; Garcia-Zubia et al., 2009; Sáenz et al., 2015; Yazidi et al., 2011). Además, existen sistemas de gestión de laboratorios remotos (RLMS) que pueden proporcionar características comunes y facilitar el desarrollo de nuevos laboratorios (Harward et al., 2008; Lowe et al., 2009; Orduña et al., 2011; Orduña et al., 2018; De Jong et al., 2013; Tawfik et al., 2013; Maiti et al., 2014; Schauer et al., 2014).

Gracias a los RLMS, los laboratorios se pueden compartir entre instituciones(Orduña et al., 2012) y pueden ser accesibles desde cualquier tipo de dispositivo, incluidos los móviles y las tablets (Crompton et al., 2017; Paravati et al., 2010; Gómez et al., 2014). En cuanto al diseño, se han creado arquitecturas que facilitan el escalado de los laboratorios y la adaptación de los mismos nuevos equipos reubicables (Angulo et al., 2018). La escalabilidad, como se verá más adelante, influye en la fiabilidad.

A continuación se profundiza en la fiabilidad y la escalabilidad como características exigibles a un laboratorio remoto profesional, ya que ambas son las protagonistas de los tres capítulos principales de esta tesis. En la Sección 2.2 se extienden los conceptos presentados hasta ahora y se alinean con la escalabilidad y la fiabilidad.

La última sección de este capítulo tiene como objetivo principal la descripción somera de la experimentación remota como campo de investigación. Si el lector no tiene conocimiento más o menos detallado de qué son, qué aportan y cómo funcionan los laboratorios remotos, estas últimas secciones pueden ser de ayuda. A cambio, si el lector conoce este campo, no es necesaria su lectura.

2.2 Fiabilidad y escalabilidad en experimentación remota

En esta sección se desarrollan con más detalle las dos características de los laboratorios remotos y objetos principales de esta tesis: la fiabilidad y la escalabilidad.

2.2.1 Laboratorios remotos y experimentación remota

En los últimos años, el número de laboratorios remotos accesibles desde Internet ha ido creciendo. Desde un punto de vista general, los laboratorios remotos pueden aportar muchas ventajas como el ahorro de costes, la democratización de la tecnología o la eliminación de barreras físicas y temporales (Nedic et al., 2003; Nickerson et al., 2007; Corter et al., 2007; Lindsay and Stumpers, 2011; Cordeiro et al., 2018; Lustig et al., 2018).

Múltiples instituciones han desarrollado estos laboratorios con el fin de proporcionar a los estudiantes y profesores la capacidad de enseñar, aprender e interactuar con estos dispositivos de forma remota.

Un ejemplo es RemLabNet[1], que ha desarrollado su propio Sistema de Gestión de Laboratorios Remotos (Schauer et al., 2014, 2016), y alberga múltiples experimentos remotos, permitiendo a los usuarios experimentar en campos como la física, la química o las ciencias ambientales.

Del mismo modo, iSES[2] ha desarrollado su propio sistema de enseñanza en línea (Schauer et al., 2006), que alberga una serie de laboratorios remotos con fines muy específicos en física. Entre ellos, es posible verificar el principio de incertidumbre de Heisenberg, o realizar el experimento de Franck-Hertz, y también, realizar experimentación con microcontroladores (Brom et al., 2018).

La *Grid of Online Laboratory Devices Ilmenau*, o GOLDi[3], da acceso a un laboratorio remoto con diferentes opciones de configuración. Aquí es posible seleccionar diferentes unidades de control (un CPLD, una FPGA, un microcontrolador, un FSM, etc.) y diferentes sistemas físicos, como un ascensor, una célula de producción o un almacén (Henke et al., 2016b). Esto permite al usuario tener una gran flexibilidad en la forma de realizar sus experimentos.

[1] http://www.remlabnet.eu
[2] https://www.ises.info/index.php/en/laboratory
[3] https://www.goldi-labs.net/

El Instituto Superior Técnico (IST) de la Universidad de Lisboa ha creado su propio proyecto de laboratorio remoto (Fernandes et al., 2010), conocido como *e-lab*[1]. Se trata de un portal web que da acceso a diferentes laboratorios remotos de física y química (Leal and Leal, 2013).

La Universidad Federal de Santa Catarina en Brasil también tiene su propio proyecto de laboratorios remotos (Da Silva et al., 2016), conocido como RExLab[2]. En él podemos encontrar varios laboratorios, como los de experimentación con cuadros eléctricos de corriente alterna y corriente continua, o los orientados a la ingeniería, como la programación de microcontroladores basados en placas Arduino (de Lima et al., 2016).

Otro proyecto significativo es LaboREM, desarrollado en el Instituto Universitario Tecnológico (IUT) de Bayona (Francia). Se trata de un laboratorio remoto (Letowski et al., 2019; Lavayssière et al., 2022) que permite experimentar a distancia con componentes electrónicos. A través de una cámara y un brazo robótico el usuario puede insertar diferentes placas de circuito impreso en distintas ranuras para crear una serie de circuitos electrónicos, que posteriormente pueden ser alimentados, caracterizados y medidos.

Otro enfoque similar es VirtualRemoteLab[3], desarrollado en la Facultad de Física de la Universidad Ludwig Maximilian de Munich, Alemania. Este laboratorio (Thoms and Girwidz, 2017) permite realizar experimentos de espectrometría óptica a distancia. Dado que el equipo del laboratorio es costoso y es necesaria una calibración precisa, han optado por proporcionar al usuario dos tipos de configuraciones experimentales: una real y otra pregrabada. El real permite una experimentación real, controlando el equipo en tiempo real, mientras que el montaje virtual permite experimentar con contenidos pregrabados del propio laboratorio.

Los laboratorios remotos descritos anteriormente no pretenden configurar una lista exhaustiva, pero son algunos de los trabajos de investigación que han hecho avanzar el estado del arte de los laboratorios remotos de forma muy significativa. Todos ellos presentan laboratorios remotos en tiempo real y, por tanto, proporcionan un control remoto en tiempo real del hardware objetivo. Sin embargo, como laboratorios en tiempo real, es probable que sean relativamente caros de desarrollar y mantener (Samuelsen and Graven, 2013; Villar-Martínez

[1] https://www.e-lab.ist.utl.pt/

[2] https://rexlab.ufsc.br/

[3] http://virtualremotelab.net

et al., 2021). Además, la mayoría de los laboratorios descritos admiten un solo usuario al mismo tiempo.

Aunque esta barrera se supera en algunos casos a través de la replicación (Esche et al., 2003; Villar-Martínez et al., 2019) o de medios específicos del laboratorio (Gustavsson et al., 2007).

Aparte del mencionado reto de la concurrencia, se han dedicado importantes esfuerzos de investigación hacia el desarrollo de arquitecturas que puedan ser aprovechadas para crear diferentes laboratorios remotos de forma más eficaz (Bermúdez-Ortega et al., 2015; Wang et al., 2016), y algunas de esas arquitecturas son de hecho la base de los laboratorios remotos descritos anteriormente (Henriques et al., 2015; Henke et al., 2016a,b; Fäth et al., 2018). En cualquier caso, y como se verá en el Capítulo 4, estos laboratorios remotos no son muy fiables en general.

A pesar de la existencia de todas las herramientas descritas anteriormente, el uso de los laboratorios remotos como herramienta sustitutiva o complementaria en las aulas aún no está extendido. Esto podría deberse, al menos en parte, a las limitaciones prácticas que aún persisten (Lahoud and Krichen, 2010; Jona et al., 2015; Lima et al., 2019; Sáenz et al., 2021), y que deberían resolverse o atenuarse para obtener un mayor uso de los laboratorios remotos.

2.2.2 **Sistema de gestión remota de laboratorios WebLab-Deusto**

WebLab-Deusto (Orduña et al., 2018a) es un Sistema de Gestión de Laboratorios Remotos (RLMS) creado por el grupo de investigación del mismo nombre[1] para facilitar la creación de laboratorios remotos.

Si se creara un laboratorio remoto desde cero, no solo habría que desarrollar los componentes específicos para cada copia del laboratorio en sí (como el control del hardware, el sistema de programación del microcontrolador o dispositivo electrónico, el sistema de cámaras en tiempo real o el cliente web), sino que también habría que desarrollar una serie de componentes que son comunes a muchos laboratorios remotos diferentes, independientemente de su naturaleza, como son:

- Gestión de usuarios y autenticación
- Gestión de grupos de usuarios

[1] https://weblab.deusto.es

- Autorización de usuarios
- Gestión de colas y flujos de usuarios
- Integración de los laboratorios en los sistemas de gestión del aprendizaje
- Compartir laboratorios entre diferentes entidades a través de la federación
- Extracción de datos y análisis del aprendizaje

Los sistemas de gestión de laboratorios remotos proporcionan la mayoría de esos componentes comunes, junto con otras herramientas para facilitar el desarrollo de los laboratorios. De este modo, los desarrolladores de laboratorios remotos solo tienen que centrarse en los componentes específicos de su laboratorio remoto, y pueden ahorrar miles de horas de desarrollo. Como consecuencia, el laboratorio resultante suele ser mejor y más fiable, ya que así los desarrolladores pueden utilizar su tiempo de forma más eficiente, y la mayoría de los componentes ya están bien probados en laboratorios anteriores.

Los laboratorios remotos creados bajo la arquitectura centrada en la escalabilidad que se describe en secciones posteriores basan una de sus capas en el WebLab-Deusto RLMS (Orduña et al., 2018a) como sistema de control y gestión del laboratorio. También se apoyan en WeblabLib (Orduña et al., 2019). Se trata de una librería de código abierto basada en Python, que forma parte de WebLab-Deusto, y que está específicamente pensada para facilitar el desarrollo de laboratorios remotos y su integración con WebLab-Deusto.

Varias iniciativas de laboratorios remotos en todo el mundo han utilizado WebLab-Deusto para sus proyectos. Este laboratorio de FPGA (Winzker et al., 2018; Winzker and Schwandt, 2019; Schwandt and Winzker, 2019), creado por la Biblioteca Universitaria y Distrital Bonn-Rhein-Sieg o el laboratorio LabRem de física/electrónica (Marchisio et al., 2015) creado por la Facultad de Ciencias Exactas, Ingeniería y Agrimensura Universidad Nacional de Rosario, son algunos ejemplos. Existen otros sistemas alternativos de gestión remota de laboratorios, que comparten algunas de las características, como (Benmohamed et al., 2005; Lowe et al., 2009; Yeung et al., 2010; Maiti et al., 2014; Schauer et al., 2014; Spilakova and Schauer, 2015; Schauer et al., 2015, 2016, 2017).

Esta tesis también utiliza WebLab-Deusto como base para su desarrollo, y será en ella donde se integre la arquitectura WebLabPRO a desarrollar.

2.2.3 LabsLand

LabsLand es una empresa que proporciona laboratorios remotos y también es el nombre de su plataforma de laboratorios remotos (Orduña et al., 2016; Orduña et al., 2018). LabsLand ofrece productos y servicios como suscripciones a su gran red mundial de laboratorios remotos, venta de equipos de laboratorios remotos que los clientes pueden desplegar en sus propias instituciones para integrarlos en la plataforma y desarrollo de nuevos laboratorios remotos. Estos laboratorios se desarrollan a menudo en colaboración con universidades. Los resultados obtenidos en esta tesis han sido desplegados en la plataforma LabsLand para validar su funcionalidad y eficiencia, de esta forma los diseños implementados huyen de ser simples resultados de investigación (pruebas de concepto) para convertirse en productos profesionales con uso real.

En este sentido la investigación presentada en esta tesis da lugar a una transferencia de resultados, denominada WebLabPRO. Cabe decir que LabsLand ha adaptado WebLabPRO a sus necesidades finales.

2.2.4 Escenario actual y desafío en escalabilidad y fiabilidad

Como se ha descrito anteriormente, los laboratorios remotos pueden ser tan eficaces desde el punto de vista educativo como los laboratorios prácticos (Ma and Nickerson, 2006; De Jong et al., 2013; Brinson, 2015), siempre que mantengan una determinada QoS (Quality of Service).

Cuando los laboratorios remotos se caen, tienen problemas de conexión, no están bien mantenidos o no pueden soportar un gran número de usuarios, no es posible mantener una QoS adecuada, lo que puede llevar a los usuarios e instructores a renunciar a los laboratorios remotos por otras alternativas como los laboratorios virtuales o los laboratorios prácticos tradicionales, aunque tengan desventajas para sus casos de uso. Los estudios realizados con profesores (Lahoud and Krichen, 2010) señalaron que la fiabilidad era la característica fundamental que se esperaba en un laboratorio remoto. Asimismo, Salzmann y Gillet (Salzmann and Gillet, 2007) afirman que:

> *Robustness toward hardware faults or unavailability is also a key issue for the acceptability of the remote experimentation paradigm by the students. If at connection time they are not able to access the chosen experiment, they may lose motivation and interest.*

> *(Traducido) La robustez frente a los fallos de hardware o la falta de disponibilidad es también una cuestión clave para la aceptación del paradigma de la experimentación a distancia por parte de los estudiantes. Si en el momento de la conexión no pueden acceder al experimento elegido, pueden perder la motivación y el interés.*

y que:

> *Remote laboratories maintenance is a difficult and time-consuming task when a 24/7 availability is targeted. The first step in providing a wide availability is to detect problems; this implies that the physical equipment and its associated software are capable of self-diagnoses.*
>
> *(Traducido) El mantenimiento de los laboratorios remotos es una tarea difícil y que requiere mucho tiempo cuando se pretende una disponibilidad 24/7. El primer paso para ofrecer una amplia disponibilidad es detectar los problemas; esto implica que el equipo físico y su software asociado sean capaces de autodiagnosticarse.*

Veamos de dónde procede esta falta de fiabilidad. Tradicionalmente, los laboratorios remotos se han derivado de prototipos de investigación. Como muestra la Figura 2.1, el objetivo fundamental de estos prototipos es que un dispositivo sea controlable a distancia, y nada más. Cuestiones como la calidad del servicio o la escalabilidad no se tienen en cuenta durante su proceso de diseño.

Algunas de estas pruebas de concepto evolucionan hasta convertirse en laboratorios de mayor calidad, que pueden tener o no múltiples instancias, y lograr una mayor estabilidad. El objetivo de estos laboratorios no es solo permitir el control remoto, sino también hacer que el laboratorio sea útil para los usuarios potenciales. Estos laboratorios suelen ser utilizados por profesores y estudiantes de la institución que los ha desarrollado y mantiene, aunque también se espera que alumnos de otras instituciones también lo hagan. Sin embargo, en estos casos tanto la calidad de servicio como la escalabilidad son limitadas, y los hacen inadecuados para un uso generalizado, abierto y profesional. (Sáenz et al., 2021) ha evaluado estas cuestiones con mayor detalle en su estudio, y afirma que:

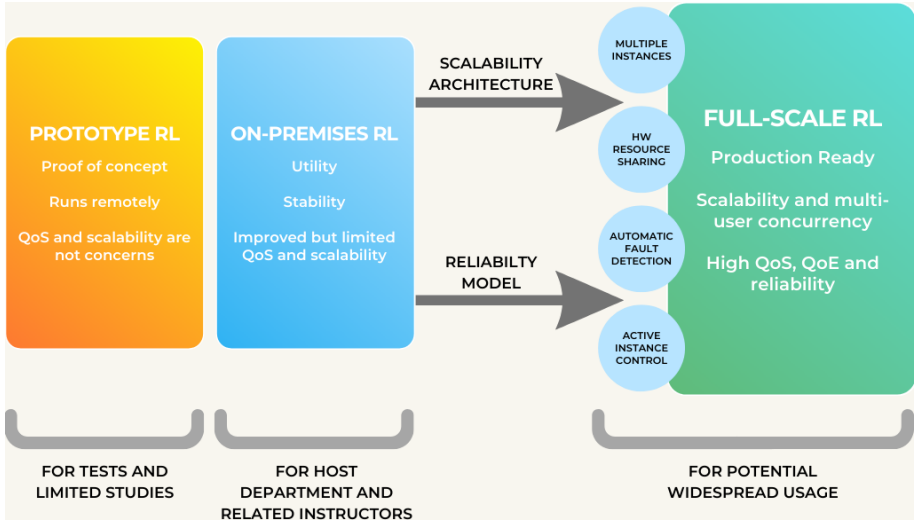


Figura 2.1: Categorización y evolución de los laboratorios remotos.

> *Due to the fact that there are no clear road maps or recommendations concerning the best selection of technologies and that different standardization attempts coexist, the development of online labs has become very challenging.*
>
> *(Traducido) Debido a que no existen hojas de ruta ni recomendaciones claras sobre la mejor selección de tecnologías y a que coexisten diferentes intentos de estandarización, el desarrollo de los laboratorios en línea se ha convertido en todo un reto.*

La escalabilidad es un concepto informático y así un laboratorio remoto es escalable si este facilita la inclusión de nuevas instancias de un mismo experimento remoto. El ideal de escalabilidad es el plug & play: simplemente el hardware de la nueva instancia se conecta al servidor y en ese mismo instante la nueva instancia ya está disponible para el usuario. La escalabilidad es una característica crucial para un laboratorio remoto, y en la Figura 2.1 marca el paso del primer al segundo nivel. Por ejemplo, si el experimento remoto es un robot controlado por microcontrolador y solo hay una instancia o copia del mismo, entonces es posible que se formen colas de usuarios para acceder a esa

única copia del robot, lo que reducirá la experiencia del usuario, sin embargo si hubiera 5 copias del robot, la cola de acceso se reduciría de una forma clara. Lo fácil y potente que sea añadir nuevas instancias o copias del robot, indicará cuánto escalable es un laboratorio remoto. Si añadir una nueva copia no es fácil, difícilmente se puede decir que ese laboratorio remoto es escalable.

Por otra parte, la fiabilidad de un laboratorio remoto nunca puede ser completa, aunque puede ser maximizada de manera que sus efectos no sean perceptibles por el usuario, ¿cómo? En primer lugar, diseñando cuidadosamente el laboratorio remoto, pero toda vez que los problemas no siempre se pueden anticipar, el segundo nivel de fiabilidad es investigar si un experimento remoto falla para no ofrecerlo al usuario y evitarle una mala experiencia, y en tercer lugar dando una solución a la falta de una instancia o copia del experimento.

El sistema de detección y solución del fallo debe ser automático si es posible, sin intervención humana, para que sea óptimo. Los fallos se detectan mediante test automáticos, y cuantos más sean estos, más fallos detectarán, sin perder de vista que el estado normal de un experimento remoto no es el de estar continuamente bajo test. En muchos casos la recuperación de un fallo consiste simplemente en un reseteo automático, pero en otros, por ejemplo se rompe un motor, no se puede hacer otra cosa que retirar esa instancia del experimento remoto. En esta situación la escalabilidad es de gran ayuda, ya que gracias a esta habrá más copias del experimento remoto y por tanto el usuario será redirigido a una copia activa del experimento, y con el mismo criterio se reordenará la cola de usuarios. La escalabilidad tiene entidad por sí misma, pero es de gran ayuda para mejorar la fiabilidad de un experimento remoto.

Los retos centrales de esta tesis son la escalabilidad y la fiabilidad y sus resultados se han sustanciado en dos artículos publicados en IEEE Access. Por un lado, (Villar-Martínez et al., 2019) aporta el diseño de una arquitectura de laboratorios remotos centrada en la alta escalabilidad a través de la replicación, el abaratamiento de costes y la compartición de componentes hardware entre diferentes instancias del laboratorio. Esta arquitectura está inicialmente orientada a la experimentación con microcontroladores, FPGAs y otros dispositivos programables.

Por otro lado, (Villar-Martínez et al., 2021) completa la anterior arquitectura de laboratorios remotos con un modelo basado en la detección de fallos para promover la fiabilidad de los laboratorios remotos. La arquitectura WebLabPRO utiliza una solución multicapa para evaluar el rendimiento de un

laboratorio, detectar fallos, resolverlos si es factible, desconectar la instancia que ha fallado en caso contrario, y habilitarla de nuevo cuando se repare y vuelva a funcionar. Todo esto se hace de forma automática y está orientado a garantizar la calidad de servicio aunque suponga desconectar instancias. Esto es importante ya que:

- Es mejor desconectar una instancia e impedir el acceso, a que el estudiante pierda tiempo y se frustre obteniendo resultados no validos
- El modelo de detección de fallos pretende complementar una arquitectura basada en la replicación por escalabilidad, en la que desconectar una instancia defectuosa implica simplemente que la carga se redirija a una que funciona.

En WebLabPRO la escalabilidad permite dar servicio a más alumnos/instituciones, y a su vez permite no cortar el servicio si en una instancia del experimento se detecta automáticamente que está fuera de servicio, ya que en este caso, y sin intervención humana, la instancia estropeada será retirada y su carga de trabajo con los alumnos será asumida por las otras instancias del mismo experimento. Si la instancia fuera única y estuviera estropeada, todo el experimento remoto quedaría fuera de servicio. Ambos diseños tienen fundamentos tecnológicos muy distintos, pero un fin común: la fiabilidad.

Estas soluciones, como se muestra en la Figura 2.1, permiten la creación de laboratorios remotos con capacidad para soportar un uso generalizado. Los laboratorios a gran escala tienen una mayor QoS, son fácilmente escalables y pueden ser utilizados en un entorno de producción por cientos o miles de usuarios simultáneamente. Emplean técnicas como las instancias múltiples, la compartición de recursos de hardware, la detección automática de fallos y el control interactivo de instancias.

2.3 Caracterización de los laboratorios remotos

Esta sección tiene objetivo introducir los conceptos generales de la experimentación remota, de manera que la tesis cuente con un escenario más completo y mejor descrito.

Un experimento remoto, RE, desde el punto de vista tecnológico está condicionado por una doble vertiente. Por un lado hay que caracterizar al propio

experimento remoto y por otro hay que hacer lo mismo con el Remote Lab Management System, RLMS. El RLMS ofrece al usuario el experimento remoto, y por tanto lo condiciona.

El RLMS y el RE son comparables al laboratorio y el experimento de un laboratorio clásico. Aunque el experimento diseñado por el profesor sea perfecto, si el laboratorio no está abierto, o no tiene condiciones de trabajo, etc., entonces la experiencia del alumno será negativa o incompleta.

2.3.1 Tipos de experimentos remotos

Primeramente es necesario relacionar experimento, experimento remoto y laboratorio remoto. Este último puede ofrecer al alumno más de un experimento remoto, o solo uno, en cuyo caso experimento y laboratorio se confunden. En esta subsección se caracterizan los experimentos en sí pero vistos ya como si fueran a ser remotizados o ya lo estuvieran, clasificándolos según distintos criterios.

En esta subsección se relatarán y clasificarán los experimentos remotos desde distintos puntos de vista, pero lo fundamental en esta subsección será clasificar los experimentos remotos desde la propia naturaleza de los mismos, de manera que los aspectos tecnológicos queden fuera de esta clasificación. En esta primera clasificación es más importante la palabra "experimento" que el adjetivo "remoto". En otras secciones se profundizará en la clasificación usando criterios tecnológicos, de usabilidad, etc.

Naturaleza del experimento bajo experimentación

Llevando la discusión a un campo más académico, se debe atender a la clasificación hecha por (Bencomo, 2002) y mostrada en la Tabla 2.1 para situar el laboratorio remoto. Inicialmente un laboratorio remoto ofrece al usuario completar un experimento real sin estar delante de él.

Arriba a la izquierda se encuentra el laboratorio hands-on donde el recurso es real y el alumno está frente a él. Debajo, el laboratorio remoto se distingue del anterior solo en que el alumno no está sentado frente al recurso, sino Internet mediante.

Un simulador reproduce la realidad utilizando para ello un modelo lógico-matemático programado en un computador. En un simulador es relevante la interfaz que use, ya que si esta es sofisticada (tipo realidad virtual o aumentada)

Tabla 2.1: Clasificación de los laboratorios remotos. Adaptado de (Bencomo, 2002).

		Naturaleza del recurso	
		Real	Simulado
Acceso al recurso	Local	Lab tradicional	Lab virtual monousuario
	Remoto	Lab remoto	Lab virtual multiusuario

Tabla 2.2: Extensión de la clasificación de los laboratorios remotos. Adaptado de (Rodríguez-Gil et al., 2017b).

		Naturaleza del recurso	
		Real	Simulado
Acceso al recurso	Local	Lab tradicional	Lab virtual monousuario
		Lab híbrido	
	Remoto	Lab remoto	Lab virtual multiusuario

puede que el alumno tenga la sensación de estar inmerso en la realidad, aunque ningún dato sea real. Esta tesis no aborda la simulación.

Los laboratorios remotos de esta tesis pertenecen a la categoría *Laboratorio remoto* de la anterior clasificación.

(Rodríguez-Gil et al., 2017a) rehace la propuesta inicial (Bencomo, 2002) para proponer la clasificación de la Tabla 2.2. En un experimento híbrido parte es real y parte es simulada.

Un ejemplo de laboratorio híbrido es el desarrollado en (Rodríguez-Gil et al., 2017b). Este experimento ofrece al alumno controlar mediante una FPGA (un tipo de controlador electrónico) programada en VHDL un depósito de agua que cuenta con una válvula, dos bombas y varios sensores de nivel. La FPGA es real y en ella el alumno carga un programa en VHDL escrito por él, este programa accede a los sensores del depósito de agua y en base a ellos controla el estado de las bombas, lo que modifica los sensores de nivel y por tanto puede provocar una nueva acción de la FPGA. En este experimento remoto, el depósito es simulado, no es real, no hay un depósito con miles de litros.

Esta solución no es real, pero es mucho más barata de crear y mantener, y

además no entraña riesgo de operación. En este caso la simulación ha sido de un equipo sencillo, pero podría ser un reactor nuclear o una planta industrial.

Este tipo de experimento remoto, ver Figura 2.2, destaca cuando lo más importante es la tarea del alumno para programar la FPGA (o para diseñar el controlador del depósito, central nuclear, etc.), y no lo es tanto el proceso a controlar.

Interacción del usuario con el experimento remoto

Según (Harward et al., 2008), la forma en la que el alumno interactúa con el experimento remoto puede ser de tres tipos: batch, interactivo u online y sensorizado, según sea la forma de obtener y analizar los datos, es decir, según sea la forma de interactuar con el experimento.

Este criterio puede ser analizado bajo dos puntos de vista:

- Cómo se obtienen los datos del experimento o cómo se interactúa con él:
  - Interactivo u online. El alumno interactúa continuamente con el experimento a través de la interfaz, cambia sus entradas y ve sus resultados en tiempo real. La interfaz y la webcam son críticas.
  - Batch. El alumno prepara el experimento en la interfaz y lo "lanza", pasado un tiempo (segundos o más) recibe los resultados. La webcam no suele ser importante ya que no hay información visual.
  - Sensorizado. El alumno simplemente monitoriza y analiza los datos que provienen de sensores, pero en ningún caso influye en la ejecución del experimento. Es un subtipo de online, pero sin interactuación.
- Cómo se analizan los datos del experimento:
  - Online o en tiempo real. En el mismo momento en que se generan los datos -visuales mediante webcam o numéricos mediante tablas- el alumno los analiza.
  - Offline o en tiempo diferido. Los resultados del experimento son analizados en un instante distinto al de su generación, pueden ser horas, días o años. Este tipo de experimentos reciben el nombre de Data Set.

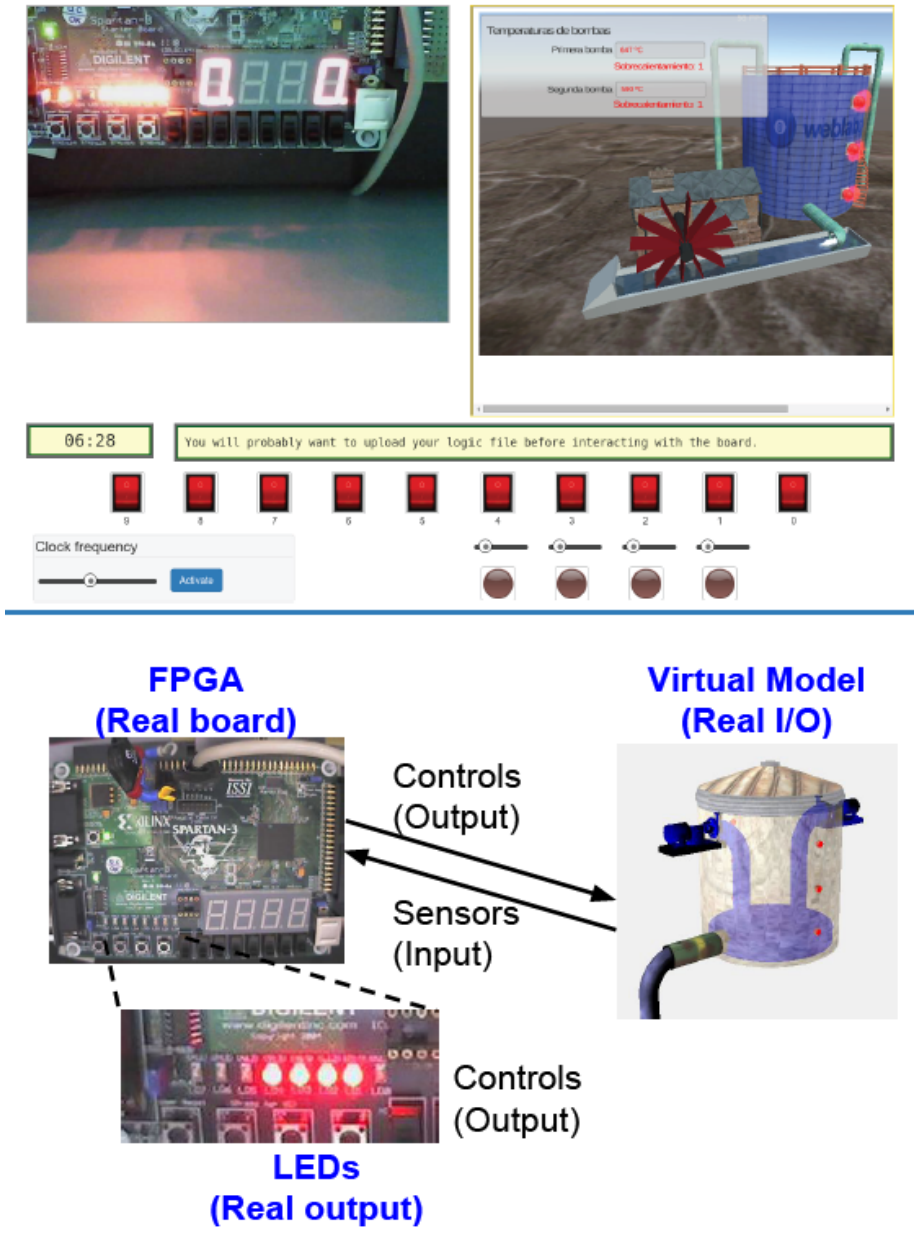


Figura 2.2: Ejemplo de experimento remoto híbrido.

Los laboratorios remotos contemplados en esta tesis son interactivos y online.

Observabilidad y controlabilidad del experimento remoto

Otra clasificación es aquella que atiende al tipo de entradas y salidas disponibles durante la experimentación remota. Las variables de entrada y salida pueden ser observables o controlables, es decir, se puede acceder a ellas y conocer su valor (observabilidad) y se puede acceder a ellas y cambiar su valor por métodos directos o indirectos hasta llevarlas a un estado conocido (controlabilidad).

- Las variables de entrada deben ser controlables por el alumno, y así este puede elegir qué valores toman en cada momento. Y por supuesto deben ser observables, el alumno debe conocer su valor a través de la interfaz del experimento remoto. Variables independientes de un experimento científico.

- Las variables de salida deben ser observables, es decir, el alumno debe poder ver en la interfaz los valores que toman dichas variables. Variables dependientes de un experimento científico.

- Las variables de salida serán controlables o no dependiendo del tipo de experimento.

Este enfoque coincide con la clasificación de (Müller and Erbe, 2007) que hablaba de Read/Write en función de las posibilidades de interactuación y control que ofrecía el experimento remoto. Dieter liga ambas características, read/write, con la seguridad y la complejidad del propio experimento tal y como se muestra en la Figura 2.3.

Por supuesto no todas las variables de entrada son controlables (write), ni todas las de salidas son observables (read), ni en un laboratorio remoto ni en uno normal.

Volviendo a las variables de entrada y de salida, las de salida deben ser observables ya que sino el experimento resultaría incompleto o incorrecto. Quedando la controlabilidad de las entradas como elemento de diseño. Según (García-Zubía et al, 2006) el experimento remoto en su diseño puede ser:

Remote Sensing ("read"): Acquiring information about system behaviour: - Remote observation - Remote monitoring - Remote measurement - Remote recording - ….	MINIMAL RISK AND SAFETY PROBLEMS
Remote Action ("write"): Changing of system behaviour from remote: - Remote parameter assignment - Remote programming - Remote manipulation of physical objects - ….	INCREASING RISK AND SAFETY PROBLEMS

Figura 2.3: Clasificación de experimentos remotos. Adaptado de (Müller and Erbe, 2007)

- No controlable. El alumno solo puede dar al botón de ON, solo puede lanzar el experimento y no puede modificar el valor de ninguna variable de entrada. El alumno ve cómo evoluciona el experimento y cómo cambian las señales de salida. Es similar al experimento sensorizado de la clasificación de Harward.

- Parametrizable. El alumno puede elegir para una variable de entrada un valor entre un juego de valores. Puede parametrizar el experimento, pero no puede cambiar su estructura. Por ejemplo elegir el valor de una resistencia, modificar la velocidad del motor de un robot móvil o elegir entre varios el valor de las constantes de un regulador PID. Una vez hecho esto, o cada vez que se cambian los parámetros, el experimento es como el anterior.

- Lógica o estructural. El alumno puede cambiar la estructura o la lógica del experimento, hacer varios experimentos distintos con uno solo. Por ejemplo, montar cada vez un distinto circuito eléctrico (VISIR), programar de forma libre un robot móvil o un Arduino o diseñar el algoritmo de un PID. Cada experimento puede ser totalmente distinto del anterior.

En esta clasificación, la controlabilidad de las variables de salida es el elemento relevante. En el primer caso el alumno solo puede ver lo que pasa, no puede modificar el fenómeno observado. Mientras que en el último caso el alumno puede crear o modificar el experimento, de manera que el comportamiento de las variables de salida estará controlado (o no) por él.

Los laboratorios remotos contemplados en esta tesis son observables y controlables en la medida de lo posible y están dentro de la categoría Lógica o estructural, ya que consisten en dispositivos programables tipo microcontrolador, FPGA, etc.

Experimento combinacional o secuencial

Si en un experimento remoto el comportamiento de la salida en el instante t solo depende del comportamiento de la entrada en dicho instante, entonces el experimento es combinacional: las distintas combinaciones de las entradas generan los posibles estados de las salidas. Su comportamiento se puede predecir a priori, sobre todo si el número de combinaciones de entrada no es muy alto.

Si en un experimento remoto el comportamiento de la salida en un instante *t* no solo depende de la entrada en ese instante, sino que también depende de la salida en el instante anterior, *t-1*, entonces el experimento es secuencial y pasa por distintos estados. En este caso es muy difícil predecir el comportamiento del sistema, excepto si el experimento pasa por pocos estados. Un sistema secuencial se comporta, en el mejor de los casos, como un autómata finito determinista.

Un experimento observable, no controlable, es combinacional en general. En realidad un sistema controlable puede ser combinacional, pero la estrategia de control es tan básica que difícilmente da lugar a un experimento remoto controlable didácticamente interesante.

Un experimento controlable es en general un sistema secuencial ya que el control se basa en conocer el estado actual del experimento, *y(t-1)*, para "llevarlo" a un estado *y(t)* deseado. El controlador puede ser un algoritmo software, o un regulador PID hardware, o un circuito electrónico cuyo conjunto de estados va a ser definido por el alumno y por tanto difícilmente se puede estimar el número de estados ni la relación entre los mismos. Así aunque el autómata es finito y determinista en su evolución, es imposible o muy difícil predecir su comportamiento a priori.

Los laboratorios remotos contemplados en esta tesis son secuenciales, ya que su controlabilidad es compleja y tiene forma de algoritmo.

2.3.2 Partes de un laboratorio remoto

En esta subsección se describirán las distintas partes de un laboratorio remoto, tanto desde un enfoque descriptivo como desde otro tecnológico.

Un experimento remoto permite a un usuario controlar vía web un experimento de carácter tecnológico o científico. La experiencia del usuario tiene dos fases:

- Acceder al experimento remoto. Remote Lab Management System, RLMS.
- Controlar y observar el experimento remoto. Remote Experiment, RE.

La primera fase no es obligatoria pero sí recomendable. El alumno debe introducir algún tipo de credencial para acceder a un recurso escaso. A veces el acceso es tan simple que ambas fases se confunden.

A la confusión anterior desde el punto de vista del usuario, se une el caos de algunos diseñadores que han seguido los siguientes pasos:

1. Primero montaron un experimento controlado electrónicamente,
2. luego lo remotizaron y
3. finalmente le añadieron una página de acceso y presentación.

Este enfoque bottom-up es propio de un diseñador amateur de experimentación remota, en el que generalmente predomina la habilidad hardware frente al diseño software. Sin embargo si se quiere que la experimentación remota esté en el aula del siglo XXI, entonces el servicio debe ser profesional. Los experimentos y su control son importantes (hardware) pero el modo en que se accede a ellos también lo es (parte software).

Un RLMS es el Moodle de los experimentos remotos. Si solo hay un experimento, entonces no hace falta, pero si hay varios experimentos entonces es muy importante. Por ejemplo, el acceso lo gestiona el RLMS, no cada RE; las colas de alumnos las gestiona el RLMS, no el RE; tener varios RE iguales lo gestiona el RLMS; compartir un RE entre varios centros educativos, lo gestiona el RLMS, etc. La idea es bien sencilla: todo aquello que sea común para cualquier experimento ofrecido, debe estar implementado en el RLMS.

Sin embargo un RE se encarga solo del propio experimento y de su remotización, aislándose de todo lo demás. Las partes fundamentales de un RE son:

- Sensores. Miden las señales de entrada y/o salida del experimento, suelen ser electrónicos.
- Actuadores. Controlan las señales de entrada y/o salida del experimento, suelen ser electrónicos.
- Controlador. Es el encargado de controlar el comportamiento del experimento mediante un algoritmo creado o no por el usuario. No siempre existe.
- Experimento. El péndulo, el robot, la planaria en el líquido, el circuito electrónico, etc. Es el objeto del experimento.

- Webcam. Ofrece al usuario una imagen del comportamiento del experimento. Puede ser visto como un sensor específico que no siempre existe.

- Interfaz web. Permite al usuario ver los valores de los sensores, el vídeo de la webcam, controlar los actuadores, modificar las entradas virtuales (botones, interruptores. . .) y cambiar el algoritmo del controlador, si este existe.

No es necesario que un RE tenga todos los elementos anteriores, pero son muy comunes. En muchos casos no hace falta el controlador, ya que no se quiere controlar un fenómeno sino simplemente verlo. Ya se ha comentado antes que en los laboratorios remotos con experimentos científicos los controladores no son habituales, mientras que en la mayoría de los laboratorios remotos tecnológicos (robots, Arduino...) el controlador es el elemento principal ya que la actividad del alumno consiste en crear un algoritmo. Estos laboratorios remotos son los que importan en esta tesis.

La Figura 2.4 muestra la interfaz del laboratorio remoto de una FPGA donde el alumno puede crear algoritmos que controlen motores mediante botones e interruptores.

2.3.3 Características y requisitos de los laboratorios remotos

Como ya se ha visto en la subsección anterior es necesario distinguir entre experimento remoto y laboratorio remoto. En el primero predomina la parte de control del experimento mediante un equipo hardware-software, mientras que en el segundo predomina la plataforma software que ofrece dichos experimentos remotos a los usuarios. A esta plataforma se la conoce como Remote Lab Management System, RLMS, y dentro de esta, la tesis se centra en la arquitectura que da acceso al experimento remoto, WebLabPRO.

Así pues la forma en la que se accede al experimento depende del RLMS, mientras que la experiencia depende directamente del experimento remoto. Históricamente se le dio más importancia al experimento que al RLMS, pero ahora mismo el RLMS condiciona totalmente la experiencia de usuario.

Los laboratorios remotos habrán alcanzado un aceptable nivel de aceptación cuando sean ofrecidos por los centros escolares, las universidades, las editoriales, etc., ya que eso supondrá que son necesarios y fiables, dando tanta

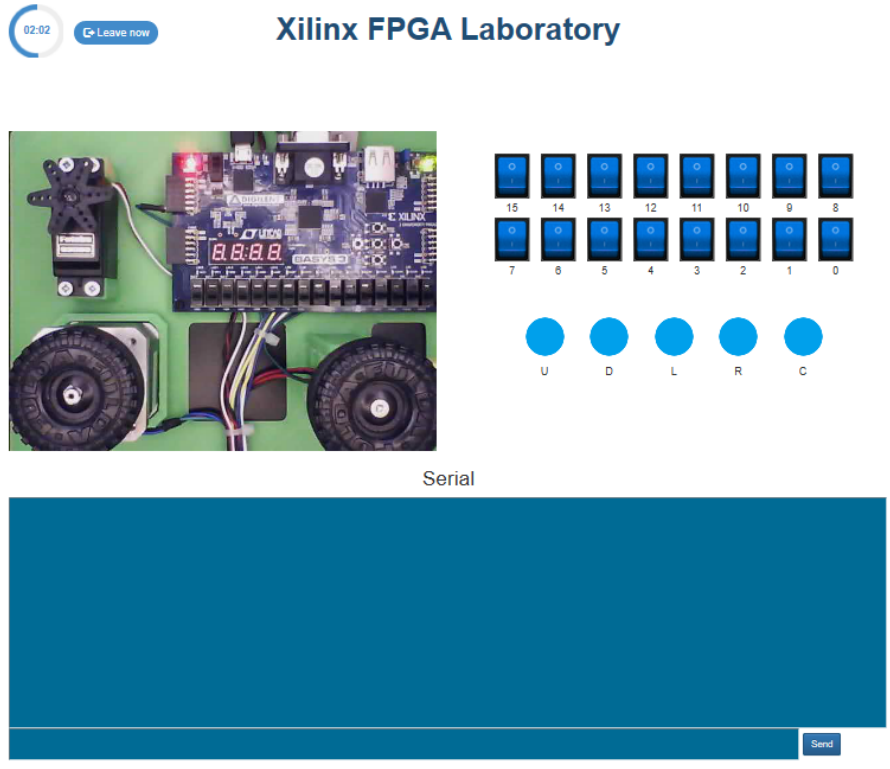


Figura 2.4: Experimento remoto con FPGA de LabsLand.

importancia a su necesidad como a su fiabilidad. Ninguna institución va a integrar en su día a día algo en lo que no confía del todo. Este aspecto es el objeto central de esta tesis: laboratorios remotos profesionales.

En esta subsección se van a abordar los requisitos exigibles a la arquitectura de acceso a los experimentos remotos para que sean profesionales. Las características/requisitos están descritas en detalle más adelante, pero antes tiene sentido agruparlas en cuatro conceptos:

- Universalidad, accesibilidad y fiabilidad. Describe si un laboratorio es accesible en cualquier escenario tecnológico del aula o centro educativo, o incluso fuera de él.
- Users management. Describe cómo acceden los usuarios al laboratorio remoto y el registro de su actividad.
- Learning support. Describe si el laboratorio remoto facilita el trabajo del alumno y del profesor en el aula.
- Escalabilidad y extensibilidad. Describe cuánto fácil es extender el laboratorio remoto añadiendo nuevos experimentos o replicando los ya existentes.

A continuación se describen con detalle cada una de las cuatro categorías anteriores que deben ser vistas como requisitos del diseño profesional de un laboratorio remoto:

- Universalidad y accesibilidad
  - Fiabilidad. El laboratorio está disponible y no está "fuera de servicio", "en mantenimiento", etc. Desgraciadamente es muy común que un laboratorio remoto después de ser diseñado deje de estar operativo en un tiempo de meses.
  - Universalidad. Un laboratorio remoto debe poder ser accedido desde cualquier tipo de computador, Tablet, Smartphone, etc. y bajo cualquier sistema operativo y usando cualquier navegador. Esta característica se explica por sí sola ya que los alumnos no quieren verse supeditados en exceso a diferentes tecnologías.

  - Instalación. Un laboratorio remoto debe ser accesible sin exigir al alumno o al profesor la instalación de ningún software adicional, ni de ningún plug-in, ni de ningún otro requisito que condicione su nivel de seguridad ni la política de su centro educativo (en muchos centros se prohíbe explícitamente la instalación de software externo).
  - Seguridad. El acceso de un laboratorio remoto en ningún caso debe comprometer la seguridad del centro educativo al exigir abrir ciertos puertos no seguros, al tener que retirar ciertos firewall, etc. La seguridad seguramente está por encima de la utilidad.
  - Eficiencia. Es crítico que el laboratorio remoto consuma la menor cantidad de recursos sin degradar la calidad del experimento que ofrece. El ancho de banda es el ejemplo más claro, cuanta menos se consuma mejor, ya que muchos centros educativos tienen restricciones en la misma. No olvidemos que el vídeo es parte fundamental del experimento remoto; es la forma en la que el alumno ve evolucionar su experimento.

- Gestión de usuarios

  - Gestión de usuarios y claves. Si un laboratorio remoto exige para entrar en él un usuario y clave, lo mejor es que el alumno lo pueda hacer usando sus claves del centro educativo, utilizando para ello un método seguro tipo LTI. No es en absoluto una buena práctica el pedir a los alumnos que se registren en un dominio nuevo y que usen distintas user/pass, siendo esta situación totalmente indeseable cuando los alumnos están por debajo de cierta edad. La gestión de usuarios es un elemento crítico y está más o menos resuelto tecnológicamente.
  - Gestión de colas. En un caso ideal, cada vez que un alumno quiere acceder a un laboratorio remoto no debería esperar, tendría siempre un experimento libre, pero en el caso real si solo hay un experimento para todos los alumnos de todos los centros, entonces es probable que tenga que esperar en una cola. La correcta y flexible gestión de las colas por parte del laboratorio remoto es una de sus características principales, y por supuesto el disponer de varias copias del mismo experimento mediante escalabilidad.

  - Recogida de datos y Learning Analytics. Cada vez es más importante que los profesores y el centro educativo puedan observar y conocer qué uso hace el alumno del experimento remoto: cuántas veces se conectan, cuándo, cuánto tiempo, qué hacen, etc. Los propios alumnos aprecian que el profesor pueda conocer su evolución. Los datos obtenidos serán objeto de análisis por parte del profesor.

- Escalabilidad y extensibilidad

  - Extensibilidad o adaptabilidad. Un buen laboratorio remoto no debe estar pensado para un solo tipo de experimento, sino para varios. Lo fácil que es integrar un nuevo tipo de experimento remoto es lo que describe la extensibilidad del laboratorio.
  - Escalabilidad. No es fácil entender la escalabilidad para un alumno o profesor, pero es crítica ya que determina la facilidad con la que se pueden añadir nuevas copias del mismo experimento remoto, es decir, habla de la facilidad con la que da servicio a más alumnos y centros educativos.
  - Federación. Si un experimento está replicado en varios sitios, por ejemplo en España e Inglaterra, puede que los alumnos de España solo usen el suyo, y los de Inglaterra solo el suyo, o puede que los federen. En este caso cada grupo de alumnos tendrá acceso a los dos experimentos, y no solo a uno, lo que incrementa su disponibilidad y relaja la cola de acceso. Técnicamente es una característica compleja de implementar y conlleva otras características como balanceo de carga, respuesta ante fallos, tracking de usos, etc., sin olvidar el acuerdo de uso que los socios deben firmar.

- Learning support

  - Plataformas de aprendizaje. Cada vez más los profesores utilizan LMS como Moodle, Google Classroom, etc. como plataformas para favorecer el aprendizaje de los alumnos. Es básico que un laboratorio remoto permita integrar sus experimentos en Moodle, por ejemplo, para que el profesor pueda organizar todo su material de trabajo en una sola plataforma (la del centro educativo), sin obligar al alumno a entrar en otras.

- Soporte didáctico. El profesor debe poder consultar y debatir la utilidad didáctica del experimento remoto accedido. Por ejemplo, puede pedir aclaraciones de uso o incluso tener dudas del propio hecho científico-tecnológico que debe probar. Un profesor jamás enseñará en clase algo de lo que no esté seguro, es excepcional que un profesor se arriesgue delante de los alumnos.
- Soporte técnico y mantenimiento. ¿Qué pasa si un robot se atasca en un punto porque ha sido mal programado? ¿qué pasa si se rompe un péndulo al lanzarlo? Toda vez que el experimento es real, las situaciones anteriores se pueden dar, y en este caso lo importante es que el proveedor del laboratorio tenga capacidad de respuesta: tenga varias copias del experimento, o cuente con un sistema de detección de fallos.
- Precio y sostenibilidad. La mejor manera de disponer de un servicio de calidad es pagando por él. Además de esta forma se asegura su sostenibilidad. Es normal que las universidades ofrezcan sus laboratorios remotos de forma gratuita, pero no suelen ser sostenibles en el tiempo. Está claro que no es un requisito técnico, ni mucho menos, pero condiciona toda la experiencia.

Las características principales que se abordan en esta tesis como fuente de requisitos son la fiabilidad y la escalabilidad, reforzando la sostenibilidad mediante soluciones de bajo coste.

2.4 **Resumen**

Los laboratorios remotos atienden a una casuística muy variable como han mostrado las características anteriores. De este conjunto, la tesis se centra en experimentos remotos basados en dispositivos programables con el objetivo de mejorar su escalabilidad y fiabilidad dando lugar a la arquitectura denominada WebLabPRO.

Para ello, las aportaciones de esta tesis son:

1. Descripción de un conjunto de laboratorios remotos creados bajo el conjunto de soluciones tecnológicas previamente introducidas.

2. Análisis cuantitativo realizado sobre un laboratorio remoto FPGA, que se basa en la arquitectura de alta escalabilidad y el modelo de alta fiabilidad desarrollados previamente.

3. Conclusiones, basadas en el análisis cuantitativo anterior, teniendo en cuenta principalmente la disponibilidad y la calidad del servicio prestado por el laboratorio desde el punto de vista del usuario potencial.

CAPÍTULO

3

Diseño, implementación y validación de una arquitectura de alta escalabilidad para experimentación remota

Los laboratorios remotos son una herramienta especialmente prometedora para una educación STEM eficaz. Ofrecen acceso universal a través de Internet a diferentes dispositivos de hardware en los que los estudiantes pueden experimentar y pueden poner a prueba y mejorar sus conocimientos. Sin embargo, la mayoría de ellos tiene una limitación importante, y es la escalabilidad, entendida como la facilidad que tiene una arquitectura para integrar nuevas copias de un experimento remoto, posibilitando el acceso concurrente por un gran número de estudiantes. Este capítulo aborda la escalabilidad para un laboratorio remoto en el ámbito de los sistemas programables.

3.1 Introducción

El uso práctico y masivo de los laboratorios remotos en diferentes instituciones y por parte de muchos estudiantes, tal y como se ha presentado en el Capítulo 2, plantea algunos retos adicionales importantes. En primer lugar, debe ser posible dar servicio a múltiples usuarios de manera simultánea. A menudo, los profesores requerirán que toda su clase, individualmente o en pequeños grupos, utilice los laboratorios al mismo tiempo. Asimismo, es posible que diferentes instituciones educativas necesiten utilizar el laboratorio también al mismo tiempo. En segundo lugar, el servicio de laboratorios remotos debe ser fiable. Los profesores confían en que el servicio estará disponible cuando lo necesiten, y cómo lo necesiten. Estos dos son los retos que aborda esta tesis de manera específica.

Estos retos pueden aliviarse parcialmente permitiendo la reserva de los laboratorios mediante calendarios. Sin embargo, para otros muchos laboratorios, la única forma práctica de superarlos es replicarlos varias veces, esto es, lograr tener múltiples copias de un mismo experimento remoto. El laboratorio remoto de robótica de LabsLand (Angulo et al., 2017), por ejemplo, permite a los estudiantes aprender robótica y programación Arduino con hardware real. Múltiples instancias del mismo robot se despliegan en varias instituciones, por lo que es posible dar servicio a varios estudiantes simultáneamente. Este es el objetivo fundamental de la escalabilidad: facilitar la replicación de un experimento remoto con nuevas instancias.

Al tener múltiples copias idénticas, la carga de trabajo puede distribuirse entre ellas, pudiendo así dar servicio a varios estudiantes simultáneamente. Además, también puede aportar fiabilidad, sobre todo en casos como el del laboratorio de robótica, en el que el robot cuenta con numerosas piezas mecánicas, susceptibles de romperse o desgastarse. Así, si una instancia de experimentación fallara, y existieran copias adicionales, el acceso al experimento remoto se mantendría garantizado.

El objetivo principal de este capítulo es proponer una nueva arquitectura mixta hardware-software, basada en otras ya existentes (Angulo et al., 2018), que pretende promover el desarrollo de laboratorios prácticos que se adapten al mencionado uso en el mundo real. Esta arquitectura recibe el nombre de WebLabPRO. La arquitectura debe servir de base para el desarrollo de laboratorios remotos multi-instancia, especialmente aquellos centrados en la experimentación de sistemas basados en microcontroladores y en la experimenta-

ción de sistemas embebidos, aunque también adaptable a otros ámbitos. Esta arquitectura tiene dos objetivos principales e interrelacionados. El primero es promover el desarrollo de laboratorios remotos que puedan ser replicados fácilmente, y que contemplen la replicación en su diseño inicial. El segundo es centrarse en la eficiencia de costes del laboratorio, que es un aspecto clave para los laboratorios que están destinados a ser replicados y a escalar a un gran número de usuarios.

Para evaluar la arquitectura escalable WebLabPRO se ha implementado un laboratorio remoto de experimentación con Arduino como ejemplo. Este laboratorio ha sido diseñado desde el principio para ser fácilmente replicado, para soportar múltiples usuarios concurrentes y ser de bajo coste. Éste, se ha utilizado para verificar que se cumplen los requisitos técnicos de la arquitectura, y se ha comparado con otras soluciones existentes en términos de escalabilidad y coste.

Este capítulo está organizado de la siguiente manera. La Sección 3.2 describe con más detalle el estado del arte sobre los laboratorios remotos y las arquitecturas relevantes. La Sección 3.3 analiza los objetivos y requisitos de la arquitectura propuesta. La Sección 3.4 presenta una visión general de la arquitectura propuesta. La Sección 3.5 describe cada capa con más detalle. La Sección 3.6 describe el laboratorio remoto para experimentación con Arduino que se ha creado para evaluar la arquitectura propuesta. Las Secciones 3.7, 3.8, y 3.9 evalúan la arquitectura desde tres puntos de vista; técnico, costes y escalabilidad, respectivamente. Finalmente, la Sección 3.10 resume las conclusiones y la Sección 3.11 describe futuras líneas de trabajo.

3.2 Estado del arte de la escalabilidad en experimentación remota

Los laboratorios remotos son una herramienta fundamental para la enseñanza de los sistemas embebidos, ya que permiten acceder a la experimentación con los dispositivos más novedosos sin necesidad de una actualización continua de los laboratorios clásicos. La plataforma Arduino es considerada por los expertos como la herramienta perfecta para abordar el desarrollo de sistemas embebidos básicos (Rodriguez-Sánchez et al., 2016; El-Abd, 2017). A pesar del bajo coste de los sistemas de desarrollo basados en esta plataforma, disponer de un laboratorio remoto proporciona al profesor una herramienta que ofrece

una abstracción total del hardware y facilita el acceso al ciclo de desarrollo de software que requiere la tecnología (Werner et al., 2016).

Actualmente existen múltiples de laboratorios remotos que proporcionan experimentación real con Arduino a través de Internet. Algunos de estos sistemas (Fotopoulos et al., 2016; Anzhelika et al., 2015; de Lima et al., 2016) están enfocados a permitir un acceso unitario, es decir, solo permiten dar servicio a un único usuario por sesión de experimentación. Esto limita el uso en clase, ya que generalmente, en este ámbito, seesperan múltiples accesos simultáneos.

Otros laboratorios se centran en el desarrollo de la experimentación remota de bajo coste (Alexander and Radhakrishnan, 2015; Mostefaoui and Benachenhou, 2015; Fernández-Pacheco et al., 2019), pero no se integran en un RLMS que proporcione características como la autenticación, la integración con el LMS, la federación o la gestión de colas y reservas, cuestiones que generalmente garantizan la escalabilidad. Todos estos sistemas se centran en la reducción de la infraestructura necesaria para el despliegue de los laboratorios, y a veces incluso proporcionan la capacidad de equilibrar la carga entre varias instancias del mismo laboratorio. Sin embargo, no proporcionan ningún medio particular para facilitar la replicación física de las instancias del laboratorio.

Algunos laboratorios, además, requieren que el alumno disponga del hardware para desarrollar su propia experimentación en casa (Sarik and Kymissis, 2010; Albiol et al., 2017). Estos laboratorios, conocidos como *pocket laboratories* o *pocket labs*, dan a los usuarios la oportunidad de experimentar en casa con dispositivos de hardware, a menudo más complejos, resolviendo la barrera de acceso físico, y aunque siguen siendo una buena alternativa para proporcionar desacoplamiento físico, no son extremadamente rentables, porque la relación entre el hardware del laboratorio y el usuario es de tipo uno a uno.

Desde el punto de vista arquitectónico, los laboratorios (Fotopoulos et al., 2016; Anzhelika et al., 2015; de Lima et al., 2016) presentan una distribución centralizada centrada únicamente en el propio experimento concreto, mientras que los laboratorios (Alexander and Radhakrishnan, 2015; Mostefaoui and Benachenhou, 2015; Fernández-Pacheco et al., 2019) han sido diseñados sobre una arquitectura distribuida centrada en la reducción de costes para favorecer la escalabilidad.

En este capítulo se propone una arquitectura enfocada a la escalabilidad que permita el despliegue del número deseado de instancias de experimenta-

ción a un bajo coste, para dar acceso simultáneo a un gran número de alumnos y no dificultar la capacidad de experimentación.

A continuación se analizan en profundidad dos de los laboratorios remotos de última generación y sus arquitecturas (Anzhelika et al., 2015; de Lima et al., 2016) con el objetivo de crear datos comparativos homogéneos, esto es, que todos los laboratorios permitan la experimentación con la plataforma Arduino. Estos laboratorios y sus arquitecturas son representativas del estado del arte de la experimentación remota con microcontroladores y sistemas embebidos. La comparación final indicará en qué aspectos y cómo la solución WebLabPRO aquí presentada se compara respecto a los dos laboratorios que constituyen el estado del arte: RELDES Arduino y RExLab Arduino.

Los dos laboratorios han sido seleccionados por múltiples razones. En primer lugar, el objetivo de ambos laboratorios es muy similar, permitir al usuario programar una placa Arduino de forma remota, lo que coincide con el objetivo de la arquitectura propuesta. En segundo lugar, ambos laboratorios y sus arquitecturas emplean *single-board computers*, tipo Raspberry Pi, compatibles con Linux, realizando la función de servidor de control, que también coincide con parte del sistema de control de la arquitectura WebLabPRO propuesta. En tercer lugar, ambos se han desarrollado teniendo en cuenta la eficiencia de costes.

En las siguientes subsecciones se presentan estos dos laboratorios remotos similares con capacidad de experimentación con tarjetas de desarrollo Arduino. Se proveen detalles técnicos sobre su arquitectura y sus implementaciones, con el fin de establecer la base de comparación. Estos laboratorios serán analizados y comparados con la implementación de este trabajo en las posteriores secciones de este capítulo.

3.2.1 **Laboratorio remoto RELDES Arduino**

El laboratorio remoto RELDES Arduino (Anzhelika et al., 2015) es un laboratorio remoto diseñado específicamente para el diseño y la experimentación de sistemas embebidos utilizando placas de desarrollo Arduino UNO como experimento remoto. Ha sido desarrollado por el Departamento de Herramientas de Software de la Universidad Técnica Nacional de Zaporizhzhya, en Ucrania. Desde el punto de vista arquitectónico, el laboratorio RELDES sólo tiene dos componentes principales, un servidor principal de laboratorio y los propios

experimentos, basados en placas de desarrollo Arduino.

El servidor principal de este laboratorio se encarga de gestionar todas las tareas del laboratorio remoto, que se describen a continuación:

- Servidor de acceso, para gestionar la autorización y el control de colas.
- Servidor Web, que soporta tanto el cliente web como un IDE integrado.
- Compilador.
- Programador.
- Servidor de transmisión de vídeo en tiempo real.

Este laboratorio tiene cuatro instancias de experimentación basadas en tarjetas de desarrollo Arduino. Estos componentes no están orientados a la escalabilidad, sino a ofrecer diferentes configuraciones de experimentación. Cada experimento utiliza las mismas placas de desarrollo Arduino UNO, solo que cada uno de ellos tiene un periférico de salida diferente. El laboratorio ofrece las siguientes configuraciones que se muestran a continuación:

- Arduino UNO + 4 diodos LED PWM
- Arduino UNO + servomotor
- Arduino UNO + pantalla HD44780
- Arduino UNO + sensor ultrasonidos HC-SR04
- Arduino UNO + pantalla HD44740

El usuario de este laboratorio remoto es capaz de programar una placa de desarrollo Arduino UNO en combinación con uno de los periféricos de salida descritos anteriormente. Una vez programado en el IDE integrado, el usuario monitoriza los resultados de la experimentación a través de un *stream* de video en tiempo real. Cabe destacar que, una vez programado, el usuario no dispone de medios para controlar el montaje de la experimentación. No hay conexiones entre el servidor y el experimento remoto, salvo un cable USB para realizar únicamente las tareas de programación.

3.2.2 Laboratorio remoto RExLab Arduino de RELLE

El laboratorio remoto RExLab Arduino de RELLE (de Lima et al., 2016) es un sistema de experimentación remota con Arduino desarrollado por la Universidad Federal de Santa Catarina, en Brasil.

El laboratorio, centrado también en la experimentación con Arduino, es ligeramente más avanzado desde el punto de vista de la arquitectura que su homólogo ucraniano. Este laboratorio está formado por tres componentes clave, que son el RLMS, el servidor del laboratorio y el *setup* de experimentación.

En este caso, todas las tareas de gestión de usuarios las realiza el RLMS, y sólo las tareas de control, programación y compilación las realiza el servidor del laboratorio. El laboratorio remoto ofrece dos instancias de experimentación iguales, implementadas mediante la replicación total del servidor del laboratorio y de los *setups* de experimentación, que están conectadas con el servidor RLMS vía Ethernet a través de una red de acceso local. En este caso, cada *setup* de experimentación está formado por una placa Arduino UNO y los componentes que se muestran a continuación:

- 4x diodos LED
- 1x servomotor
- 1x pantalla HD44780
- 1x potenciómetro
- 1x sensor de humedad
- 1x sensor de temperatura

Los usuarios, tras acceder al laboratorio a través de un sistema RLMS que gestiona el control de colas, accesos y autorizaciones, se presentan con un IDE integrado que permite la programación del microcontrolador tanto en lenguaje Arduino como en un lenguaje basado en bloques visuales.

Una vez programado, los usuarios pueden ver la instancia de experimentación a través de un *stream* en tiempo real. El usuario, en este caso, puede interactuar con el Arduino a través de una consola serie o mediante unos botones virtuales, controlados directamente a través de pines GPIO en el servidor del laboratorio.

3.3 Objetivos y requisitos de la arquitectura de escalabilidad

El objetivo principal en cuanto a la escalabilidad de la arquitectura WebLabPRO es satisfacer las necesidades previamente analizadas que están presentes en el estado del arte de las arquitecturas de laboratorio remoto, especialmente en las diseñadas para la experimentación remota con sistemas electrónicos programables.

El objetivo principal de este capítulo se ha dividido en varios objetivos y requisitos, que se explican a continuación. Los objetivos generales son:

- **Escalabilidad para el soporte de múltiples usuarios concurrentes:** La arquitectura debe diseñarse de forma que soporte un elevado número de usuarios que accedan a los distintos experimentos del laboratorio de forma concurrente. De esta manera, múltiples usuarios pueden experimentar con los experimentos integrados al mismo tiempo. En esta tesis, el término concurrente no se refiere a que múltiples usuarios accedan a un mismo experimento físico, sino que estos accedan simultáneamente a diferentes copias del mismo experimento, las cuales conforman el laboratorio remoto.

- **Adaptabilidad:** La arquitectura debe proporcionar los medios para interconectar fácilmente con una amplia gama de dispositivos electrónicos programables, como tarjetas de desarrollo con FPGAs, tarjetas de desarrollo con microcontroladores, sistemas robóticos basados en microcontroladores, etc.

- **Eficiencia de costes:** La arquitectura debe desarrollarse de forma que minimice los costes de despliegue de un laboratorio remoto, especialmente porque está diseñada para producir laboratorios multi-instancia, donde la mayor parte del hardware se replica muchas veces.

Los principales requisitos técnicos son los siguientes:

- **Experimentación de sistemas embebidos:** La arquitectura debe, como se ha descrito anteriormente, ser capaz de proporcionar una experiencia de experimentación real y remota sobre una variedad de plataformas embebidas.

- **Modularidad:** La arquitectura debe ser reconfigurable de manera que se adapte con éxito a diferentes escenarios de despliegue.
- **Universalidad:** La arquitectura debe, desde la perspectiva del usuario, ser accesible con independencia de la plataforma, el dispositivo o el sistema operativo, sin necesidad de reconfiguración de la red ni dependencia de software o plugins.
- **Fiabilidad:** La arquitectura debe mejorar la fiabilidad del laboratorio en su conjunto, como efecto secundario de la escalabilidad y la replicabilidad. Facilitar el despliegue de múltiples instancias de laboratorio iguales ayuda a mantener un grupo de instancias disponibles, incluso cuando uno o algunos de ellos fallan.

En las siguientes subsecciones se analizan de forma más exhaustiva los objetivos y requisitos de la arquitectura WebLabPRO.

3.3.1 Alta escalabilidad para soporte de múltiples usuarios

La experimentación de sistemas embebidos requiere que se programe físicamente un dispositivo embebido por cada usuario. Estos dispositivos embebidos son dispositivos finitos. Para dar soporte a un elevado número de usuarios concurrentes, como los de un entorno de aula, es necesario aumentar el número de instancias de experimentación disponibles, creando así un laboratorio multiinstancia. La arquitectura propuesta debe diseñarse de forma que pueda soportar con éxito un elevado número de instancias de experimentación iguales.

3.3.2 Adaptabilidad

En los últimos años ha surgido una amplia gama de dispositivos embebidos y plataformas de desarrollo embebidas (Liggesmeyer and Trapp, 2009), y cada una de ellas puede tener diferentes protocolos de comunicación y conectores. Esto aumenta la necesidad de una fácil adaptabilidad.

WebLabPRO debe proporcionar las técnicas de hardware y software necesarias para poder soportar esta variedad de sistemas de interconexión. Esta cuestión es fundamental puesto que, ante una actualización constante de los dispositivos integrados y las plataformas de desarrollo, la arquitectura debe proporcionar los medios para adoptar fácilmente la nueva tecnología con una

reconfiguración mínima y, cuando sea posible, evitar la necesidad de reimplementar el laboratorio desde cero.

3.3.3 **Eficiencia de costes**

Tradicionalmente, los laboratorios remotos han requerido el uso de costosos servidores y varios componentes de hardware diferentes para poder desarrollarse. Con el auge de los *single-board computers* (Maksimović et al., 2014) y otros dispositivos de hardware de bajo coste, el coste de despliegue de los laboratorios remotos se ha reducido.

WebLabPRO, además de ofrecer soporte para ser desplegada sobre dispositivos de hardware de bajo coste, también debería reducir los costes aún más al compartir recursos de hardware entre las múltiples instancias, con el objetivo de reducir significativamente el coste por instancia.

3.3.4 **Experimentación de Sistemas Embebidos y Usabilidad**

WebLabPRO debe proporcionar los medios para facilitar una verdadera experimentación remota sobre cualquier dispositivo o sistema embebido. Para garantizar que la experiencia de experimentación sea satisfactoria y no se vea obstaculizada es un requisito que la arquitectura proporcione una serie de formas de interactuar con el sistema embebido una vez éste se ha programado. Los usuarios deben poder programar sus scripts en el dispositivo, monitorizarlos a través de un stream en tiempo real e interactuar con ellos a través de diferentes medios, que pueden incluir entradas o salidas digitales y analógicas, control de periféricos de entrada y salida, y comunicación a través de diferentes protocolos.

3.3.5 **Modularidad**

Para adaptarse mejor a los distintos escenarios de despliegue, WebLabPRO debe diseñarse de forma modular. Para lograr este diseño modular, el núcleo de la arquitectura se subdivide en módulos o entidades más pequeñas, que se encargan de tareas más sencillas, y se interconectan a través de interfaces y protocolos de comunicación estándar del sector, como Ethernet, I2C, SPI, JTAG o USB. De este modo, los propietarios de laboratorios pueden seguir diferentes

diseños de topología para adaptarse mejor a los requisitos de configuración de sus laboratorios.

3.3.6 **Universalidad**

El significado de *universalidad* varía según el contexto. En esta tesis, el requisito de universalidad significa que los laboratorios remotos creados utilizando WebLabPRO deben ser lo más ampliamente compatibles (García-Zubia et al., 2009). Estos laboratorios remotos deben ser accesibles con independencia de la plataforma, dispositivo o sistema operativo, abriendo la posibilidad de acceder al laboratorio no sólo desde un ordenador, sino también desde tablets y smartphones. Para lograr este requisito de universalidad los laboratorios remotos deberían estar totalmente basados en la Web, y no deberían requerir ningún puerto de acceso diferente a los puertos comunes HTTP 80 y HTTPS 443, con el fin de eliminar los problemas de acceso derivados del uso de cortafuegos.

Además, para potenciar la universalidad, es aconsejable que la arquitectura sea integrable en diferentes Sistemas de Gestión de Aprendizaje o LMS, como Moodle, para poder integrarse sin problemas con otras herramientas basadas en web que puedan tener las instituciones educativas.

3.3.7 **Fiabilidad**

Debido a que un laboratorio remoto basado en WebLabPRO puede tener múltiples instancias del mismo *setup* de experimentación, la fiabilidad del laboratorio en su conjunto está directamente relacionada con el número de estas instancias que se encuentran en estado disponible.

El RLMS o Sistema de Gestión de Laboratorios Remotos, componente central de la arquitectura encargado de las tareas de administración del laboratorio, debe ser capaz de gestionar adecuadamente tanto las instancias de laboratorio activas como las que fallan, con el fin de mantener el mayor número posible de instancias de experimentación disponibles para su uso. En caso de que falle una instancia de laboratorio, el conjunto de los usuarios se distribuiría entre las instancias restantes en funcionamiento. Los tiempos de espera para acceder al laboratorio podrían aumentar en función del número de instancias funcionales restantes, pero el servicio se mantendría en línea. La fiabilidad es el objetivo del Capítulo 4.

3.4 Visión general de la arquitectura de escalabilidad

La arquitectura WebLabPRO propuesta ha sido diseñada con el objetivo de mejorar la escalabilidad de los laboratorios remotos de sistemas embebidos, con respecto al número de usuarios finales, con bajo coste y alta flexibilidad en cuanto a los dispositivos soportados.

Para conseguir una escalabilidad y una flexibilidad casi horizontales, manteniendo unos costes reducidos, la arquitectura se ha dividido en cuatro capas principales. de diseño desacoplado Esto favorece la facilidad de desarrollo e implementación y separa mejor las tareas de cada capa. La Sección 3.5 analiza estas capas desde una perspectiva de bajo nivel y con mayor nivel de detalle técnico.

La Figura 3.1 muestra una visión general de la arquitectura propuesta, que está desacoplada en cuatro capas principales, y sus principales componentes.

En primer lugar, la capa de nivel más baja de la arquitectura, que es la Capa de Hardware, engloba todos los dispositivos de hardware necesarios para gestionar adecuadamente el experimento del laboratorio remoto. Normalmente, un *single-board computer* con una serie de componentes hardware es capaz de interactuar físicamente con el experimento para replicar las acciones que el usuario realizaría en un laboratorio *hands-on*. El experimento remoto en sí, que es el núcleo del laboratorio remoto, también forma parte de la Capa de Hardware.

En segundo lugar, la Capa de Servidor de Interfaz es una entidad encargada de abstraer todas las particularidades relacionadas con el control y el hardware a través de una API REST simplificada. Su objetivo es almacenar y distribuir todas las tareas de control del laboratorio que representan acciones a aplicar en el experimento por los controladores de hardware existentes.

Estas tareas se reciben a través de un servidor web que expone una API REST mediante peticiones GET y POST, y se guardan en una estructura de datos FIFO. A continuación, las tareas se distribuyen secuencialmente a los controladores de hardware disponibles para ser ejecutadas. Tanto las acciones de almacenamiento como de distribución son realizadas por una base de datos en memoria o *message broker*. Como se puede ver en la Figura 3.2, esta entidad junto con la Capa de Hardware se puede dividir en tres subsistemas menores cada uno. Estos subsistemas se explican de forma más exhaustiva en el Sección 3.5.

En tercer lugar, la Capa de Servidor de Laboratorio, es un subsistema en-

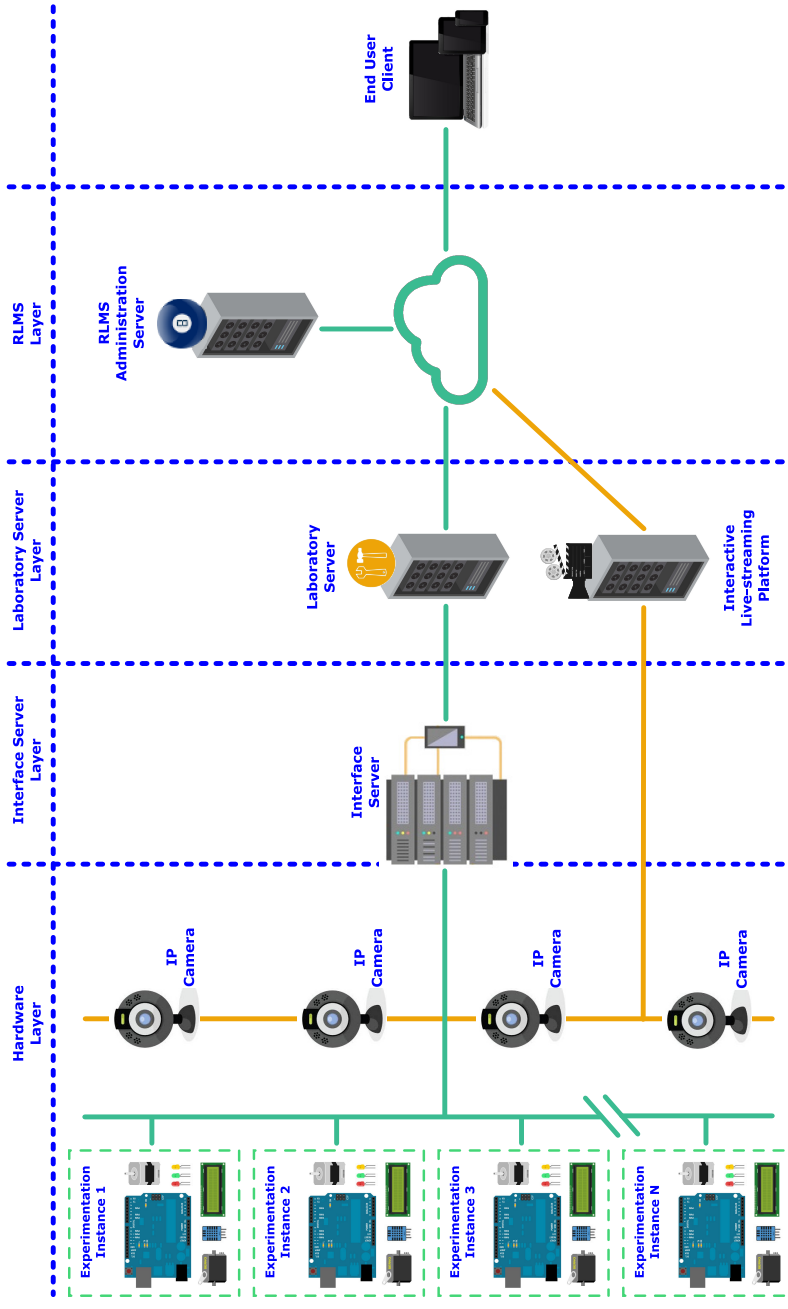


Figura 3.1: Vista general de la arquitectura WebLabPRO para escalabilidad.

cargado de soportar la interacción requerida entre el usuario y el experimento. Este componente soporta el cliente basado en la web que se muestra al usuario, el cual incluye un *stream* en tiempo real del experimento y los diferentes controles virtuales del laboratorio. Además, esta entidad se comunica con la Capa de Servidor de Interfaz para propagar aguas abajo las acciones del usuario y recabar el estado actual y la información del experimento.

En cuarto lugar, la Capa RLMS o Capa del Sistema de Gestión de Laboratorios Remotos es una entidad encargada de controlar todas las tareas de administración necesarias para gestionar adecuadamente las diferentes sesiones de laboratorio que puedan surgir. Estas tareas incluyen la autorización de usuarios, la gestión de reservas, la gestión de colas o el balanceo de carga, entre otras. También sirve como herramienta de integración, actuando como puente entre el laboratorio remoto y los diferentes sistemas de gestión de aprendizaje o LMS desde los que acceden los alumnos.

Por último se despliega una Plataforma Interactiva de Live-streaming de forma paralela a los otros cuatro subsistemas. Este componente, que también está formado por varias capas para garantizar la escalabilidad y la fiabilidad, se encarga de proporcionar al usuario del laboratorio remoto un *stream* en tiempo real, que es capturado in-situ por una cámara. Esta plataforma hace abstracción de las peculiaridades específicas de la cámara y proporciona estas funcionalidades de forma fiable. Esta plataforma es el resultado de la tesis doctoral de Luis Rodríguez Gil (Rodríguez-Gil, 2017).

En el espectro de los laboratorios remotos se pueden diferenciar al menos dos tipos de laboratorios. Los que requieren retroalimentación visual y los que no. Algunos laboratorios remotos, como los de procesamiento por lotes o *batch*, por ejemplo, ofrecen a sus usuarios la posibilidad de poner en cola sus trabajos y recuperar los resultados del laboratorio en un momento diferido. En estos casos, los resultados del laboratorio suelen ser capturados por diferentes sistemas de hardware y, por tanto, no es necesario transmitir una imagen en directo de la configuración del laboratorio.

Por otro lado, algunas configuraciones de laboratorios remotos requieren algún tipo de información visual, a menudo en tiempo real. Algunos de estos laboratorios utilizan periféricos de hardware de salida que son, por naturaleza, visuales, como diodos LED, displays de 7 segmentos, pantallas, servomotores o actuadores, entre otros. En estos casos, los usuarios necesitan poder observar el comportamiento de estos dispositivos para comprobar si el resultado del ex-

perimento es correcto o no. Este tipo de laboratorios suelen ser interactivos, es decir, el usuario puede controlar esos periféricos en tiempo real, en este caso, la arquitectura del laboratorio debe ser capaz de ofrecer al usuario una captura en tiempo real del comportamiento del laboratorio. El laboratorio remoto Arduino, que se explica en detalle en la Sección 3.6, es un caso de laboratorio remoto en el que es imprescindible la retroalimentación visual en tiempo real, debido al uso de periféricos visuales que son controlables a través del microcontrolador en tiempo real.

Como se puede observar en la Figura 3.1, la información generada por el usuario final fluye de derecha a izquierda hasta llegar al sistema embebido remotizado, y la información generada o procesada por éste, fluye de izquierda a derecha hasta ser mostrada o proporcionada al usuario final. La arquitectura sirve como medio de comunicación entre el usuario y el dispositivo embebido, proporcionando un desacoplamiento físico y manteniendo al mismo tiempo una capacidad de interacción de alta calidad, en tiempo real.

3.5 Capas de la arquitectura de escalabilidad

En esta sección se describirá con más detalle cada una de las capas y componentes que forman la arquitectura, y que se introdujeron brevemente en la Sección 3.4. Todas las entidades expuestas pueden manejarse como elementos desacoplados para crear la topología que mejor se adapte al escenario de despliegue. Puede apreciarse una vista en detalle de esta arquitectura en la Figura 3.2.

3.5.1 Capa de Hardware

La capa de hardware de la arquitectura abarca tres entidades arquitectónicas principales.

1. **Periféricos de salida:** Se encuentran en el nivel más bajo de la arquitectura. Estos periféricos son controlados exclusivamente por el dispositivo embebido y, por tanto, por el propio programa de prueba del usuario. Estos periféricos son opcionales, porque en algunos casos, ya están incluidos en las placas de desarrollo de componentes lógicos programables, y pueden ir desde simples LEDs o displays de 7 segmentos hasta periféricos más complejos, como pantallas o servomotores.

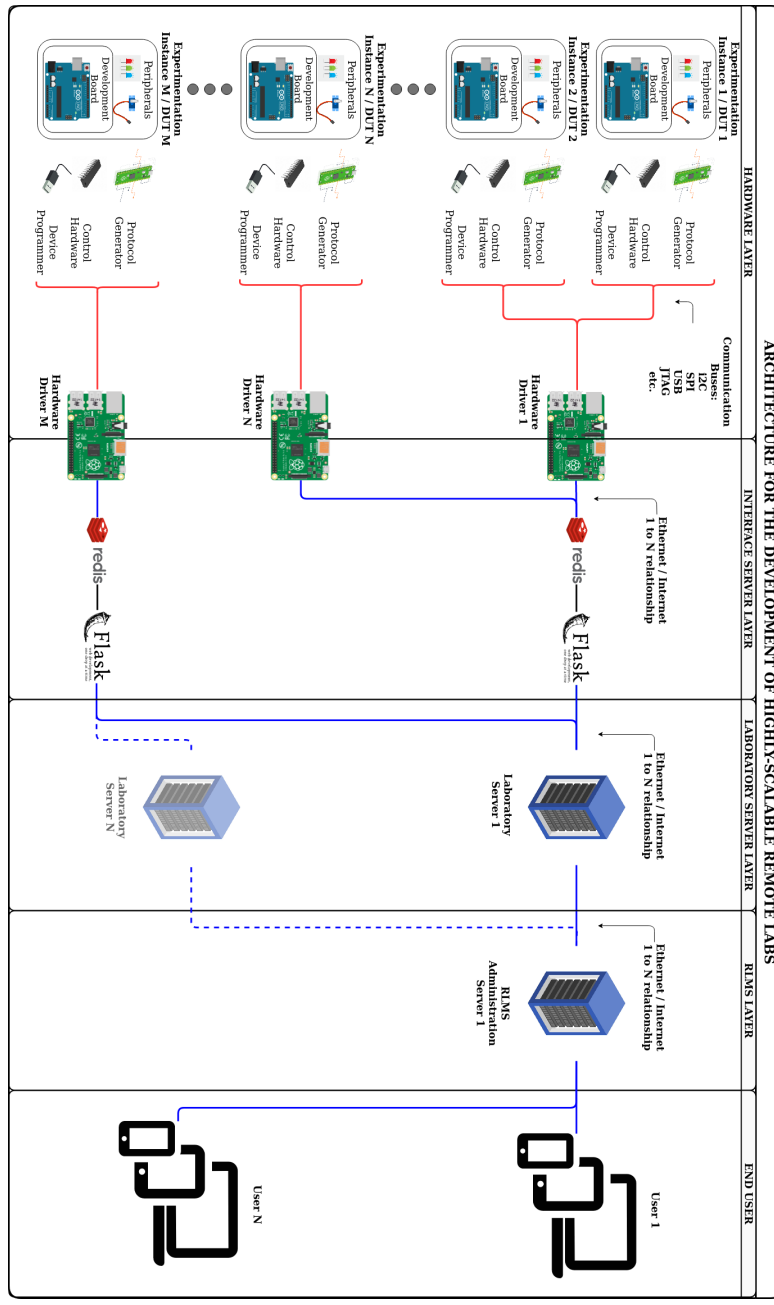


Figura 3.2: Vista en detalle de la arquitectura WebLabPRO para escalabilidad.

2. **Dispositivo Programable:** Los periféricos de salida descritos anteriormente son controlados directamente por el dispositivo programable. Este dispositivo embebido va integrado en una tarjeta de desarrollo que se adapta a las necesidades del usuario. Los dispositivos embebidos pueden variar enormemente para adaptarse a los diferentes requisitos de experimentación y aprendizaje, desde tarjetas de desarrollo basadas en microprocesadores hasta FPGAs. Estos dispositivos integrados, junto con los periféricos de salida, forman el *setup* de experimentación. El *setup* de experimentación forma una entidad que puede ser replicada N veces, con el fin de servir a usuarios concurrentes.

3. **Capa de Hardware de Control:** Para controlar, interactuar y comunicarse con el *setup* de experimentación es necesario gestionar las señales analógicas y digitales. La capa de hardware de control es una entidad, formada por varios dispositivos, que se encarga de interactuar eléctricamente con el *setup* de experimentación. Esta capa incluye diferentes circuitos integrados, como expansores de puertos, convertidores de digital a analógico, programadores JTAG, interfaces CAN-Bus, con el fin de introducir y capturar señales hacia y desde el *setup* de experimentación.

3.5.2 Capa del Servidor de Interfaz

Todos los experimentos remotos son controlados por la Capa del Servidor de Interfaz, una capa que se asienta sobre la Capa de Hardware, que también está formada por tres entidades independientes. La frontera que separa estas dos capas es un bus de interconexión, como SPI, I2C o CAN-Bus entre otros.

1. **Controlador Hardware:** El controlador de hardware es una entidad de software, que normalmente se encuentra en un *single-board computer*. Su propósito específico es convertir las peticiones del usuario que llegan a través del componente del *broker* de mensajes, en comandos de control que gestionan los dispositivos de la Capa de Hardware de Control. Estos comandos realizan cambios en el *setup* de experimentación. Por ejemplo, cuando un usuario quiere cambiar alguna señal de entrada digital, en una instancia específica, llega una tarea que describe esta acción a través del componente broker de mensajes. El Controlador Hardware interpreta esta tarea, y crea los comandos de control SPI necesarios para

cambiar el comportamiento de un determinado pin de entrada. Se puede desplegar cualquier número de controladores Hardware para gestionar diferentes buses de interconexión o cualquier número de instancias. En el caso de la comunicación hacia arriba, desde el *setup* de experimentación hacia los usuarios, el Controlador de Hardware trabaja de forma inversa, interpretando las diferentes señales y transmitiéndolas a través del componente *message broker*.

2. **Componente Message Broker:** Por encima de los controladores de hardware se encuentra el componente Message Broker. Esta entidad está formada por un almacén de datos en memoria de código abierto que, en este caso, actúa como estructura de datos FIFO y como transmisor de mensajes. Las órdenes del usuario se reciben a través de la API REST como tareas. Por ejemplo, *programar un fichero ejecutable*, *cambiar la salida de un pin a nivel alto* o *leer el valor de una entrada digital*. Estas tareas se encuentran en la estructura de datos FIFO, a la que se puede acceder a través de Ethernet en una red de acceso local como capa horizontal, gracias a las capacidades de intermediación de mensajes del *in-memory data store*. A continuación, las tareas se transmiten a los controladores de hardware y se realizan las acciones necesarias. El sistema también funciona en sentido contrario para la información de laboratorio que se necesita transferir al usuario.

3. **Servidor de Interfaz Web:** Para proporcionar un acceso unificado a través de comandos HTTP se necesita otra entidad. El Servidor Web de la Interfaz, que se encuentra sobre el componente Message Broker, es otra entidad de software que actúa como convertidor entre el *in-memory data store* y el mundo HTTP. El Servidor Web de la Interfaz expone una API REST para ofrecer un método de control basado en simples comandos HTTP. Este servidor web, mediante el uso de métodos GET y POST, interpreta y carga en la estructura de datos FIFO las tareas a realizar, y lee los estados de salida para propagar esta información a través de las capas superiores. Para cada acción se ofrecen diferentes *endpoints*, como por ejemplo */program*, para tareas de programación, */serial* para tareas de comunicación en serie, o */potentiometer* para tareas relacionadas con el potenciómetro virtual. A cada **endpoint** se accede mediante comandos GET y POST para ordenar un cambio o recuperar

información, respectivamente.

3.5.3 Capa de Servidor de Laboratorio

Sobre la Capa del Servidor de Interfaz se encuentra la Capa del Servidor de Laboratorio. Esta capa, que engloba dos entidades de software, se encarga de dar soporte a dos características importantes. Una entidad soporta el cliente basado en la web del laboratorio que incluye los controles de usuario, la consola de comunicación y la transmisión de vídeo en directo. Y otra entidad abstrae toda las particularidades de la comunicación entre el cliente basado en la web y la capa del Servidor de Interfaz.

1. **Cliente basado en la Web:** El Cliente basado en la Web, que es una entidad de software, se encarga de transmitir visualmente la configuración del laboratorio al usuario final. Este cliente puede incluir un *stream* en tiempo real, una utilidad de programación, una consola de comunicaciones, interruptores virtuales, botones, potenciómetros y diodos LED y diferentes opciones de depuración.

2. **Servidor Web de laboratorio:** El Servidor Web del Laboratorio es otra entidad de software, estrechamente acoplada con el Cliente basado en la Web. Este Servidor Web, que da soporte a la entidad anteriormente explicada, también actúa como sistema de control. Se encarga de transferir todos los comandos de los usuarios al Servidor Web de la Interfaz, y de recoger todos los datos de salida que son generados por el dispositivo embebido, para que sean mostrados en el cliente.

3.5.4 Capa del Sistema de Gestión de Laboratorios Remotos

La Capa del Sistema de Gestión de Laboratorios Remotos o Capa RMLS gestiona todas las tareas de administración que pueden surgir en un contexto de proveedor de laboratorios remotos. La Capa RLMS gestiona adecuadamente las sesiones de los diferentes usuarios concurrentes que pueden acceder a la plataforma del laboratorio remoto, y también gestiona las colas de espera, la autenticación, la reserva, el seguimiento de los usuarios o el balanceo de carga sobre varias instancias del mismo tipo de laboratorio. Después de conceder el acceso a los usuarios y asignarles una instancia de laboratorio, redirige a los usuarios al Servidor Web del Laboratorio.

3.5.5 Plataforma Interactiva de Live-streaming

Además de las capas descritas anteriormente es necesario desplegar una Plataforma Interactiva de Live-streaming, para proporcionar a los usuarios finales una visión en tiempo real de cada instancia del laboratorio. El desarrollo de este sistema queda fuera del alcance de esta tesis, que se centra en las tareas de control de los dispositivos embebidos. En la Sección 3.6, se da una explicación más profunda sobre la plataforma WILSP, que se ha utilizado en la implementación de la arquitectura.

3.6 El laboratorio remoto Arduino de LabsLand

Como prueba de concepto de WebLabPRO se ha desarrollado un laboratorio remoto que proporciona experimentación sobre la plataforma Arduino. Este laboratorio estña integrado en la red LabsLand, y se denomina comercialmente *Laboratorio Arduino de Labsland*. Este laboratorio remoto es altamente escalable y proporciona experimentación basada en la web sobre cuatro placas de desarrollo Arduino UNO [1], equipadas con microcontroladores Atmega 328P, microcontroladores AVR de 8 bits de bajo consumo que se utilizan con gran éxito tanto en entornos educativos como *do-it-yourself*.

Cada instancia de experimentación está formada por la mencionada placa de desarrollo Arduino UNO y una serie de periféricos, que incluyen diodos LED, potenciómetros, consola de comunicaciones serie, interruptores, botones, una pantalla OLED y un servomotor. Estos periféricos están preconfigurados y ya conectados de forma específica, ya que los usuarios no pueden conectar estos componentes por sí mismos debido a que el sistema es un laboratorio remoto. El esquema de la Figura 3.3 se entrega a los usuarios de antemano, para que puedan programar la placa Arduino en consecuencia. Los periféricos de entrada, como interruptores, botones y potenciómetros, se controlan de forma remota gracias al uso de la Capa de Hardware de Control, que se explica más adelante y que permite al usuario replicar esas señales eléctricas *in-situ* a través de Internet.

Cada instancia de experimentación de Arduino tiene una serie de capacidades con las que el usuario puede explorar:

- **Control de la señal de entrada analógica (4x)**

[1] https://store.arduino.cc/arduino-uno-rev3

  - Conectado cada uno a un potenciómetro como divisor de tensión.
  - Rango de tensión de 0 V a 3,3 V.
  - Control remoto a través de web-client.

- **Control de la señal de entrada digital (4x)**
  - Conectado cada uno a un interruptor y a un botón en paralelo.
  - Control remoto a través de web-client.

- **Control de la señal de salida digital (8x)**
  - 2 salidas no PWM conectadas a 2x diodos LED.
  - 2 salidas PWM conectadas a 2 diodos LED.
  - 3 salidas PWM conectadas a 1 diodo LED RGB.
  - 1 salida PWM conectada a 1 servomotor.

- **Control de comunicación Serie**
  - Accesible a través de una consola Serie similar a la de Arduino IDE.

- **Control de la comunicación I2C**
  - Puertos I2C conectados a la pantalla OLED SSD1306.
  - El usuario controla la pantalla a través de un programa de Arduino.

Estos periféricos son muy populares en el ecosistema Arduino, y suelen encontrarse en la gran mayoría de los kits de inicio de Arduino. El objetivo de este laboratorio remoto es ofrecer a los usuarios y estudiantes la posibilidad de experimentar como lo harían con un kit de inicio Arduino práctico, pero de forma remotizada, a través de Internet. Cada vez son más las escuelas y universidades que utilizan estos kits de iniciación para realizar prácticas en diferentes asignaturas, por lo que disponer de una versión remotizada y accesible a través de Internet, es algo positivo.

La flexibilidad del laboratorio está a la altura de la versión práctica de Arduino, por lo que el número de ejercicios y prácticas realizables es casi infinito, por lo que los profesores pueden adaptarlos mejor a los contenidos impartidos en clase.

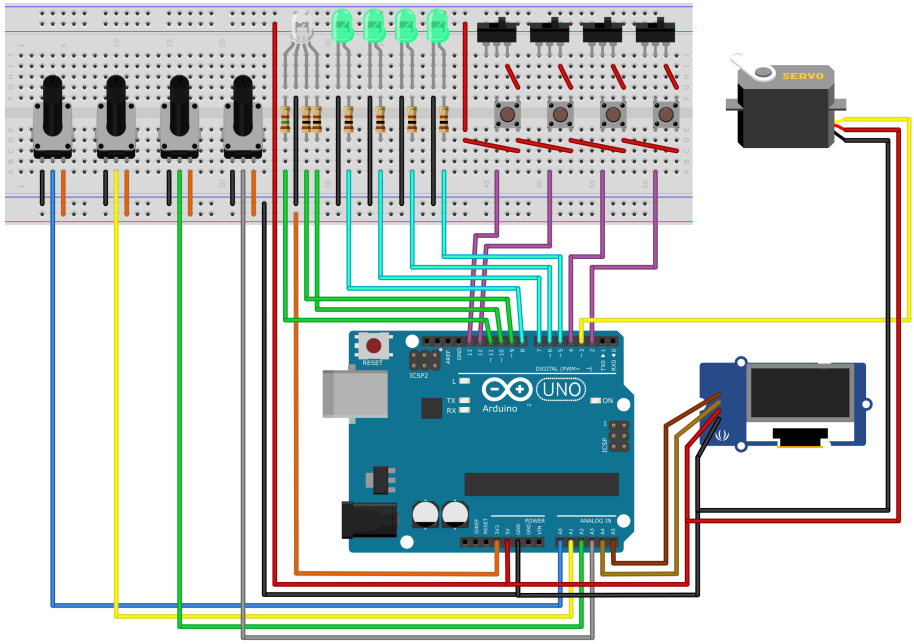


Figura 3.3: Esquemático Fritzing equivalente al laboratorio hands-on de Arduino.

La arquitectura escalable WebLabPRO se ha utilizado para desplegar el laboratorio remoto de Arduino de LabsLand y se ha integrado en su plataforma online de laboratorios remotos.

En las siguientes subsecciones se analiza la implementación del laboratorio remoto Arduino de LabsLand de acuerdo con las capas expuestas anteriormente en la Sección 3.5.

3.6.1 Capa Hardware

La Capa Hardware del laboratorio remoto Arduino de LabsLand está formada por tres entidades hardware.

1. **Periféricos de salida:** Cada una de las cuatro instancias incluidas en el laboratorio remoto Arduino de LabsLand tiene una cantidad fija de periféricos de salida, que han sido elegidos después de un análisis exhaustivo. Estos periféricos maximizan las capacidades de conexión de la placa Arduino UNO, utilizando todos los puertos de entrada y salida disponibles. Estos periféricos de salida son:

    - 1x diodo LED RGB con capacidades PWM.
    - 2x diodos LED con capacidades PWM.
    - 2x diodos LED sin capacidades PWM.
    - 1x servomotor.
    - 1x pantalla OLED SSD1306 compatible con I2C.

2. **Dispositivo embebido:** Como se ha explicado anteriormente, cada una de las instancias del laboratorio remoto Arduino de LabsLand tiene una placa Arduino UNO, en su versión SMD, como dispositivo embebido.

3. **Capa de hardware de control:** La capa de hardware de control del laboratorio remoto Arduino de LabsLand está formada por:

    - 8x convertidores digital-analógico de doble salida Microchip MCP4822 [1].

[1] https://www.microchip.com/wwwproducts/en/MCP4822

- 1x Expansor de puertos de dieciséis puertos de entrada/salida Microchip MCP23S17 [1].

Estos componentes, que están conectados a un bus SPI, se encargan de proporcionar 4 señales de entrada analógicas y 4 señales de entrada/salida digitales a cada instancia de Arduino. En total, manejan 32 señales.

3.6.2 Capa del Servidor de Interfaz

La capa de servidor de interfaz del laboratorio remoto Arduino de LabsLand está formada por tres entidades de software, que se describen a continuación. Estas entidades de software se despliegan sobre un *single-board computer* Raspberry Pi 3 Modelo B[2]. Se ha elegido este dispositivo porque tiene cuatro conexiones USB, soporta comunicación SPI, tiene suficiente potencia de cálculo para manejar las tres entidades de software, tiene conexión Ethernet y es de bajo coste.

1. **Controlador Hardware:** Esta entidad de software se ha desplegado como un programa de Python que maneja toda la gestión del hardware. Este programa de Python se conecta a la pila FIFO de Redis [3] a través de la red local. Este controlador gestiona las tareas de programación y la comunicación en serie directamente a través de la conexión USB y también gestiona la comunicación del bus SPI a través de la API SPI de Linux, y así controlar las cuatro instancias del laboratorio.

2. **Componente del *message broker*:** Para desarrollar la combinación de una estructura de datos FIFO y un *broker* de mensajes, se ha elegido Redis (Carlson, 2013), que proporciona un *in-memory data store* de código abierto que se ha utilizado con éxito en muchos proyectos conocidos para proporcionar escalabilidad (Alexeev et al., 2019; Chen et al., 2016; Banks et al., 2013). Se ha desplegado una instancia de Redis localmente sobre la Raspberry Pi 3. A esta instancia se accede a través de la conexión *loopback* tanto por el controlador hardware como por el servidor web de la interfaz.

[1] https://www.microchip.com/wwwproducts/en/MCP23S17
[2] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/
[3] https://redis.io/

3. **Servidor web de la interfaz:** Se ha desplegado un servidor web basado en Flask [1] también localmente en la Raspberry Pi 3. Este servidor web Flask expone una API REST capaz de controlar diferentes tareas para las cuatro instancias disponibles. Estas tareas incluyen para cada instancia:

    - Programar la instancia.
    - Restablecer la instancia.
    - Cambiar cualquiera de las 4 señales de entrada analógicas.
    - Cambiar cualquiera de las 4 señales de entrada digital.
    - Cambiar cualquiera de las 4 señales de salida digital.
    - Enviar un comando serie al Arduino.
    - Leer un comando serie desde Arduino.

La Figura 3.4 muestra la implementación del laboratorio remoto Arduino de LabsLand. La figura muestra una PCB, que soporta e interconecta todos los componentes de la Capa de Hardware con la Capa de Servidor de Interfaz, que se ha implementado sobre un *single-board computer* Raspberry Pi 3.

3.6.3 Capa del Servidor de Laboratorio

La capa de servidor de laboratorio remoto Arduino de LabsLand está formada por las dos entidades de software que se describen a continuación. Estas entidades se despliegan sobre un servidor Dell ProLiant. Este servidor también es utilizado por otros despliegues anteriores del laboratorio, y se ha utilizado sólo porque ya estaba conectado a la Red de Acceso Local, y tenía recursos disponibles. En caso de un despliegue masivo de un laboratorio basado en esta arquitectura, puede ser preferible un servidor aislado. Las entidades de esta capa son:

1. **Cliente basado en la web:** Se ha desarrollado un cliente basado en la web para el laboratorio remoto Arduino de LabsLand. Como se puede ver en la Figura 3.5, el cliente tiene diferentes controles digitales y analógicos, una consola de comunicación en serie y una captura en vivo de la configuración del laboratorio. Además, se ofrece a los usuarios un

[1] http://flask.pocoo.org

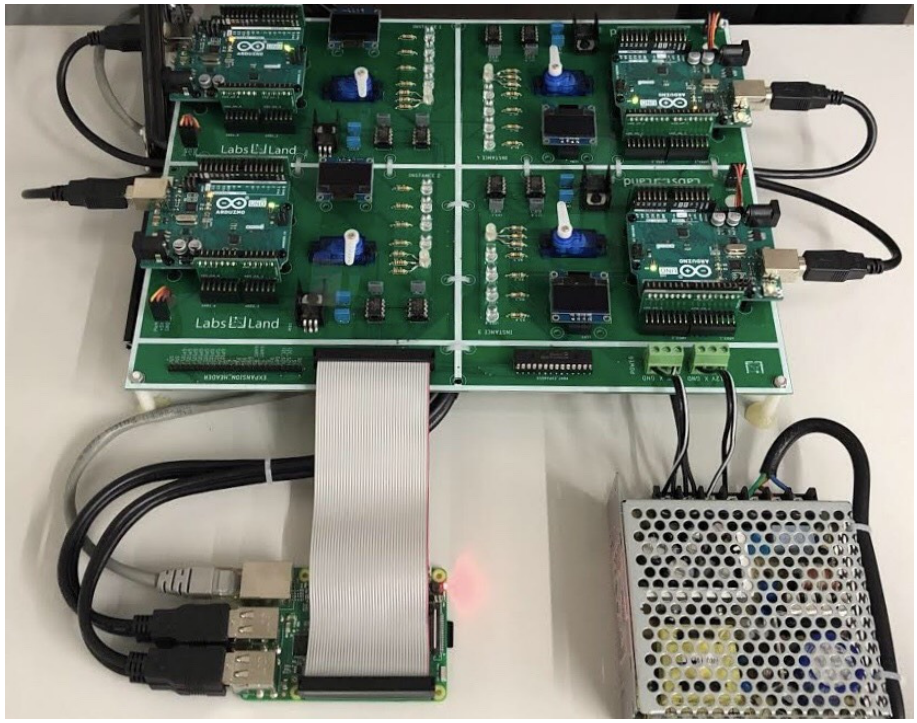


Figura 3.4: Implementación de los componentes físicos del laboratorio Arduino de LabsLand

IDE online, en el que pueden crear sus propios ficheros de programación de Arduino utilizando el Lenguaje Arduino o un lenguaje de programación basado en bloques que utiliza la librería Blockly [1] de Google. Los lenguajes de programación basados en bloques son una nueva tendencia educativa que puede mejorar el aprendizaje (Weintrop and Wilensky, 2017). Una vez que el código está listo, el cliente se encarga de la etapa de programación, en la que los usuarios seleccionan el programa adecuado. La Figura 3.6 muestra el IDE integrado y basado en la web.

2. **Servidor web de laboratorio:** Otro servidor web basado en Flask ha sido desarrollado para manejar las herramientas basadas en la web de cada usuario. Este servidor web soporta los dos IDEs integrados, el cliente de laboratorio y todas las peticiones de interconexión que se producen entre esta capa y la capa de servidor de interfaz. Para gestionar las cuatro instancias de laboratorio disponibles se han desplegado cuatro instancias virtuales de este servidor web sobre el servidor físico de Dell.

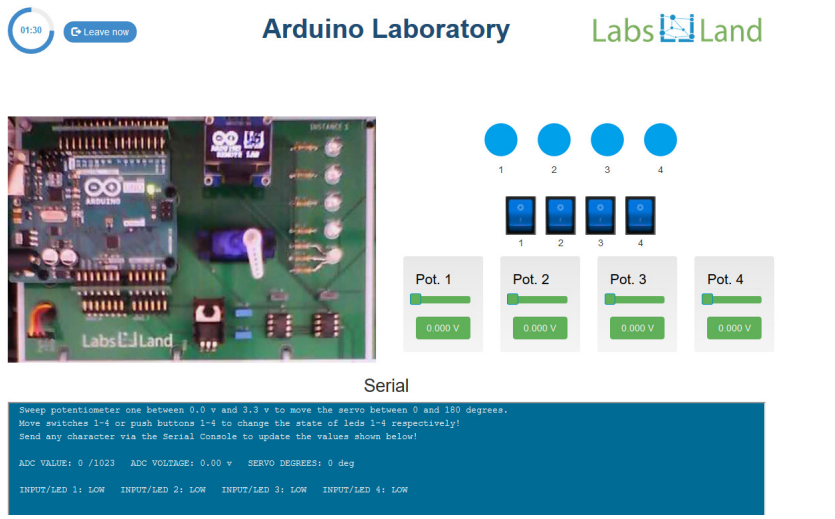


Figura 3.5: Panel de control del laboratorio remoto Arduino de LabsLand

[1] https://developers.google.com/blockly/

Figura 3.6: Entorno de desarrollo de código Arduino del laboratorio remoto Arduino de LabsLand

3.6.4 Capa del Sistema de Gestión de Laboratorios Remotos

El laboratorio remoto Arduino de LabsLand se ha integrado en la plataforma de laboratorio remoto LabsLand. Esta plataforma es compatible con el Sistema de Gestión de Laboratorios Remoto WebLab-Deusto (Orduña et al., 2018a), por lo que la integración entre laboratorio y plataforma ha sido sencilla. El uso de la librería *weblablib* (Orduña et al., 2019) simplifica el proceso de integración. Este sistema RLMS es totalmente integrable en los sistemas LMS de primer nivel que utilizan casi todas las instituciones educativas que acceden a estos laboratorios. Además, WebLab-Deusto es compatible con otras plataformas basadas en la web, con el fin de mantener el acceso universal. Estos resultados conformaron la tesis doctoral de Pablo Orduña Fernández (Orduña, 2013)

3.6.5 Plataforma Interactiva de Live-streaming

Se ha desplegado una Plataforma Interactiva de Live-streaming, denominada WILSP (Rodríguez-Gil et al., 2017a), en paralelo con cuatro subinstancias para capturar en tiempo real el comportamiento de cada instancia de laborato-

rio, y transmitir esa imagen con el mínimo retardo, en una gama de formatos disponibles, al web-cliente. Esta entidad transversal se explica de forma resumida a continuación, ya que afecta a la escalabilidad del laboratorio en su conjunto. Estos resultados provienen del trabajo doctoral de Luis Rodríguez Gil (Rodríguez-Gil, 2017), como ya se ha indicado anteriormente.

La plataforma WILSP

Un componente importante de muchos laboratorios remotos es la transmisión interactiva en directo proporcionada por una cámara web (Jara et al., 2008; Yazidi et al., 2011; Vargas et al., 2013). Mientras que en un laboratorio presencial los usuarios observan a través de sus propios ojos, en un laboratorio remoto lo hacen a través de esta cámara y utilizan esa información para interactuar con el laboratorio.

Por eso, para ofrecer una experiencia de usuario satisfactoria este *stream* debe cumplir ciertos requisitos. Los requisitos más importantes son, en primer lugar, que esté totalmente basado en la web (no se necesitan plugins ni puertos no estándar) y, en segundo lugar, que la latencia sea baja. En concreto, el *stream* debe tener una latencia lo suficientemente baja como para ser considerado interactivo. Esta latencia, también conocida como retardo de *capture-render*, debe permitir a los usuarios interactuar con el equipo percibiendo su respuesta en tiempo casi real.

Para satisfacer estos requisitos, esta arquitectura hace uso del Open Source WILSP (Rodríguez-Gil et al., 2017a) Interactive Live-Streaming Server que ha sido diseñado para ser eficaz en laboratorios remotos.

Integración de WILSP y el laboratorio remoto Arduino de LabsLand

En línea con el enfoque de ahorro de costes de la arquitectura propuesta en este capítulo, se han añadido mejoras adicionales sobre la arquitectura estándar basada en WILSP. La más significativa de ellas es la capacidad de compartir una única cámara web para más de una instancia del mismo laboratorio remoto.

Las cámaras IP independientes son relativamente caras. Por ejemplo, una D-Link DCS-2132L, utilizada en el laboratorio remoto de robótica LabsLand Arduino (Orduña et al., 2018) puede costar unos 150 euros. Esto puede ser una fracción significativa del coste total de la instancia de laboratorio remoto, teniendo en cuenta los precios de otros componentes (por ejemplo, un Rasp-

berry Pi puede costar alrededor de 40 euros, y una placa Arduino alrededor de 25 euros). Véase la Sección 3.8 para más detalles.

La plataforma WILSP se ha ampliado para poder servir *streams parciales*. De esta manera, la arquitectura propuesta en esta tesis puede utilizar una sola cámara para apuntar a dos instancias simultáneas. El hardware del laboratorio remoto Arduino que se propone como ejemplo de implementación también ha sido diseñado a propósito para este fin, disponiéndolo de forma que se aproveche la amplia relación de aspecto de las cámaras web modernas. Así, sólo se necesitan dos cámaras web para dar servicio a cuatro instancias diferentes.

La Figura 3.7 muestra el *stream* original proveniente de la cámara IP física. Puede apreciarse que el *stream* original cubre dos instancias diferentes del laboratorio remoto. El área alrededor de A es una de las instancias, mientras que el área alrededor de B es la otra. El *stream* es recibido por el WILSP modificado, que es capaz de cortarlo, rotarlo, recodificarlo y dividirlo, para servirlo como *streams* diferentes para cada instancia del laboratorio.

De este modo, es posible obtener una importante reducción de costes sin reducir significativamente la calidad de imagen percibida del laboratorio.

3.7 Evaluación técnica de la arquitectura de escalabilidad

En esta sección se utilizará la implementación de la prueba de concepto descrita en la Sección 3.6 para verificar que la arquitectura cumple con los requisitos técnicos propuestos inicialmente en la Sección 3.3; y se realizará una comparación con trabajos y arquitecturas similares del estado del arte teniendo en cuenta las características que ofrece cada arquitectura. La evaluación realizada a través de la prueba de concepto sugiere que se han cumplido dichos objetivos y requisitos.

3.7.1 Objetivos

Los objetivos propuestos fueron los siguientes:

- Alta escalabilidad para el soporte de múltiples usuarios concurrentes
- Adaptabilidad

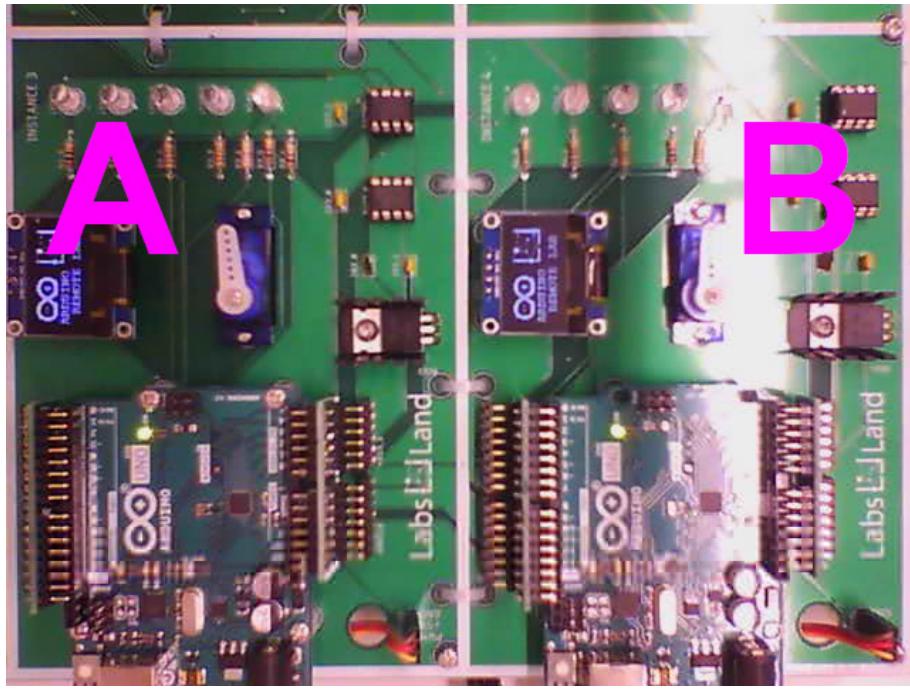


Figura 3.7: Vista en detalle de la captura real de la cámara web del laboratorio Arduino

- Eficiencia de costes

El objetivo de escalabilidad de la arquitectura se ha cumplido satisfactoriamente, gracias al uso de herramientas de alta escalabilidad, como Redis, y al uso de un diseño desacoplado combinado con diferentes protocolos, esquemas y dispositivos de comunicación. Un análisis más profundo de esta cuestión se explica en la Sección 3.9.

La arquitectura WebLabPRO, desde el punto de vista de la interconexión, puede considerarse que ofrece una gran adaptabilidad. Se han realizado esfuerzos para crear un Servidor de Interfaz capaz de conectarse a diferentes sistemas embebidos a través de buses de conexión totalmente establecidos como I2C, SPI o serie. La mayoría de los dispositivos de desarrollo embebidos de última generación pueden ser programados con estas interfaces, ya que los fabricantes suelen incorporar recursos de programación universales en las placas de desarrollo. Además, las diferentes señales analógicas o digitales necesarias para controlar el sistema embebido una vez programado, y los periféricos de salida seleccionados son tolerantes a CMOS y TTL, por lo que, una vez más, se fomenta la adaptabilidad.

En términos de rentabilidad, la arquitectura también es satisfactoria. Gracias a un diseño minucioso y a la replicación parcial del hardware, así como a la posibilidad de desarrollar el laboratorio remoto sobre *single-board computer* de bajo coste, el coste total de la implantación es muy reducido. Un análisis más profundo de la cuestión de la eficiencia de costes se explica en la Sección 3.8.

3.7.2 **Requisitos**

Los requisitos establecidos en la Sección 3.3 eran los siguientes:

- Experimentación de sistemas embebidos
- Universalidad
- Modularidad
- Fiabilidad

La arquitectura WebLabPRO es capaz de proporcionar una experiencia de experimentación de sistemas embebidos a distancia en tiempo real para

múltiples usuarios simultáneos. Los usuarios finales son capaces de programar, controlar e interactuar con el dispositivo embebido y también son capaces de monitorizar los resultados de sus programas, todo en tiempo real, gracias a los periféricos de salida y al sistema de transmisión interactiva en vivo. Esta novedosa arquitectura proporciona capacidades de interacción mejoradas con respecto a las diferentes arquitecturas de laboratorio remoto del estado de la técnica, que se explican a continuación.

La arquitectura WebLabPRO, desde el punto de vista del cliente, es universal. Está totalmente basada en la web, lo que significa que es accesible a través de puertos estándar de Internet, y desde cualquier dispositivo o plataforma combinada con cualquier sistema operativo. Los únicos requisitos de acceso son la conexión a Internet y el acceso a un navegador web. Desde la perspectiva del propietario del laboratorio, la arquitectura WebLabPRO también es universal, porque el sistema integrado de gestión remota de laboratorios WebLab-Deusto o RLMS, es totalmente compatible con los sistemas de gestión de aprendizaje de primer nivel, o LMS, como Moodle, y también es integrable en cualquier otra herramienta educativa basada en la web que la mayoría de las instituciones utilizan (Al-Busaidi and Al-Shihi, 2010).

El requisito de modularidad de la arquitectura WebLabPRO se ha logrado gracias a una combinación de diseño de interconexión basado en protocolos y en medios de comunicación estandarizados, y al empleo de entidades de hardware y software desacopladas. Las diferentes entidades de software mostradas en la Figura 3.2 pueden desplegarse sobre servidores físicos separados o pueden unirse en la misma máquina si la instalación física del laboratorio remoto lo requiere, facilitando el proceso de despliegue y potenciando la modularidad.

Además, desde el punto de vista del hardware, la arquitectura utiliza diferentes protocolos de comunicación, como Ethernet o SPI, que actúan como bordes de capa, asegurando que diferentes dispositivos, como el Servidor de Interfaz, puedan interconectarse con los diferentes experimentos remotos de diversas formas para conseguir al mismo tiempo una alta escalabilidad y una adecuada adaptación a las características físicas del lugar de despliegue.

También se ha satisfecho el requisito de fiabilidad. La arquitectura está diseñada para gestionar un gran número de instancias de experimentación. Como se ve en la Figura 3.2, está formada por tres capas diferentes, que pueden escalar verticalmente para dar soporte a un número arbitrario de usuarios, manteniendo los costes bajo control.

Un laboratorio remoto con un número incluso modesto de instancias de experimentación activas puede dar servicio a un elevado número de usuarios, como se explica con más detalle en la Sección 3.9. En caso de que una instancia de experimentación esté fuera de línea debido a un fallo de software o hardware, el resto de las instancias de experimentación seguirían estando disponibles para los usuarios de forma automática y transparente. El Sistema de Gestión de Laboratorios Remotos actúa como una capa de abstracción para ellos, y equilibra automáticamente la carga y ajusta el tiempo de espera en función de la carga de usuarios y de las instancias de laboratorio activas disponibles, evitando así la pérdida de servicio. No obstante, cabe destacar que si los componentes compartidos, como un servidor de interfaz, fallan, varias instancias de laboratorio podrían romperse al mismo tiempo. Por ejemplo, en el caso del laboratorio remoto Arduino de LabsLand, cada servidor de interfaz es compartido por 4 instancias de laboratorio. Por lo tanto, si se rompiera, las 4 instancias también fallarían. Sin embargo, los componentes que se comparten son precisamente los que no son especialmente propensos a fallar.

3.7.3 **Comparación entre el laboratorio remoto Arduino de LabsLand y los laboratorios y arquitecturas del estado del arte**

En esta subsección el laboratorio remoto Arduino de LabsLand y su arquitectura serán comparados desde una perspectiva técnica con los laboratorios remotos del estado del arte descritos anteriormente y sus arquitecturas. La Sección 3.8 aporta un análisis de eficiencia de costes comparando estos laboratorios y sus arquitecturas.

La Tabla 3.1 indica el rendimiento de los laboratorios remotos analizados y sus arquitecturas desde el punto de vista técnico. Se dan valores positivos o negativos para cada característica analizada. Como se puede observar, la evaluación de este conjunto de características sugiere que la arquitectura propuesta en este capítulo es más avanzada.

Desde el punto de vista arquitectónico, la arquitectura propuesta es avanzada en términos de escalabilidad, adaptabilidad y modularidad en comparación con las arquitecturas de laboratorios remotos descritas anteriormente. El hecho de ser una arquitectura altamente desacoplada asegura que cada componente se encarga de gestionar tareas más sencillas y, combinado con el uso de protocolos de comunicación bien establecidos entre estos componentes, consigue

	LabsLand Arduino	**RELDES**	**RELLE RExLab**
Modularidad			
Arquitectura desacoplada	✓	✗	✗
Adaptabilidad			
Comunicación USB	✓	✓	✓
Comunicación SPI	✓	✗	✗
Comunicación Serie	✓	✗	✗
Comunicación JTAG	✓	✗	✗
Usabilidad			
Stream en tiempo real	✓	✓	✓
Periféricos de entrada	✓	✓	✓
Señales analógicas de entrada	✓	✗	✗
Señales digitales de entrada	✓	✗	✓
Periféricos de salida	✓	✓	✓
Señales analógicas de salida	✓	✗	✗
Señales digitales de salida	✓	✗	✗
Consola de comunicación Serie	✓	✓	✓
Universalidad			
Entorno de desarrollo integrado	✓	✗	✓
Laboratorio basado en Web	✓	✓	✓
Integrable en LMS	✓	✗	✓
Fiabilidad			
Soporte múltiples instancias	✓	✗	✓
Resultado	**17/17**	**6/17**	**10/17**

Tabla 3.1: Comparación de las arquitecturas de los laboratorios remotos analizados y sus características.

un alto nivel de modularidad. Ambas características permiten también una alta escalabilidad, ya que los procesos de control y las interconexiones son más sen-

cillos, y facilita la replicación de componentes y ramas. Una vista detallada de la arquitectura del laboratorio remoto RELDES puede verse en la Figura 3.8.

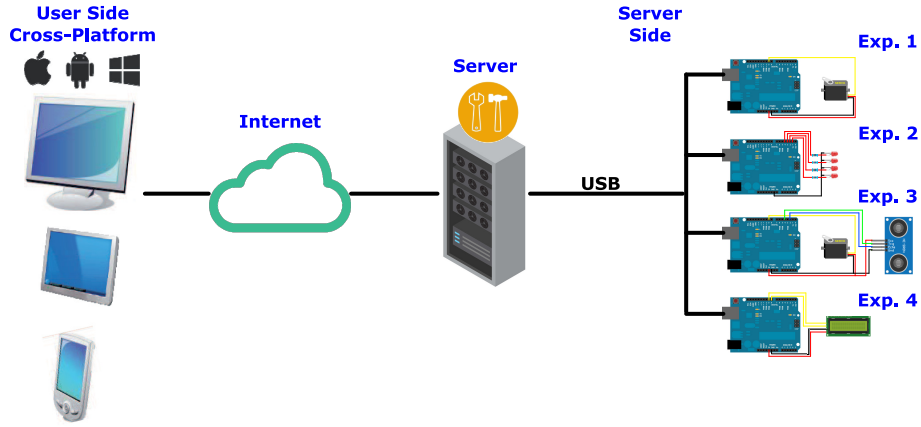


Figura 3.8: Arquitectura del laboratorio remoto RELDES. Adaptado de (Anzhelika et al., 2015)

Tanto la modularidad como la escalabilidad se ven obstaculizadas en los ejemplos descritos anteriormente, porque en ambos laboratorios las interfaces de hardware son gestionadas directamente por el servidor del laboratorio. Esto significa que, en caso de que se necesite dar soporte a un gran número de usuarios, se necesita o bien una réplica completa tanto del servidor del laboratorio como de los *setups* de experimentación, lo que aumenta los costes proporcionalmente, o bien una reconversión completa de la arquitectura del laboratorio. En el caso del laboratorio RExLab Arduino de RELLE, el laboratorio podría escalarse replicando el servidor completo del laboratorio, que se basa en un *single-board computer* de bajo coste, pero incluso en este caso, se necesita un *single-board computer* para cada instancia, lo que aumenta los costes de implementación.

Ambos laboratorios agrupan su capa de hardware y servidor de interfaz en una única entidad. Esto significa que, en general, si el laboratorio se modificara, el cambio afectaría a toda la entidad. La arquitectura WebLabPRO propuesta, sin embargo, está dividida en varias capas, incluyendo capas desacopladas de hardware y de servidor de interfaz. Dentro de esas capas, las entidades también son independientes. Por lo tanto, si se modificara el labora-

torio, el cambio sólo implicaría a las entidades afectadas, que en la mayoría de las circunstancias será sólo un pequeño subconjunto.

En los laboratorios descritos anteriormente, la adaptabilidad también se ve obstaculizada, ya que ambos sólo admiten *setups* de experimentación basados en USB. La arquitectura propuesta se ha desarrollado de forma que soporta, tanto desde el punto de vista del software como del hardware, varios protocolos entre el *setup* de experimentación y la interfaz de hardware del trabajador. Estos incluyen USB, JTAG, interfaces basados en serie y SPI, entre otros. Esto asegura la conectividad con la mayoría de las plataformas de desarrollo de sistemas embebidos y la protección para soportar el corto ciclo de vida de dichas plataformas.

Desde el punto de vista de la usabilidad, las configuraciones de laboratorio remoto descritas también pueden mejorarse en algunos aspectos. La capacidad más destacable que les falta a estos laboratorios es el control del *setup* de experimentación después de la programación. El laboratorio remoto de RELDES sólo ofrece una consola de serie unidireccional como medio de control una vez programado el *setup* de experimentación, lo que reduce las capacidades de aprendizaje de la experiencia de experimentación y amplía la brecha entre la experimentación real y la remota. La configuración del laboratorio remoto RExLab Arduino de RELLE es más avanzada en este aspecto que su competidor ucraniano, y proporciona consola serie y señales digitales de entrada. La arquitectura WebLabPRO propuesta en este trabajo proporciona capacidades de interacción más avanzadas. En particular, ofrece señales digitales de control de entrada y salida, señales analógicas de control de entrada, periféricos visuales de salida y una interfaz de consola en serie bidireccional para lograr una interacción completa y de alta calidad entre el usuario y la configuración experimental.

El aspecto de la universalidad de las tres arquitecturas analizadas se satisface de forma similar. Los tres laboratorios están totalmente basados en la web, lo que implica que son accesibles desde cualquier dispositivo (por ejemplo, ordenador portátil, tableta, smartphone) con cualquier sistema operativo y desde cualquier navegador web. El laboratorio RExLab Arduino de RELLE ofrece, al igual que la arquitectura propuesta, tanto un IDE integrado y basado en la web, que elimina la necesidad de que el usuario desarrolle sus programas en cualquier IDE específico de un proveedor y restringido por la plataforma; y también soporte para la integración con los principales sistemas de gestión del

aprendizaje, como Moodle o Google Classroom. El laboratorio RELDES, sin embargo, no ofrece esas características.

En esta tesis, el requisito de fiabilidad está fuertemente ligado a la escalabilidad y al soporte multiusuario. Básicamente, toda arquitectura que tenga la capacidad de soportar múltiples usuarios, con múltiples instancias que puedan trabajar con independencia entre ellas en caso de que una de ellas se rompa, cumple con el requisito de fiabilidad. Tanto el laboratorio remoto Arduino de LabsLand como los laboratorios RExLab de RELLE ofrecen cierto grado de escalabilidad, por lo que ambos también cumplen, al menos parcialmente, el requisito de fiabilidad.

3.8 Evaluación de la eficiencia de costes

En la Sección 3.7 se analizaron dos laboratorios remotos de última generación y sus arquitecturas desde una perspectiva técnica y se compararon con la arquitectura propuesta en este capítulo. En esta sección, las tres arquitecturas y sus laboratorios resultantes se analizan desde el punto de vista de la eficiencia de costes.

La cuestión de la eficiencia de costes está estrechamente relacionada con el número de instancias de experimentación que soporta cada laboratorio. Esta sección pretende comprobar si una arquitectura centrada en la alta escalabilidad puede aumentar la eficiencia de costes de un laboratorio remoto compuesto por múltiples instancias de experimentación, reduciendo el coste por instancia desplegada. Los diferentes niveles de compartición de hardware entre instancias pueden tener un alto impacto en el indicador de coste por instancia.

Para realizar un análisis de la eficiencia de costes de la arquitectura, se han estimado los costes de implementación de cada laboratorio remoto para cuatro escenarios diferentes, que comprenden una, cuatro, dieciséis y noventa y seis instancias de laboratorio respectivamente. Estos números específicos se han elegido para ayudar a representar la tendencia del coste por instancia de cada arquitectura, y cómo se escalarían a un número cada vez mayor de usuarios.

El número y el tipo de componentes utilizados se han ajustado a la especificación de cada arquitectura y su coste se ha unificado a través de los tres laboratorios, para que puedan ser comparados. El precio de los componentes se ha ajustado a los precios reales de reputados distribuidores de hardware europeos. Se descartan de este análisis los sistemas de gestión remota de los

laboratorios, así como los servidores estándar que necesitan los diferentes *backends* de los sistemas de transmisión en tiempo real, excluyendo las cámaras reales.

En las siguientes subsecciones se analiza el hardware de cada laboratorio desde la conexión de los sistemas de gestión remota de laboratorios hacia abajo.

3.8.1 Laboratorio remoto RELDES Arduino

El Laboratorio remoto RELDES Arduino combina un Servidor de Laboratorio, un Servidor de Interfaz y un Servidor de Gestión, todo en un módulo de software y hardware, que se basa en un *single-board computer* Raspberry Pi 3. Debido a esto, el coste del *single-board computer* Raspberry Pi debe añadirse a los gastos totales del laboratorio. Los costes recogidos en la Tabla 3.2 representan los gastos totales de hardware para el despliegue de un laboratorio RELDES Arduino con una, cuatro, dieciséis y noventa y seis instancias de experimentación. El coste de cada instancia de experimentación incluye el coste de las cuatro instancias parciales que incluye este laboratorio. Como se ha descrito en la Sección 3.7, RELDES incluye cuatro instancias parciales, que son similares pero ofrecen diferentes periféricos de salida, y que por tanto son en realidad laboratorios diferentes desde la perspectiva del usuario. Por lo tanto, este laboratorio y su arquitectura tienen como objetivo maximizar la diversidad de objetos remotizables en lugar de la escalabilidad. Para que un estudiante pueda realizar diferentes prácticas con Arduino utilizando todo el hardware propuesto, sería necesario realizar cuatro accesos independientes. Para satisfacer el requisito de usabilidad para los usuarios finales, cada iteración del laboratorio remoto, al escalar el laboratorio, necesitaría incluir estas cuatro instancias.

3.8.2 Laboratorio remoto RExLab Arduino de RELLE

El Laboratorio remoto RExLab Arduino de RELLE es compatible con los sistemas RLMS, y combina el Servidor de Laboratorio y el Servidor de Interfaz en una sola entidad, que también se basa en un *single-board computer* Raspberry Pi 3. Los costes recogidos en la Tabla 3.3 representan los gastos totales de hardware para el despliegue de un laboratorio RExLab Arduino con una, cuatro, dieciséis y noventa y seis instancias de experimentación.

Unidades para 1 instancia	Unidades para 4 instancias	Unidades para 16 instancias	Unidades para 96 instancias	Dispositivo hardware	Coste (€)
1	4	16	96	Raspberry Pi 3	38.82
4	16	64	384	Arduino UNO	22.01
1	4	16	96	Cámara	150.00
1	1	1	1	Estructura física del laboratorio	29.00
1	1	1	1	Fuente de alimentación	25.24
1	4	16	96	Cableado	2.92
1	4	16	96	Servomotor	15.05
1	4	16	96	Pantalla HD44780	10.99
1	4	16	96	Sensor ultrasónico HC-SR04	3.83
4	16	64	384	Diodos LED	0.46
4	16	64	384	Resistencias	0.28
				Coste para 1 instancia	366.85
				Coste para 4 instancias	1,304.68
				Coste para 16 instancias	5,056.00
				Coste para 96 instancias	30,064.80

Tabla 3.2: Coste del hardware para distintos números de instancias del laboratorio RELDES Arduino.

Unidades para 1 instancia	Unidades para 4 instancias	Unidades para 16 instancias	Unidades para 96 instancias	Dispositivo hardware	Coste (€)
1	4	16	96	Raspberry Pi 3	38.82
4	4	16	96	Arduino UNO	22.01
1	4	16	96	Cámara	150.00
1	1	1	1	Estructura física del laboratorio	29.00
1	1	1	1	Fuente de alimentación	25.24
1	4	16	96	Protoboard	7.15
1	4	16	96	Cableado	2.92
1	4	16	96	Servomotor	15.05
1	4	16	96	Pantalla HD44780	10.99
1	4	16	96	Sensor de humedad	3.83
1	4	16	96	Sensor de temperatura	3.83
4	16	64	384	Diodos LED	0.46
4	16	64	384	Resistencias	0.28
				Coste para 1 instancia	311.80
				Coste para 4 instancias	1,048.48
				Coste para 16 instancias	4,175.20
				Coste para 96 instancias	24,780.00

Tabla 3.3: Coste del hardware para distintos números de instancias del laboratorio RExLab Arduino de RELLE.

3.8.3 Laboratorio remoto Arduino de LabsLand

El laboratorio remoto Arduino de LabsLand basado en la arquitectura WebLabPRO propuesta es totalmente compatible con los sistemas RLMS, pero sus capas se despliegan de forma diferente. En este caso, el Servidor de Interfaz está desacoplado del Servidor de Laboratorio, lo que significa que, para gestionar de una a noventa y seis instancias de laboratorio, sólo se necesita un *single-board computer* Raspberry Pi 3. Además, para dar soporte al Servidor de Laboratorio, que puede tener hasta noventa y seis instancias virtuales de servidor web, se necesita un servidor de propósito general dedicado. Los costes recogidos en la Tabla 3.4 representan los gastos totales de hardware para el despliegue de laboratorios remotos basados en la arquitectura propuesta con una, cuatro, dieciséis y noventa y seis instancias de experimentación, lo que significa que cada configuración es capaz de dar servicio a hasta uno, cuatro, dieciséis o noventa y seis usuarios concurrentes, respectivamente.

3.8.4 Análisis del coste de los laboratorios del estado del arte

En las subsecciones anteriores se ha analizado el coste del despliegue de hardware para los tres laboratorios diferentes. En cada caso, se han seguido las pautas marcadas por cada arquitectura. En esta subsección, se analizan las tres arquitecturas desde la perspectiva de la eficiencia de costes.

Como se ha mostrado en las subsecciones anteriores y sus tablas asociadas, el coste por instancia cuando se despliega una sola instancia de cada laboratorio remoto está bastante igualado, y los gastos totales de hardware oscilan entre 300 y 400 euros. Esto no es una coincidencia, ya que para soportar una sola instancia, todas las arquitecturas analizadas requieren casi los mismos componentes básicos de hardware.

Las diferencias aparecen cuando el número de instancias requeridas aumenta, ya que no todas las arquitecturas pueden escalar de la misma manera. Como se ve en las Tablas 3.2, 3.3 y 3.4, los gastos totales no sólo varían, sino que divergen mucho a medida que aumenta el número de instancias. Tanto los laboratorios RELDES como RExLab terminan con un coste total de implantación entre 25.000 y 31.000, mientras que el coste total del laboratorio remoto Arduino de LabsLand termina por debajo de la barrera de los 15.000, para un número de noventa y seis instancias soportadas. Esta diferencia se debe a la forma en que cada arquitectura gestiona la escalabilidad. Dado que las arqui-

Unidades para 1 instancia	Unidades para 4 instancias	Unidades para 16 instancias	Unidades para 96 instancias	Dispositivo hardware	Coste (€)
0	1	1	1	Servidor de laboratorio	251.82
1	1	1	6	Raspberry Pi 3	38.82
1	4	16	96	Arduino UNO	22.01
1	2	8	48	Cámara	150.00
1	1	1	1	Estructura física del laboratorio	29.00
1	1	1	1	Fuente de alimentación	25.24
1	1	1	1	PCB de interconexión	45.00
0	0	1	6	Hub USB	9.89
1	1	4	24	Expansor de puertos SPI	2.92
2	8	32	192	Conversor D/A SPI	2.92
1	4	16	96	Servomotor	15.05
1	4	16	96	Pantalla I2C 1.3”	10.99
5	20	80	480	Diodos LED	0.46
7	28	112	672	Resistencias	0.28
				Coste para 1 instancia	347.59
				Coste para 4 instancias	919.24
				Coste para 16 instancias	2,517.21
				Coste para 96 instancias	13,347.96

Tabla 3.4: Coste del hardware para distintos números de instancias del laboratorio Arduino de LabsLand.

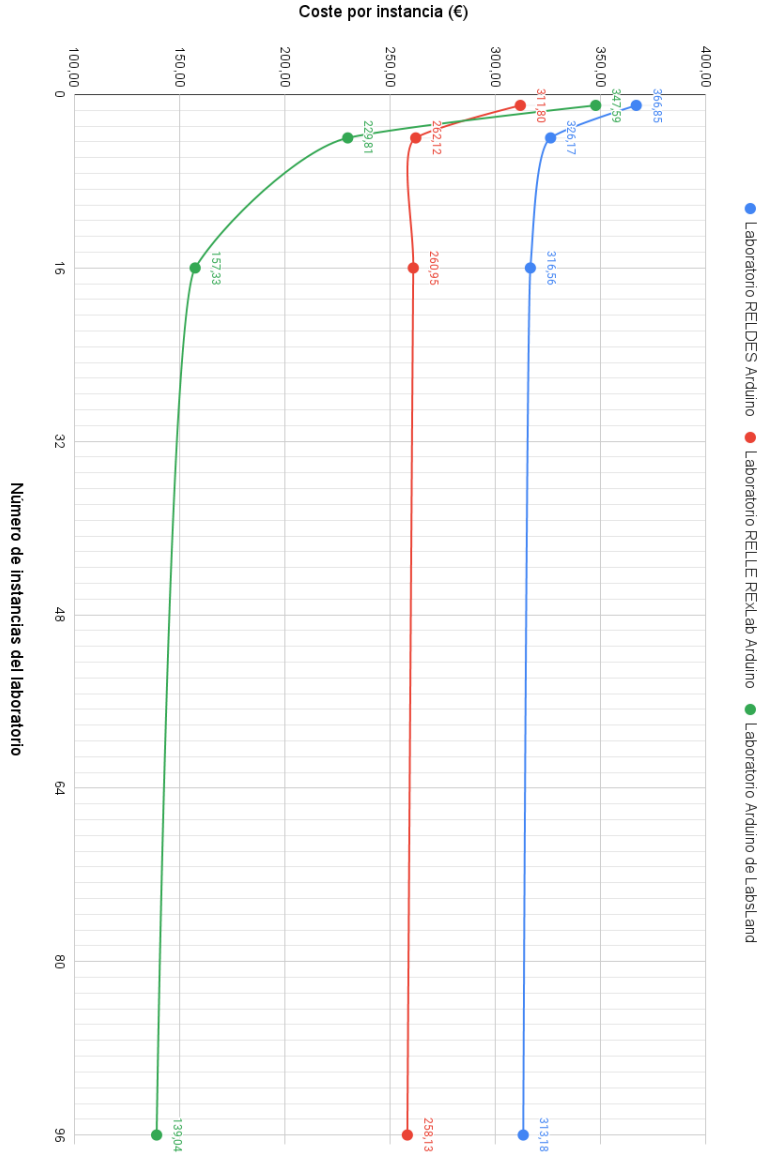


Figura 3.9: Coste por instancia para despliegues de 1, 4, 16 y 96 instancias de experimentación.

tecturas de laboratorio RELDES y RExLab no están diseñadas para compartir recursos de hardware entre cada instancia (con la excepción de la fuente de alimentación y la estructura física), se necesita una réplica completa del hardware de cada instancia para cada nueva instancia, lo que escala los costes proporcionalmente. La arquitectura está diseñada de forma que se maximiza la compartición de hardware entre instancias, lo que significa que prácticamente el único hardware que se replica al crear nuevas instancias es el objeto remotizable. En este caso, el servidor de interfaz no requiere replicación, y el hardware de control, que incluye expansores de puertos y convertidores digital-analógicos, es barato de replicar, y tampoco requiere reconfiguración del software.

Esta cuestión reduce en gran medida el coste por instancia de cada laboratorio remoto. Como se observa en la Figura 3.9, el coste por instancia para los laboratorios remotos RELDES y RExLab disminuye mínimamente para 1 y 4 instancias, y se estabiliza en valores en torno a 315 y 260 euros respectivamente, mientras que el coste por instancia del laboratorio remoto Arduino de LabsLand disminuye mucho y de forma continua a medida que se soportan más instancias, y se estabiliza en un valor inferior de 140 euros, que se estima que es la mitad del coste por instancia que las otras arquitecturas. Como se observa en la Tabla 3.5, al desplegar 96 instancias, el coste por instancia para el laboratorio remoto Arduino de LabsLand es de aproximadamente 140. De estos 140, aproximadamente, 129 se destinan a la compra del hardware que conforma el objeto remotizable, y sólo unos 10 se destinan a la compra del hardware de interconexión. En este caso, el coste del hardware de interconexión, que está muy ligado a la arquitectura elegida, sólo representa un 7,3 % del coste total de la instancia, lo que es significativamente menor que en los laboratorios RELDES Arduino y RExLab Arduino. Esto sugiere que cuando los laboratorios remotos multi-instancia se desarrollan siguiendo una arquitectura centrada en la alta escalabilidad y la eficiencia de costes, el coste del hardware directamente relacionado con las necesidades de la arquitectura puede reducirse significativamente.

Conseguir un sobrecoste mínimo sobre el coste del hardware del objeto remotizable es una acción importante, ya que esta diferencia de coste se multiplica por el número de instancias desplegadas, haciendo que el coste total aumente rápidamente cuando se despliega un elevado número de instancias de laboratorio.

El análisis de eficiencia de costes sugiere que la arquitectura WebLabPRO

	RELDES	**RELLE RExLab**	**LabsLand Arduino**
Coste total (€)	313.18	258.13	139.04
Coste experimento (€)	205.54	216.52	128.86
Coste experimento (porcentaje)	65.6 %	83.9 %	92.7 %
Coste arquitectura (€)	107.64	41.61	10.18
Coste arquitectura (porcentaje)	34.4 %	16.1 %	7.3 %

Tabla 3.5: Resumen de costes de desarrollo de los laboratorios para un total de 96 instancias de experimentación.

puede conseguir un coste mínimo de sobrecarga sobre el coste del objeto remotizable, un indicador que impacta fuertemente en los gastos totales de un despliegue de laboratorio remoto multi-instancia. También sugiere que la arquitectura propuesta es más eficiente en costes que las alternativas, especialmente cuando el número de instancias desplegadas es elevado, debido en parte al diseño que permite y fomenta el uso compartido de hardware entre grupos de instancias.

3.9 Evaluación de la escalabilidad

En esta sección se utilizará la prueba de concepto descrita en la Sección 3.6 para analizar la arquitectura WebLabPRO desde el punto de vista de la escalabilidad.

Para asegurar que la arquitectura y la implementación cumplen con todos los requisitos de los centros educativos del mundo real, el laboratorio remoto Arduino de LabsLand se ha probado en un entorno de trabajo real. Un grupo de alumnos de un instituto vasco ha utilizado el laboratorio para realizar prácticas con Arduino tanto en clase como en casa. Debido al tipo de prácticas que iban a desarrollar, cuatro instancias del laboratorio remoto Arduino de LabsLand eran suficientes para gestionar el acceso de hasta 14 usuarios, que utilizaban el laboratorio en tramos horarios. En la siguiente subsección se relata y analiza de forma más detallada esta experiencia en clase.

3.9.1 **Experiencia en clase**

Esta subseccion recoge toda la información relacionada con una prueba real de la arquitectura que se ha realizado durante el periodo de un mes en un aula y sin acceso por parte del doctorando. El laboratorio remoto desarrollado se ha utilizado en un entorno de producción real proporcionado por LabsLand con el objetivo de desarrollar una prueba completa sobre la arquitectura WebLabPRO y sobre el laboratorio remoto Arduino de LabsLand.

En el mes de marzo de 2019, durante un periodo de 30 días, el laboratorio remoto Arduino de LabsLand se ha puesto en estado de producción monitorizada para probar sus capacidades de forma extensiva y poder recoger diferentes parámetros en un entorno real. Durante esta prueba, ni el profesor ni los alumnos tuvieron conocimiento de este proceso evaluativo.

Un grupo de 14 usuarios de un instituto vasco, formado por 11 hombres y 3 mujeres, con edades comprendidas entre los 15 y los 16 años, han accedido al laboratorio remoto Arduino de LabsLand para aprender programación basada en Arduino en la clase de Tecnología. El objetivo principal de la asignatura es enseñar los principios y aplicaciones de la programación básica e introducir a los alumnos en el diseño de dispositivos basados en microcontroladores. Los alumnos aprenden sobre programación y rutinas de control en Arduino y lenguajes de programación basados en bloques, al mismo tiempo que aprenden a combinar la programación, los microcontroladores y los dispositivos embebidos con otros aparatos eléctricos y electrónicos para crear dispositivos mejorados.

Los alumnos acceden al laboratorio a través del Sistema de Gestión de Aprendizaje propio del centro, o LMS, que en este caso es Google Classroom. Gracias al soporte de la arquitectura WebLabPRO no es necesario descargar ninguna utilidad ni realizar ninguna reconfiguración. Tras acceder al IDE integrado, los alumnos programan sus rutinas de control y crean sus propios programas de control. Después de la etapa de desarrollo, acceden al objeto remotizable real, lo programan y observan los resultados de la programación a través de una captura de transmisión en vivo. Además, en el periodo de tiempo asignado para realizar estas pruebas los alumnos también pueden interactuar con la placa Arduino programada cambiando diferentes señales de entrada (tanto analógicas como digitales) e intercomunicándose con el dispositivo a través de la consola serie.

Después de que los estudiantes hayan probado su diseño de software, la

sesión de objetos remotizables termina, y los estudiantes son redirigidos al IDE integrado para realizar nuevos cambios en el código, hasta que sea correcto. Estos pasos de programación y prueba se repiten de forma cíclica tantas veces como sea necesario, con el fin de obtener el programa Arduino que funcione según lo previsto.

No es habitual que los catorce estudiantes intenten acceder a las instancias de Arduino al mismo tiempo, pero tampoco es raro que muchos de ellos accedan al mismo tiempo. Normalmente, entre un cuarto y la mitad de los estudiantes intentan acceder a las instancias del laboratorio remoto al mismo tiempo para probar su código mientras están en clase. La proporción tiende a ser baja porque el tiempo que se tarda en escribir el código suele ser mucho mayor que el que se tarda en probarlo. Si más de cuatro estudiantes intentan acceder a las instancias de Arduino, y éstas están desocupadas, los cuatro primeros estudiantes tendrán acceso inmediato, mientras que los sucesivos se pondrán en cola. A medida que se liberan las instancias, los estudiantes de la cola comienzan a acceder a las instancias de Arduino. El tiempo que los estudiantes esperan para acceder a una instancia desocupada es el tiempo de acceso.

Durante el periodo de pruebas en tiempo real de 30 días, los estudiantes han accedido a la configuración del laboratorio 493 veces. Los estudiantes han realizado 176 accesos al IDE integrado para codificar diferentes scripts y han realizado 317 accesos a los objetos remotizables reales, para probar sus scripts. La mayoría de estos usos se han concentrado en franjas temporales de corta duración (clases de una hora, principalmente). Alrededor del 20 % de los accesos corresponden a otras franjas horarias, ya que el laboratorio también se ha utilizado para realizar prácticas en casa fuera del horario lectivo. Durante las ráfagas de alta demanda, las cuatro instancias de experimentación remota de Arduino han mantenido un tiempo de acceso relativamente bajo que no afecta negativamente a la usabilidad y que varía en todos los casos entre cero y cuatro segundos, con un tiempo medio de espera de 2,2 segundos.

Otras configuraciones de laboratorios remotos pueden tener necesidades diferentes, y en algunas ocasiones el número de instancias de experimentación disponibles y el número de estudiantes deben coincidir. Durante las pruebas realizadas, los alumnos disponen de un tiempo de prueba de 1 minuto y 30 segundos, en el que prueban el código previamente escrito. La mayoría de los usuarios emplearon menos tiempo que el máximo en probar sus programas,

con un tiempo medio de prueba de 57 segundos. Este tiempo de prueba puede ser ajustado por los profesores y los propietarios de laboratorios remotos para adaptarse mejor a las diferentes necesidades de los estudiantes.

3.9.2 **Resultados**

El desarrollo del laboratorio remoto Arduino de LabsLand, basado en la arquitectura WebLabPRO, se ha llevado a cabo sin especiales contratiempos, y la escalabilidad se ha conseguido con éxito gracias a sus capas desacopladas de hardware y software.

Desde el punto de vista multiusuario, el laboratorio ha llegado a soportar hasta 14 usuarios en una franja horaria con más de 4 accesos concurrentes. Los usuarios que solicitaron acceso cuando las cuatro instancias estaban ocupadas, se pusieron en una cola de acceso. El tiempo medio de acceso, de 2,2 segundos, es manejable en un entorno de aula, lo que refuerza la idea de que para la experimentación mixta en la que los usuarios dedican su tiempo a programar y a probar, pero no a realizar ambas tareas al mismo tiempo, un número bajo de instancias puede manejar con éxito un número mucho mayor de usuarios.

El sencillo desarrollo del laboratorio remoto, el éxito en la gestión de la asignación de instancias y el bajo tiempo de espera alcanzado sugieren que la arquitectura ha cumplido efectivamente sus requisitos de escalabilidad y manejo de múltiples usuarios concurrentes.

3.10 **Conclusiones en torno a la escalabilidad en experimentación remota**

La arquitectura WebLabPRO presentada supone una mejora del estado del arte en cuanto a escalabilidad se refiere, sustanciada en la consecución de los siguientes objetivos.

En primer lugar se ha diseñado para proporcionar escalabilidad a los laboratorios remotos con gastos moderados, soportando con éxito múltiples instancias de objetos remotizables con el fin de llegar a un mayor conjunto de usuarios de forma concurrente.

En segundo lugar la arquitectura ha sido diseñada para proporcionar un fácil acceso, tanto desde el punto de vista de la interconexión del hardware como desde el punto de vista del acceso de los usuarios. La adaptabilidad y la

interconexión a diferentes dispositivos remotizables se ve reforzada por el elevado número de protocolos de interconexión soportados. El acceso del usuario es universal, debido a la capacidad de la arquitectura de ser accedida a través del protocolo HTTP, utilizando puertos estándar para evitar problemas de cortafuegos. Al estar totalmente basado en la web, los únicos requisitos necesarios para acceder al laboratorio son un navegador web y una conexión a Internet.

En tercer lugar desde el punto de vista de la usabilidad, la arquitectura proporciona diferentes señales de entrada y salida, periféricos y técnicas de conexión, que combinadas con la plataforma de transmisión en directo, consiguen sesiones de experimentación de alta calidad. Los usuarios pueden experimentar con el dispositivo embebido casi como lo harían en un laboratorio clásico.

Por último, desde un punto de vista puramente arquitectónico, la arquitectura está totalmente desacoplada, lo que aumenta la modularidad y proporciona al propietario del laboratorio una gran flexibilidad. Esto permite una fácil adaptación de los diferentes módulos de hardware y software al entorno físico en el que se despliega el laboratorio, como se mostrará en el Capítulo 5.

El laboratorio remoto Arduino de LabsLand, que ofrece 4 instancias de Arduino UNO, al ser evaluado demuestra que la arquitectura WebLabPRO en la que se basa, cumple los requisitos clave de escalabilidad, adaptabilidad, rentabilidad, usabilidad, universalidad, modularidad y fiabilidad.

Los resultados recogidos sugieren que los objetivos y requisitos descritos anteriormente se han cumplido efectivamente, y que la arquitectura WebLabPRO debería ser útil para el nuevo desarrollo y despliegue de laboratorios remotos escalables. En este sentido, la conclusión más positiva es que LabsLand basa el desarrollo de sus laboratorios remotos para experimentación con sistemas electrónicos programables en la arquitectura WebLabPRO.

3.11 **Líneas futuras de escalabilidad en experimentación remota**

En el futuro sería interesante realizar nuevos despliegues de laboratorios remotos basados en la arquitectura propuesta con otros objetos remotizables, como FPGAs u otros dispositivos embebidos, que podrían ayudar a evaluar mejor la escalabilidad y adaptabilidad de la arquitectura a diferentes dispositivos hardware.

Asimismo, otra línea de trabajo futura interesante sería explorar la conte-

nedorización de la capa de software de la arquitectura, utilizando tecnologías como Docker, lo que podría mejorar la desplegabilidad (Corbi and Burgos, 2016) por parte de terceros y trabajar en la estandarización de la arquitectura como herramienta para el despliegue de laboratorios remotos.

CAPÍTULO

4

Diseño, implementación y validación de una arquitectura tolerante a fallos para experimentación remota

Así como la escalabilidad permitía el acceso simultáneo de un gran número de estudiantes a un experimento remoto, la fiabilidad se centra en que ese acceso dé lugar a una sesión de experimentación útil. Para ello es necesario que nada falle en el experimento remoto, o que si lo hace, esta situación sea detectada, notificada y solucionada de forma transparente para el usuario. La falta de fiabilidad es el origen de la baja adopción de los laboratorios remotos en los centros educativos. Este capítulo aborda la fiabilidad y propone una solución basada en las premisas de QoS.

4.1 Introducción

Existen muchos experimentos remotos, especialmente en el campo de la tecnología y las ciencias experimentales. Sin embargo, a pesar de su utilidad, su uso real en las aulas no está actualmente muy extendido. A menudo los experimentos sólo son utilizados por los profesores que trabajaron en su desarrollo como investigadores, a pesar de que se ofrecen a otros profesores e instituciones. Una de las principales ventajas de un laboratorio remoto es que, desde el punto de vista técnico, puede ser compartido por diferentes profesores, estudiantes y centros educativos, y la mayoría de las universidades e investigadores están dispuestos a hacerlo. Sin embargo, en la práctica rara vez son utilizados por terceros, aunque en principio son de libre acceso. Tanto (Lahoud and Krichen, 2010) como (Orduña et al., 2016) analizaron la baja tasa de utilización de los laboratorios remotos y encontraron dos aspectos significativos. Por un lado, (Lahoud and Krichen, 2010) remarcó en su estudio con profesores que la fiabilidad era la característica fundamental que se esperaba en un laboratorio remoto, y por otro lado (Orduña et al., 2016) afirmó ”Sin embargo, aunque el número de iniciativas de laboratorios remotos es elevado, el impacto global de estos laboratorios es bastante limitado más allá del ámbito de la institución anfitriona o del alcance (y duración) de los proyectos en los que ésta participa. Hay casos en los que los laboratorios son utilizados regularmente por otras instituciones, pero siguen siendo excepciones y los laboratorios remotos aún no se utilizan de forma generalizada. No es el caso de los laboratorios virtuales (simulaciones), en los que los costes de mantenimiento y el trabajo necesario una vez desarrollados suelen ser bajos”.

En otras palabras, los profesores no están dispuestos a planificar e integrar en su clase un recurso que falla o puede fallar, especialmente cuando no pueden estar seguros de saberlo de antemano y no pueden planificar fácilmente una alternativa en caso de que el laboratorio falle. La fiabilidad de los laboratorios remotos ofrecidos gratuitamente sigue planteando demasiadas incertidumbres para que se utilicen como una herramienta fiable en las aulas.

Estas incertidumbres son más aceptables y menos preocupantes para los diseñadores originales del experimento remoto, ya que éstos tendrán más información sobre el estado del laboratorio, probablemente podrán identificar y comprender los problemas, e incluso podrán tener el control físico del equipo para arreglarlo ellos mismos en caso de que falle. Por ejemplo, un problema como que un servidor se apague accidentalmente, o que se desconecte un simple

cable, a menudo puede ser resuelto en tiempo real por el diseñador original. Un profesor ajeno, sin embargo, no tendría ningún control, y tendría que cancelar la actividad prevista, con las correspondientes consecuencias.

Teniendo en cuenta la utilidad didáctica y las ventajas de los laboratorios remotos, hay motivos para pensar que la experimentación remota tendría mayor presencia en las aulas si su fiabilidad inspirara más confianza a los profesores.

En esta tesis se analizan en detalle los problemas más comunes que se producen al acceder a los experimentos remotos, y se proponen estrategias para mitigar estos problemas verificando en tiempo real la disponibilidad y el funcionamiento de los laboratorios. También se describe un modelo de replicación y federación de esos experimentos remotos para mejorar la disponibilidad del servicio una vez detectados los fallos mediante las estrategias mencionadas. Al disponer de varias copias de un mismo laboratorio y saber si un laboratorio está en condiciones de funcionar, es posible aumentar la fiabilidad del servicio desde la perspectiva de los estudiantes, redirigiéndolos automáticamente a instancias que funcionan.

El resto del capítulo se organiza como sigue: La Sección 4.2 agrupa múltiples laboratorios remotos relevantes y analiza su fiabilidad. La Sección 4.3 presenta un conjunto de dos soluciones propuestas interconectadas, incluyendo un ejemplo de integración para cada una de ellas. En la Sección 4.4 se evalúan en términos de fiabilidad dos experimentos remotos basados en la arquitectura propuesta. El capítulo finaliza con las Secciones 4.5 y 4.6 que abordan las conclusiones y las futuras líneas de trabajo, respectivamente.

4.2 **Estado del arte de la fiabilidad en experimentación remota**

El estudio llevado a cabo para este estado del arte indica que ninguno de los laboratorios remotos que serán descritos a continuación utiliza ninguna técnica de detección y corrección de fallos. Así pues, este estado del arte se centra en estudiar el nivel de fiabilidad de un conjunto de laboratorios remotos.

Metodológicamente, se ha seleccionado un conjunto de laboratorios remotos y se ha accedido a ellos durante una serie de días para evaluar su fiabilidad y funcionamiento, siendo importante remarcar que los responsables de dichos laboratorios remotos no fueron informados previamente de este análisis, para

evitar que tomaran medidas preventivas. Los laboratorios seleccionados fueron cinco, y el análisis realizado se detalla a en esta sección.

Los laboratorios se eligieron porque son bien conocidos por los investigadores y los usuarios, se comparten abiertamente y son ejemplos representativos de laboratorios remotos. Es importante señalar que el objetivo del análisis no es determinar la fiabilidad de determinados laboratorios ni compararlos entre sí. La fiabilidad no es necesariamente un objetivo de los laboratorios, que no tienen por qué ofrecer una alta disponibilidad durante el periodo de pruebas. Los objetivos de este estudio preliminar son, en cambio, clasificar la amplísima diversidad de fallos observables en una cantidad manejable de categorías, proporcionar una visión general del ecosistema de los laboratorios remotos en lo que respecta a la fiabilidad, y justificar la importancia de la fiabilidad para los laboratorios destinados a una amplia aplicación por parte de terceros y, por tanto, la relevancia del esfuerzo posterior para detectar y mitigar los problemas de los experimentos remotos.

El usuario realiza varios pasos cuando accede y controla un experimento remoto. La Figura 4.1 muestra algunos de ellos. Desde el punto de vista del usuario, la experiencia consiste en los siguientes pasos:

- **Acceso al laboratorio:** En primer lugar, los usuarios acceden a la URL del experimento remoto. Normalmente se muestra un catálogo de los experimentos remotos disponibles.
- **Acceso al experimento remoto:** A continuación, los usuarios acceden al experimento remoto propiamente dicho haciendo clic en el icono o enlace correspondiente. A veces, si la plataforma del laboratorio remoto proporciona acceso a un solo experimento, este paso y el anterior se fusionan en uno solo.
- **Configuración del experimento remoto:** Una vez dentro del experimento remoto, los usuarios normalmente tendrán la oportunidad de establecer los parámetros del experimento o de programar el dispositivo subyacente. Por lo tanto, los usuarios normalmente tendrán el control sobre la configuración del laboratorio.
- **Retroalimentación del experimento remoto:** Ya sea durante el experimento o al finalizarlo, la interfaz del laboratorio muestra los resultados a través de un *stream* en tiempo real, tablas, archivos o medios similares.

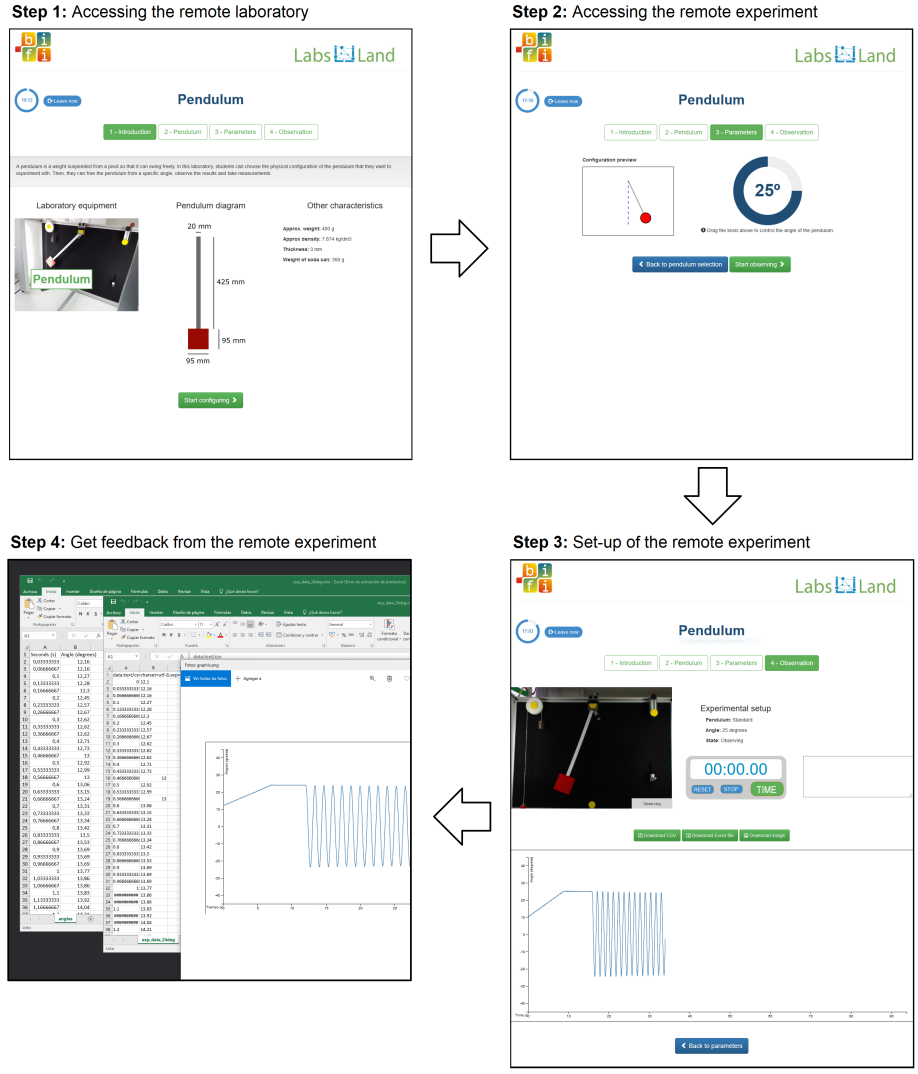


Figura 4.1: Pasos seguidos por los usuarios durante la interacción con un laboratorio o experimento remoto.

El diseño de un laboratorio remoto es muy complejo, ya que integra software, hardware, telecomunicaciones, diseño web y otros campos. Incluso si el experimento remoto se diseñó y funcionó correctamente y cumplió todos los

requisitos cuando se puso en marcha por primera vez, con el tiempo y con cierta frecuencia pueden producirse diferentes errores en cualquiera de las cuatro etapas descritas anteriormente. Estos errores serán difíciles de predecir: cortes de energía, servidores no disponibles, problemas de hardware, cámaras desenfocadas o mal colocadas y otros problemas. El objetivo de esta tesis no es detectar los errores derivados de las fases de diseño del experimento remoto. Se trata, en cambio, de detectar los fallos que surgen durante su uso convencional. El objetivo final es poder saber cuándo un laboratorio no está en condiciones de funcionar antes de que la experiencia del usuario se vea afectada negativamente, y utilizar ese conocimiento para solucionar y mitigar estos fallos, aumentando la fiabilidad del laboratorio y la confianza del usuario.

4.2.1 **Análisis de experimentos remotos**

Ahora que la experiencia de los experimentos remotos se ha clasificado en cuatro fases diferentes, esta sección analizará un conjunto pequeño pero representativo de los experimentos remotos más avanzados en términos de fiabilidad, bajo los criterios específicos que se detallan a continuación.

Para seleccionar el conjunto de laboratorios remotos que se van a analizar, se han elegido aquellos laboratorios que se ajustan a 4 criterios principales:

1. El laboratorio cuenta con experiencia contrastada y reconocida en el ámbito de la educación.
2. El laboratorio permite el acceso a más de un experimento.
3. El laboratorio concede a la comunidad educativa un acceso abierto durante las fechas en las que se realizó el análisis.
4. El acceso al laboratorio no depende de tecnologías propietarias o dependientes de la plataforma (por ejemplo, National Instruments, LabView, Java, Flash).

Como resultado, se eligieron cinco laboratorios remotos que dan acceso a un total de 42 experimentos remotos diferentes.

- **Laboratorio 1** agrupa diecisiete experimentos remotos proporcionados por iSES (Brom et al., 2018) [1], Internet School Experimental System.

[1] https://www.ises.info/index.php/en/laboratory

- **Laboratorio 2** agrupa seis experimentos remotos proporcionados por RemLabNet (Ožvoldová and Schauer, 2016) [1].

- **Laboratorio 3** agrupa doce experimentos remotos proporcionados por RExLab (Da Silva et al., 2016) [2].

- **Laboratorio 4** agrupa tres ejemplos de instancias de experimentos remotos proporcionados por WebLab-Deusto (Orduña et al., 2011) e integrados en LabsLand (Orduña et al., 2016) [3].

  Estos experimentos remotos serán los que integren la solución propuesta de fiabilidad, que se detalla en la Sección 4.3. Estos experimentos serán integrados en LabsLand para la validación de la solución propuesta.

- **Laboratorio 5** agrupa cuatro experimentos remotos proporcionados por GOLDi (Henke et al., 2019) [4], Grid of Online Laboratory Devices Ilmenau. Dos de ellos se encuentran en la Universidad Tecnológica de Ilmenau y dos en Ucrania.

La mayoría de los 42 experimentos remotos se centran en experimentos científicos en el ámbito de la física y la biología. Los nueve experimentos restantes, sin embargo, se centran en dispositivos programables.

Los experimentos remotos se analizaron en cuatro fechas diferentes, repartidas en casi un mes: 2020/04/16, 2020/04/26, 2020/05/04 y 2020/05/12. Se accedió a cada experimento en cada una de las cuatro fechas mencionadas y se analizó en función de cinco criterios para caracterizar su comportamiento.

- **Disponible:** El experimento está disponible para el usuario. Existe un enlace, una URL u otra forma de acceso que permite al usuario acceder al experimento. Un experimento no disponible sería aquel que está marcado como fuera de línea o que carece de una URL o un enlace de acceso.

- **Accesible:** El experimento es accesible para el usuario. Tras utilizar el enlace, la URL u otra forma de acceso, el sistema muestra al usuario el

[1] http://www.remlabnet.eu/
[2] https://rexlab.ufsc.br/
[3] https://labsland.com/en
[4] https://www.goldi-labs.net/

panel de la interfaz del experimento remoto. Un experimento inaccesible sería aquel que no permite al usuario acceder al experimento aunque éste muestre su voluntad de acceder. Un claro ejemplo de esto es cuando la interfaz muestra el mensaje *loading* y se congela.

- **Visible:** El experimento es visible a través de la cámara web (si el experimento tiene una cámara web, que es muy común). Se puede considerar que el experimento no es visible si la cámara web no funciona correctamente (pero debería hacerlo) y el usuario no tiene retroalimentación visual del experimento. En muchos casos, esta retroalimentación es crítica, pero en otros casos simplemente proporciona información complementaria o inmersión. Por ejemplo, en un experimento típico de robótica o en un experimento con microscopio suele ser crítico, pero no tanto en un experimento de circuitos eléctricos en el que los valores de medición se proporcionan por otros medios.

- **Controlable:** El usuario puede configurar o modificar los valores de los parámetros del experimento a través de la interfaz o cargar el programa con el algoritmo de control que controla el hardware. De lo contrario, la manipulación del experimento no es efectiva, o aunque el usuario active controles como interruptores y botones, éstos no funcionan. Otro escenario para un experimento incontrolable es aquel en el que el hardware a controlar está roto.

- **Observable:** El experimento puede ser observado en su totalidad por el usuario a través de la interfaz, y la retroalimentación de datos es correcta y consistente. Se puede considerar que no es observable si los datos que se muestran no son correctos, o los datos que se publican en la interfaz no son consistentes con lo que se muestra en la cámara web.

La Figura 4.3 muestra el resumen del comportamiento de los 42 experimentos remotos analizados en los cinco laboratorios remotos. Los experimentos están agrupados dentro del laboratorio correspondiente y para cada criterio se indica el número de días que el experimento falló. A continuación se explicitan algunas de la situaciones encontradas, que ayudan a comprender los criterios utilizados.

Por ejemplo, si el Experimento 1 en el Laboratorio 1 tiene un valor de dos en la columna ”visible”, significa que en dos de los cuatro días en que se

accedió al experimento, la cámara web no funcionó.

En otros casos, no fue posible rellenar todas las casillas si determinados fallos impedían obtener la información requerida. Por ejemplo, el Experimento 9 del Laboratorio 1 no fue accesible durante los cuatro días, por lo que fue imposible determinar el resto de los criterios, por lo que varias casillas están vacías. En el caso del Experimento 10 del Laboratorio 1, fue inaccesible la mitad de los días, y en uno de los días en que fue accesible, era visible, pero incontrolable.

Laboratory	**Experiment ID**	**Experiment name/description**
Laboratory 1: ISES	Experiment 1	Diffraction of electromagnetic radiation on microobjects I
	Experiment 2	Temperature monitoring
	Experiment 3	Electromagnetic induction
	Experiment 4	Natural driven oscillations
	Experiment 5	Magnetic fields in a coil
	Experiment 6	Solar energy conversion
	Experiment 7	Diffraction of electromagnetic radiation on microobjects II
	Experiment 8	Photoelectric effect
	Experiment 9	Polarization
	Experiment 10	Protection against ionizing radiation by distance
	Experiment 11	Monitoring of the natural background radiation
	Experiment 12	Study of spectra
	Experiment 13	Faraday phenomenon
	Experiment 14	Water level control
	Experiment 15	Rectifier
	Experiment 16	Series RLC circuit
	Experiment 17	VA characteristics of a LED - Measurement of Planck constant
Laboratory 2: RemLabNet	Experiment 1	Induction
	Experiment 2	Solar energy
	Experiment 3	Rectifier
	Experiment 4	Series RLC circuit
	Experiment 5	Diffraction of electromagnetic radiation in microobjects
	Experiment 6	Heisenberg
Laboratory 3: RExLab	Experiment 1	Painel Elétrico CC (DC electric panel)
	Experiment 2	Painel Elétrico CA (AC electric panel)
	Experiment 3	Ambiente para Desenvolvimento em Arduino (Arduino development environment)
	Experiment 4	Meios de Propagação de Calor (Heat propagation media)
	Experiment 5	Microscópio Remoto (Remote microscope)
	Experiment 6	Plano Inclinado (Inclined plane)
	Experiment 7	Disco de Newton (Newton's disc)
	Experiment 8	Conversão de Energia Luminosa em Elétrica (Conversion of light to electric energy)
	Experiment 9	Condução de calor em barras metálicas (Heat conduction in metal bars)
	Experiment 10	Arduino/block.ino
	Experiment 11	VISIR
	Experiment 12	Lab Água (Water laboratory)
Laboratory 4: WebLab-Deusto (fault-tolerance schemes bypassed)	Experiment 1	Xilinx VHDL FPGA Laboratory (one specific instance)
	Experiment 2	Arduino Remote Laboratory (one specific instance)
	Experiment 3	Arduino Robot Remote Laboratory (one specific instance)
Laboratory 5: GOLDi	Experiment 1	MCU + 3-axis portal laboratory
	Experiment 2	CPLD + production cell laboratory
	Experiment 3	MCU + 3-axis portal laboratory
	Experiment 4	CPLD + production cell laboratory

Figura 4.2: Laboratorios remotos de última generación y experimentos analizados.

	Experiment	Available	Accessible	Visible	Controllable	Observable	Result
Laboratory 1	Experiment 1	0	0	2	0	0	Only camera non-critical failure
	Experiment 2	0	0	1	0	0	Only camera non-critical failure
	Experiment 3	0	0	3	0	0	Only camera non-critical failure
	Experiment 4	0	1	3	0	0	Single one-time critical error
	Experiment 5	4	-	-	-	-	Unreliable experiment
	Experiment 6	0	0	0	0	0	Fully reliable experiment
	Experiment 7	0	0	0	0	0	Fully reliable experiment
	Experiment 8	0	0	3	0	0	Only camera non-critical failure
	Experiment 9	0	4	-	-	-	Unreliable experiment
	Experiment 10	0	2	0	1	0	Unreliable experiment
	Experiment 11	0	2	0	0	0	Unreliable experiment
	Experiment 12	0	2	0	0	0	Unreliable experiment
	Experiment 13	4	-	-	-	-	Unreliable experiment
	Experiment 14	0	0	2	0	0	Unreliable experiment
	Experiment 15	0	0	1	0	0	Only camera non-critical failure
	Experiment 16	0	0	0	0	0	Fully reliable experiment
	Experiment 17	0	0	2	0	0	Only camera non-critical failure
Laboratory 2	Experiment 1	0	0	4	0	0	Only camera non-critical failure
	Experiment 2	0	3	0	0	0	Unreliable experiment
	Experiment 3	0	0	1	0	0	Only camera non-critical failure
	Experiment 4	0	0	0	0	0	Fully reliable experiment
	Experiment 5	0	4	-	-	-	Unreliable experiment
	Experiment 6	0	4	-	-	-	Unreliable experiment
Laboratory 3	Experiment 1	0	1	1	0	1	Unreliable experiment
	Experiment 2	0	1	0	0	0	Single one-time critical error
	Experiment 3	0	1	1	2	2	Unreliable experiment
	Experiment 4	0	4	0	-	-	Unreliable experiment
	Experiment 5	0	0	0	0	3	Only camera non-critical failure
	Experiment 6	0	0	0	0	0	Fully reliable experiment
	Experiment 7	0	0	0	0	0	Fully reliable experiment
	Experiment 8	0	0	0	0	0	Fully reliable experiment
	Experiment 9	0	3	0	0	0	Unreliable experiment
	Experiment 10	0	0	0	4	-	Unreliable experiment
	Experiment 11	0	0	0	0	0	Fully reliable experiment
	Experiment 12	0	0	0	4	-	Unreliable experiment
Laboratory 4	Experiment 1	0	0	0	2	-	Unreliable experiment
	Experiment 2	0	0	0	0	0	Fully reliable experiment
	Experiment 3	2	0	2	0	0	Only camera non-critical failure
Laboratory 5	Experiment 1	0	0	0	0	0	Fully reliable experiment
	Experiment 2	0	0	0	0	0	Fully reliable experiment
	Experiment 3	0	0	4	-	-	Unreliable experiment
	Experiment 4	0	0	2	0	0	Unreliable experiment

Figura 4.3: Resumen del análisis de los experimentos a distancia durante cuatro días.

Si la cámara web de un experimento no funciona (como se refleja en la columna *visible*), la situación puede tener un efecto diferente en los estudiantes y su aprendizaje. Si la celda *visible* está coloreada en verde claro, significa que el experimento está parcialmente operativo, aunque la experiencia del usuario no es completa. Por ejemplo, el Experimento 3 del Laboratorio 1 es un experimento inductivo que utiliza un motor eléctrico, como puede verse en la Figura 4.4. En este caso, el resultado del experimento se ve en el gráfico adjunto y, por tanto, la cámara web es sólo ilustrativa, aunque refuerza claramente la experiencia del usuario y su confianza en el experimento. En otros casos, la cámara es fundamental. Por ejemplo, en el Experimento 14 del Laboratorio 1, la cámara muestra el nivel de un líquido que se está controlando, y en el Experimento 3 del Laboratorio 4, la webcam muestra el movimiento del robot.

En estos dos casos, el fallo de la cámara web dificulta el desarrollo del proceso de experimentación, por lo que las casillas están marcadas en rojo.

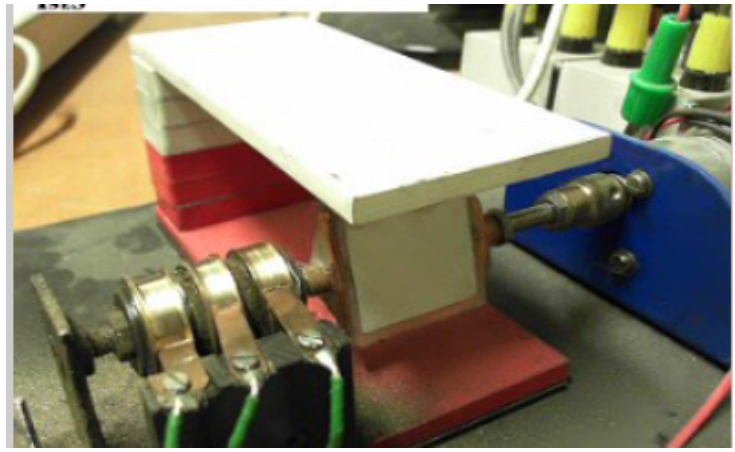


Figura 4.4: Vista de la cámara web del setup de experimentación del Laboratorio 1, Experimento 3.

Después de evaluar el comportamiento de cada uno de los experimentos con respecto a los criterios previamente establecidos, éstos se clasificaron según su fiabilidad. La Figura 4.3 enumera cuatro posibles resultados en su última columna de la derecha. Aquellos laboratorios que no experimentaron errores están marcados en color verde, y fueron categorizados como experimentos totalmente fiables. Aquellos laboratorios que sólo tuvieron fallos de visibilidad pero en los que la cámara web no era crítica se han marcado en color verde más claro, y fueron categorizados como laboratorios fiables con fallos no críticos. Esta categoría es especial, ya que incluye experimentos que, aun presentando fallos, no habrían impedido su uso en una clase, por ejemplo. Sin embargo, se han marcado en amarillo aquellos laboratorios en los que sólo se encontraron fallos durante uno de los cuatro días de pruebas. Durante el periodo de pruebas, no se demostró que fueran especialmente poco fiables, y al haber fallado sólo un día, se puede decir que podrían haberse utilizado en clase sin entorpecer demasiado la experiencia del usuario. Por último, los experimentos que presentaron errores críticos durante más de un día se clasificaron como experimentos poco fiables. En esta categoría se incluyen, entre

Tabla 4.1: Operatividad de los laboratorios remotos analizados.

Operabilidad	**Porcentaje**
Experimentos totalmente operativos	26,1 %
Semi-operativo con errores no crítico	23,8 %
No operativo durante al menos un día	50,0 %

otros, aquellos experimentos cuya cámara web es crítica y han tenido fallos de visibilidad.

4.2.2 **Análisis cuantitativo**

Se analizaron un total de 42 experimentos remotos de cinco laboratorios remotos cuyo funcionamiento se refleja en la Tabla 4.1. De ellos, 11 (26 %) fueron totalmente operativos durante los cuatro días y 21, (50 %), presentaron un fallo grave durante al menos un día. El resto de los experimentos, 10 (24 %), fueron parcialmente operativos. En estos últimos, la experiencia educativa fue correcta pero presentaron algunos problemas menores. Por ejemplo, la webcam no funcionaba pero no era crítica para el aprendizaje, ya que sólo proporcionaba información visual al usuario para aumentar la implicación en el experimento.

Los experimentos remotos evolucionaron de forma diferente durante el periodo de pruebas. La Tabla 4.2, muestra que nueve (21 %) experimentos no funcionaron en absoluto durante el análisis, número similar al de los experimentos que funcionaron en todas las ocasiones.

La Tabla 4.3 describe esos errores, teniendo en cuenta que un mismo experimento puede presentar varios errores simultáneamente. La distribución de los errores no es homogénea.

En primer lugar, hay errores de accesibilidad. En este caso, el alumno, tras acceder a la interfaz del laboratorio remoto y hacer clic en un experimento, ve que éste no aparece, no se carga o que no se puede controlar la interfaz del experimento (31 % del total). Esta situación es muy frustrante para el usuario.

Los errores en la funcionalidad de la cámara web están presentes en 15 experimentos (35 %), y producen un efecto muy negativo en los usuarios, ya que no ven lo que está sucediendo y experimentan una pérdida de control. En

Tabla 4.2: Porcentaje de laboratorios según su operatividad durante un número de 4 días.

Operabilidad por día	**Porcentaje**
Completamente operativo durante 4 días	26,2 %
Inoperativo por 1 día	19,0 %
Inoperativo por 2 día	21,4 %
Inoperativo por 3 día	7,1 %
Inoperativo por 4 día	21,4 %

cinco de esos casos (12 % del total), este fallo es crítico, ya que si el alumno no ve el hardware controlado, por ejemplo un robot, no hay experiencia ni aprendizaje.

Alrededor del 20 % de los fallos indican que, una vez que los estudiantes acceden al experimento, son incapaces de controlarlo o de extraer los datos correctos de esa sesión de experimentación. Por ejemplo, el alumno puede no ser capaz de programar un dispositivo Arduino con el código proporcionado, o una vez programado, es incapaz de interactuar con los botones o interruptores que controlan el dispositivo programado. Otros casos pueden mostrar parámetros irreales, como cero miliamperios de corriente fluyendo a través de un circuito que alimenta una bombilla, o una masa mostrada de menos 200 kilogramos al pesar un objeto particular.

Solo tres experimentos, el 7 %, están registrados como offline (no disponibles). En estos casos, los usuarios pueden saber de antemano que el experimento no estará disponible como material complementario para sus clases, y pueden tomar medidas proactivas. Esta situación es, por tanto, muy diferente del caso de un experimento inaccesible, ya que en este último caso, tanto el proveedor del experimento como el usuario pueden no ser conscientes del fallo, y pueden sentirse frustrados por este descubrimiento inesperado.

La mayoría de los experimentos que presentaron fallos más de una vez tendieron a presentar de nuevo el mismo tipo de fallo. De los 31 experimentos que presentaron algún tipo de fallo, 26 presentaron el mismo fallo durante el periodo de prueba. Sólo cinco de los experimentos presentaron múltiples tipos

Tabla 4.3: Relación de tipos de errores y cómo afectaron a los experimentos a distancia.

Tipo de error	**Porcentaje**
El experimento está fuera de línea	7,1 %
El experimento no es accesible	30,9 %
El experimento no es visible	35,7 %
El experimento no es controlable	11,9 %
El experimento no es observable	7,1 %

de fallos. El experimento que destaca en este sentido es el Experimento 3 del Laboratorio 3. Fue inaccesible el primer día, la cámara web (un componente crítico en este experimento) falló el segundo día, y presentó errores de control de la interfaz los dos últimos días. También hay que tener en cuenta que puede haber otro tipo de problemas en los laboratorios a los que no se puede acceder. Al no poder acceder a ellos, no se puede comprobar, por lo que estos casos se marcan con guiones en la Figura 4.3.

4.2.3 **Análisis cualitativo**

Del análisis anterior se deduce que la mayoría de los laboratorios remotos ofrecen experimentos erróneos o no disponibles, lo que genera frustración y falta de confianza en la experimentación remota para los usuarios potenciales, como los profesores y estudiantes que buscan incorporar estas tecnologías como herramientas de aprendizaje en el aula. En las siguientes subsecciones se analizan de forma más detallada cada uno de los tipos de error.

Experimento no disponible o fuera de línea

En este tipo de fallos, el experimento directamente no está disponible. Esta circunstancia se comunica normalmente de forma clara y directa al usuario potencial, o al menos éste se da cuenta inmediatamente del hecho al intentar acceder. Desde la perspectiva de la experiencia del usuario, este fallo es el menos negativo. En otros tipos de fallos, los usuarios invierten tiempo y esfuerzo

antes de darse cuenta de que el experimento no puede completarse, o incluso se les proporcionan resultados incorrectos. En este caso, se evitan estos problemas.

La indisponibilidad puede deberse a diferentes motivos, como tareas de mantenimiento o reparación, o a fallos más graves detectados automáticamente por el RLMS o Sistema de Gestión de Laboratorios Remotos.

Cabe destacar que, aunque es menos grave que otros errores que tienen un efecto más negativo en la experiencia del usuario, la falta de disponibilidad sigue siendo, por supuesto, negativa. Si los usuarios esperaban utilizar el recurso en clase, por ejemplo, y lo habían planeado de antemano, tendrían que suspender dicha sesión.

La Figura 4.5 muestra tres paneles de acceso para diferentes experimentos, que muestran los fallos de disponibilidad. A la izquierda, se puede ver un mensaje que aparece cuando los usuarios intentan acceder a un experimento, indicando que está fuera de línea. A la derecha, hay dos paneles de acceso a diferentes experimentos, con semáforos rojos, indicando que ninguno de los dos experimentos está disponible.

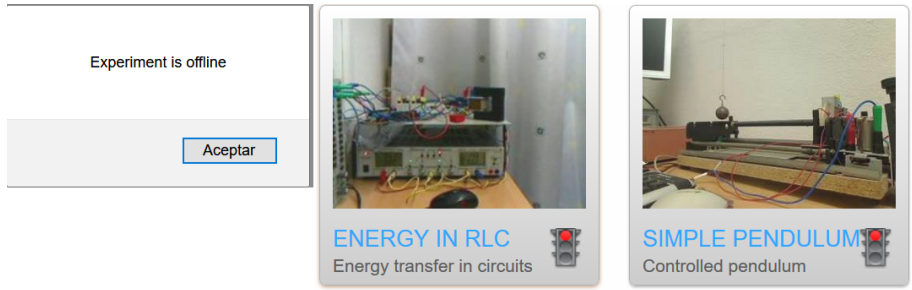


Figura 4.5: Tres paneles diferentes de acceso al experimento remoto que indican los fallos de disponibilidad.

Experimento no accesible

Según los resultados del análisis anterior, éste es el segundo error más común en los experimentos remotos, representando el 30,9 % de los casos. En estos casos, aunque el experimento parece estar disponible, no se puede acceder a la interfaz de control del experimento, o el panel de control del experimento

no se carga correctamente. Esto puede deberse a una amplia gama de causas, incluyendo problemas de red, problemas relacionados con la energía, errores de programación o errores en el sistema de gestión de usuarios/colas. La Figura 4.6 muestra un panel de control de un experimento cuyos controles no se cargan correctamente.

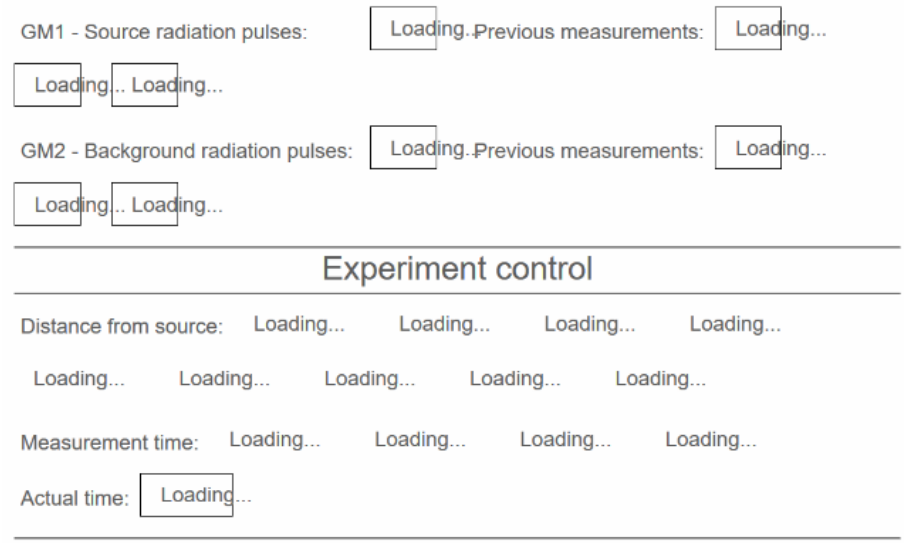


Figura 4.6: Ejemplo de errores de carga y gestión de colas en experimentos remotos.

Experimento no visible

El error más común encontrado en el análisis está relacionado con la visibilidad de la sesión de experimentación. En el 35,7 % de los casos, el experimento no se pudo ver correctamente debido a problemas con la imagen de la webcam. En algunos casos, la imagen en tiempo real no se actualiza correctamente o no se actualiza en absoluto (por lo que los resultados mostrados por los datos no coinciden con lo que se muestra en la imagen) y en otros casos la imagen no está disponible (debido a errores de red, errores de laboratorio, errores relacionados con la energía, etc.).

En algunos de los experimentos, la vista en tiempo real no es crítica, porque los datos de la experimentación también se transmiten al usuario por otros medios, pero en el resto de los experimentos, la transmisión en tiempo real del montaje de la experimentación es crítica. En estos casos, el experimento remoto queda inoperativo, como en los casos 1 y 2.

La Figura 4.7 muestra un experimento de oscilación en el que la cámara web no es un componente crítico. En este caso, el usuario no puede ver el montaje de la experimentación, pero los datos medidos se siguen transmitiendo correctamente al usuario a través de un gráfico.

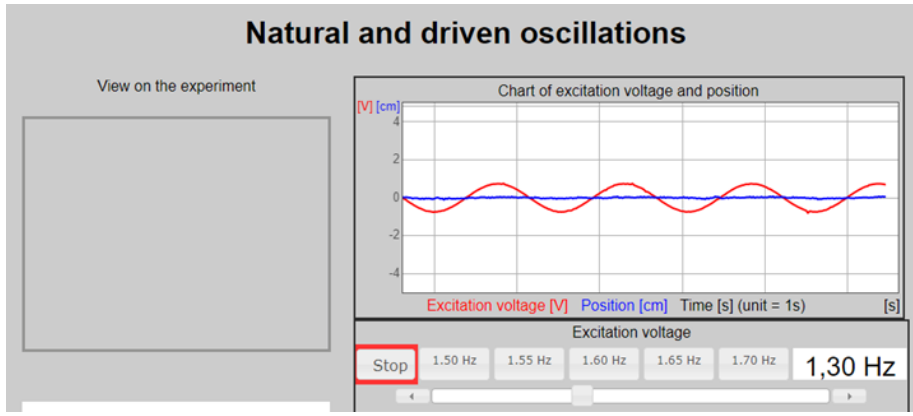


Figura 4.7: Ejemplo de un experimento remoto que no muestra la vista de la cámara web.

Cabe destacar que, en general, este tipo de error fue algo volátil. Es decir, unos días estaba presente y otros no. En concreto, de los 15 experimentos con errores de la webcam, sólo dos mostraron errores continuos durante todo el periodo de prueba. En otras palabras, la webcam aporta inestabilidad al experimento, probablemente debido a problemas de red y de configuración, y esta inestabilidad genera inseguridad en el usuario. Si los profesores no están seguros de si los estudiantes podrán ver el péndulo o no, es probable que sean muy reacios a incorporar el laboratorio en sus planes de clase.

Experimento incontrolable

El análisis ha demostrado que, aunque un experimento esté disponible y sea accesible, puede no ser controlable. En este caso, cuando se accede al experimento, no se puede controlar por una o varias razones:

- Las entradas del experimento (interruptores, botones, etc.) no funcionan.
- El hardware del experimento (motores, servomotores, sensores, etc.) no funciona.
- Los dispositivos programables del experimento (FPGA, microcontroladores, etc.) están rotos o apagados.
- Los dispositivos programables del experimento (FPGAs, microcontroladores, etc.) no pueden ser programados.

Estos errores suelen estar relacionados con el hardware o la red, y en casi todos los casos no permiten a los usuarios realizar sus sesiones de experimentación.

La Figura 4.8 muestra un experimento en el que se debe programar un dispositivo FPGA. En este caso, es imposible cargar el archivo .bit que programa el dispositivo FPGA, y por lo tanto el experimento es incontrolable. La Figura 4.9 muestra un experimento en el que se bombea agua mediante una bomba eléctrica a un depósito. En este caso, la bomba eléctrica está rota y es imposible transferir el agua al tanque. Los errores que entran en esta categoría suelen ser difíciles de detectar para el personal de mantenimiento del laboratorio, ya que puede ser necesaria una sesión completa de experimentación para probar completamente todos los dispositivos de hardware.

Experimento no observable

Este es uno de los errores menos comunes en los experimentos, pero es particularmente negativo desde la perspectiva de la experiencia del usuario. En este caso, el experimento es perfectamente controlable por el usuario, pero los datos de retroalimentación transmitidos al usuario son incorrectos o incoherentes. En muchos experimentos remotos, los datos de los sensores se capturan y manipulan antes de ser transmitidos al usuario mediante un panel de control, y pueden producirse algunos errores.

Figura 4.8: Ejemplo de experimento remoto no controlable por el usuario.

Figura 4.9: Ejemplo de experimento remoto no controlable por el usuario.

La Figura 4.10 muestra una incoherencia en los resultados de un experimento de medición eléctrica. En este caso, los datos mostrados en el *stream* de vídeo en tiempo real difieren de los datos proyectados en el panel de control. La cámara web muestra un valor de 12,23 V para el punto de medición V2, mientras que el panel de control indica una tensión de 8,74 V. Lo mismo ocurre con el punto de medición A2, donde se muestra una corriente de 0 mA en la imagen y una corriente de 16 mA en el panel de control. En este caso, el usuario puede no saber qué conjunto de valores es el correcto, lo que hace inútil la sesión de experimentación. Estos errores también son difíciles de detectar por el personal de mantenimiento del laboratorio, ya que hay que realizar una sesión de experimentación completa para comprobar todos los dispositivos de captación de señales.

En algunos casos, es posible que sólo se comunique al usuario un único conjunto de resultados. Si estos resultados son erróneos, es posible que los usuarios no se den cuenta y saquen conclusiones equivocadas, o que tarden mucho tiempo en darse cuenta de que están obteniendo resultados incorrectos.

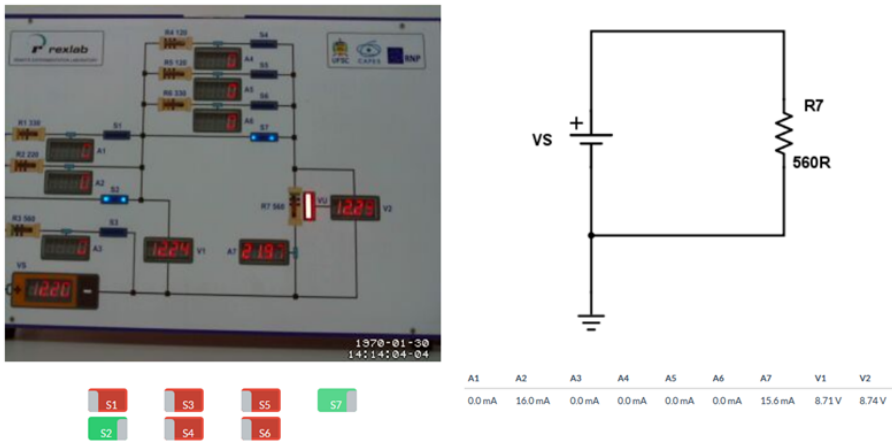


Figura 4.10: Ejemplo de un experimento remoto que no es observable por el usuario.

4.2.4 Conclusiones del análisis de los laboratorios remotos

Un laboratorio remoto con sus experimentos remotos es un diseño complejo que combina software, hardware, diseño gráfico y comunicaciones y, por tanto, puede fallar por múltiples razones, creando frustración y falta de confianza en el usuario final, ya sea estudiante o profesor. Este es un obstáculo importante que puede impedir un mayor uso de la experimentación remota en las aulas.

Las siguientes secciones describen una solución tecnológica que mitiga los problemas descritos anteriormente. Esta solución integra un conjunto de pruebas automáticas en la arquitectura del laboratorio remoto, con el propósito de detectar fallos e identificar aquellos experimentos que han fallado como no disponibles hasta que sean reparados. Así, el objetivo es que el sistema conozca el estado de un laboratorio en tiempo real.

Una vez conocido el estado de cada experimento, es posible emplear esta información de diferentes maneras. En primer lugar, el personal de mantenimiento de los experimentos puede conocer inmediatamente los posibles problemas y resolverlos, si es posible. Por ejemplo, si un pequeño fallo en un dispositivo, como una cámara web, está provocando el fallo del experimento, podría bastar con reiniciar la cámara web.

En segundo lugar, si el laboratorio lo soporta, es posible que dichos errores sean resueltos de manera automática, sin necesidad de una intervención manual.

En tercer lugar, esta información también puede servir de base para un modelo de tolerancia a fallos más completo. Si hay varias instancias del mismo experimento disponibles, y se sabe cuáles están fallando, es posible dirigir a los usuarios sólo a las que funcionan. De esta manera, los usuarios no experimentarán realmente fallos y el experimento estará disponible todo el tiempo, incluso si internamente algunas instancias del experimento están experimentando problemas. Este modelo se presenta a continuación.

4.3 Solución propuesta para fiabilidad en experimentación remota

Como muestra el análisis de las secciones anteriores, los laboratorios remotos suelen ser poco fiables, en general. Mientras que algunos podrían ser fiables simplemente teniendo especial cuidado con el software y el hardware y pro-

bando ambos a fondo, hay otros que contienen muchas piezas móviles, o piezas que se desgastan. No se puede esperar que estos laboratorios funcionen de forma fiable durante mucho tiempo si no se toman medidas específicas.

Para lograr la fiabilidad, la solución propuesta en esta tesis trabaja bajo un supuesto clave: todas las instancias de los experimentos individuales acaban rompiéndose. Por lo tanto, para proporcionar fiabilidad a los usuarios finales, será necesario, en primer lugar, detectar que una instancia concreta del experimento está fallando y, en segundo lugar, tener y poder proporcionar a ese usuario una instancia diferente del experimento que funcione, de forma transparente.

4.3.1 Capas de detección de fallos

Como se ha explicado anteriormente, la clave de la primera parte de esta arquitectura es poder detectar que un experimento está fallando. Si esto no es posible, el sistema asumirá que está funcionando, los usuarios accederán a él, y potencialmente sufrirán una muy mala experiencia de usuario. En ocasiones, simplemente no podrán trabajar en el laboratorio como se esperaba, y tendrán que reprogramar sus clases, lo que puede suponer un inconveniente importante. En otras ocasiones, la experiencia puede ser incluso peor: no se darán cuenta de que el experimento está fallando, sino que se les proporcionarán datos de experimentación incorrectos. Invertirán su tiempo en sacar conclusiones incorrectas y no volverán a confiar en el experimento.

Para detectar los fallos, la solución propuesta se basa principalmente en cuatro capas de pruebas:

- Capa 1: Captura de excepciones: *exception catching*.
- Capa 2: Controles periódicos y genéricos de estado del experimento: *generic healtcheck*.
- Capa 3: Controles periódicos y específicos del estado del experimento remoto: *specific healtcheck*.
- Capa 4: Ejecución de pruebas periódicas de alta especificidad.

La Figura 4.11 muestra estas cuatro capas. Las capas inferiores tienden a ser las menos específicas del experimento (1), mientras que las superiores tienden a ser más específicas del experimento (4). Asimismo, las capas inferiores

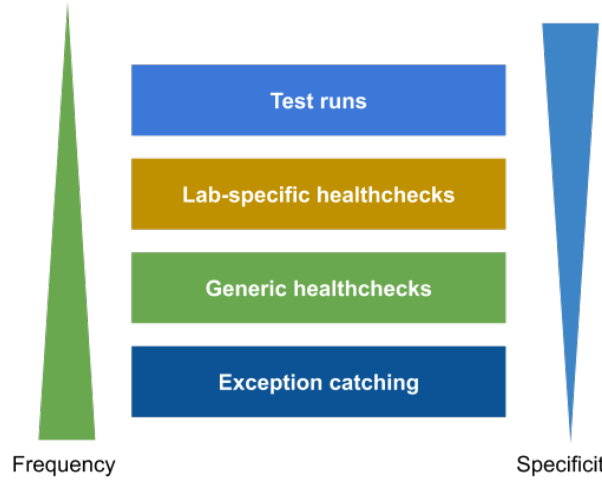


Figura 4.11: Diagrama de las cuatro capas de detección de fallos dispuestas en términos de frecuencia y especificidad.

pueden ejecutarse con más frecuencia, mientras que las superiores lo hacen con menos frecuencia (porque suelen requerir más recursos y tiempo).

La primera capa se basa en las excepciones explícitas que se capturan y registran mientras un usuario real está utilizando el experimento o cuando una prueba está en marcha (las pruebas en sí mismas se describirán como la cuarta capa, sin embargo). Por ejemplo, si un estudiante está utilizando un experimento de péndulo remoto, y el sistema de control de dicho péndulo lanza una excepción, el sistema del laboratorio puede asumir que está roto. Esto tiene al menos dos limitaciones importantes. Sólo detecta los fallos una vez que se han producido. Por lo tanto, la instancia del experimento puede apagarse automáticamente, pero normalmente sólo después de proporcionar una mala experiencia a al menos un usuario. Además, su eficacia es limitada: no todos los errores pueden detectarse fácilmente o lanzar excepciones. Por ejemplo, en muchos casos, si un servomotor se rompiera mecánicamente, el software que lo controla, podría no detectar dicha rotura. El experimento se comportaría aparentemente con normalidad desde el punto de vista del software, pero el péndulo (en este caso) se mantendría inmóvil.

La segunda capa se basa en comprobaciones periódicas y genéricas del

estado de varios componentes del experimento. Por ejemplo, un experimento podría tener componentes como un servidor de dispositivos para controlar el equipo remoto, una cámara web o un osciloscopio. La mayoría de los *health-check* genéricos implicarán acciones como el envío de solicitudes HTTP GET o POST a determinados puntos finales, o simplemente la comprobación de determinados componentes. La complejidad de estas comprobaciones puede variar. Normalmente, todas las comprobaciones del estado esperarán una respuesta satisfactoria. Por ejemplo, en el caso de las solicitudes HTTP GET, lo más frecuente es que esperen una respuesta con un código de estado 200. En otros casos, pueden esperar datos más específicos. Por ejemplo, las comprobaciones de estado de la cámara web esperan recibir no sólo un código de estado 200, sino una imagen aparentemente válida. Es decir, datos binarios de un tamaño determinado que puedan cargarse como, por ejemplo, un JPEG. Las comprobaciones de esta capa son genéricas, en el sentido de que no son específicas del experimento. Normalmente no comprobarán en detalle que la instancia específica del experimento funciona como se espera, sino que se asegurarán de que ciertos componentes clave no están fuera de línea o fallan de forma obvia.

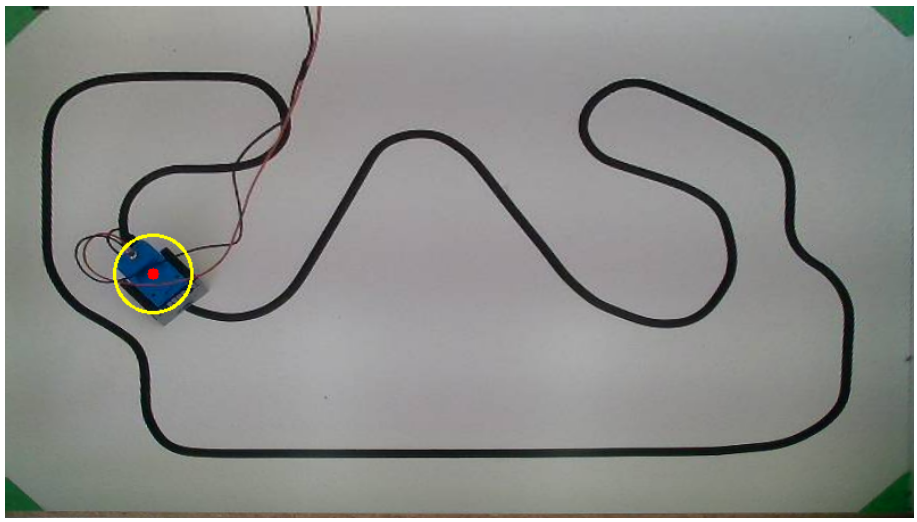


Figura 4.12: Fotograma del robot basado en Arduino del laboratorio de robótica siendo rastreado mediante técnicas de visión por ordenador.

La tercera capa es similar a la anterior, pero incluye comprobaciones específicas del experimento. Este tipo de esquema se asemeja a los utilizados en otras áreas para ayudar a garantizar la fiabilidad y la detección de fallos.

Las comprobaciones de estado de esta capa contendrán código de comprobación automático específico para el experimento. Esto detectará ciertos fallos que no se detectarían mediante comprobaciones genéricas de alto nivel. El chequeo en sí mismo puede ser casi arbitrariamente complejo. Por ejemplo, un experimento que incluya sensores podría intentar leer los datos recibidos por éstos y evaluar su rango, e informar de un error si no consigue leer los datos o si los valores no están dentro de un rango válido. Sin embargo, aunque potentes, estas comprobaciones de estado también tienen ciertas limitaciones prácticas. Una de ellas es que normalmente se ejecutan periódicamente, mientras un usuario puede estar utilizando activamente el experimento. Por tanto, deben ejecutarse con relativa rapidez y, en general, deben evitar interferir con la experiencia del usuario. Además, algunos fallos de hardware siguen siendo difíciles de detectar.

La cuarta capa se basa en ejecuciones de prueba periódicas de alta especificidad. Estas ejecuciones de prueba simulan el uso real del experimento, el cual es reservado durante un tiempo, tal como lo haría un usuario real. La ventaja de este método es que es posible realizar pruebas más exhaustivas y orientadas a los resultados. En lugar de comprobar componentes aislados, pueden centrarse en la experiencia de uso.

En el caso de un experimento de robótica, estas pruebas periódicas se ejecutan cada pocas horas. El software de control reserva una sesión de experimentación y carga en el robot un programa ejemplo, que hace que dicho robot siga una línea. En este momento, el robot comienza a moverse. Simultáneamente, un sistema basado en visión por ordenador determina las coordenadas del robot dentro de la pista, tal y como muestra la Figura 4.12.

Después de un cierto tiempo, el sistema evalúa si el robot ha seguido correctamente la línea y a qué velocidad. Si el robot no se ha movido, no ha seguido la línea con suficiente precisión o se ha movido demasiado despacio, el sistema asume que la instancia está rota. Esto puede deberse a varias causas: un motor que falla, sensores infrarrojos cubiertos de polvo, etc. Sin embargo, sea cual sea la causa, puede utilizarse para determinar que el robot no funciona como se esperaba y, por tanto, marcar la instancia como fallida, de modo que los usuarios puedan ser redirigidos a otras instancias que funcionen.

4.3.2 **Despliegue de la solución propuesta para detección de fallos en el laboratorio remoto de robótica de LabsLand**

El laboratorio remoto de robótica de LabsLand (Angulo et al., 2017) cuenta con un robot real encerrado en un espacio, el cual los alumnos pueden programar a distancia. En la Figura 4.13 se muestran dos instancias del laboratorio desplegadas en las oficinas de LabsLand en Bilbao. Otras instancias están desplegadas en universidades de todo el mundo y son muy similares (por ejemplo, en la Universidad de Deusto (García-Zubía et al., 2018), en la Universidad de Fort Hare (Kwinana et al., 2020), en la UNAD (Buitrago et al., 2020) o en la UNED de Costa Rica. Este es un buen ejemplo para evaluar la solución propuesta de detección de fallos, ya que puede presentar todos los casos de fallo mencionados anteriormente. Se trata de un laboratorio relativamente complejo con partes mecánicas y múltiples fuentes potenciales de fallo, como pueden ser la acumulación de polvo en sensores de distancia, rotura de los motores o engranajes mecánicos, o fallos de la cámara de visión en tiempo real, entre otros.

Las Figuras 4.14 y 4.15 muestran la arquitectura de una instancia del laboratorio. El IDE en línea que los estudiantes utilizan para la programación está en la nube. El RLMS, que puede ser compartido con otros laboratorios, puede estar alojado en cualquier lugar (en el mismo espacio del laboratorio, o en el centro de cómputo de la institución que lo aloja, o incluso en la nube). El Servidor del Laboratorio está implementado sobre un *single-board computer*, en este caso un Raspberry Pi 3, y proporciona la lógica específica del robot Arduino: controla el hardware del robot, gestiona la programación y, en este caso, incluso proporciona la UI a través de la cual los estudiantes acceden al robot. El Hardware del Experimento es el Arduino interno que es programado por el estudiante a través de la Raspberry Pi 3 y está conectado directamente a los sensores y actuadores. La cámara web proporciona una vista del robot, y el servidor interactivo de transmisión en vivo es el componente que ofrece esa vista a los usuarios.

Algunas de las pruebas que se realizan para verificar que determinadas instancias están en funcionamiento son las siguientes:

Figura 4.13: Laboratorio de robótica Arduino desplegado en las instalaciones de LabsLand en Bilbao.

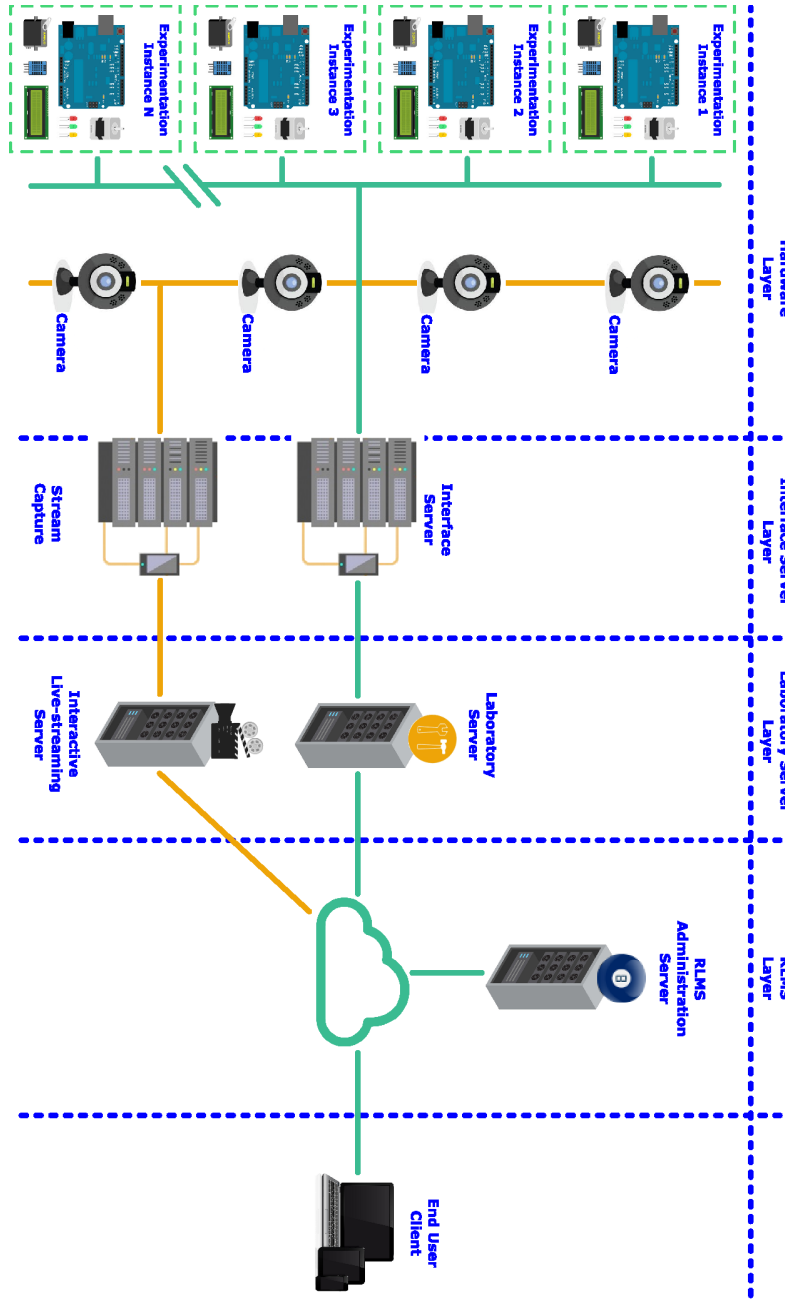


Figura 4.14: Visión general de la arquitectura multi-instancia dividida en varias capas de alto nivel. De (Villar-Martínez et al., 2019).

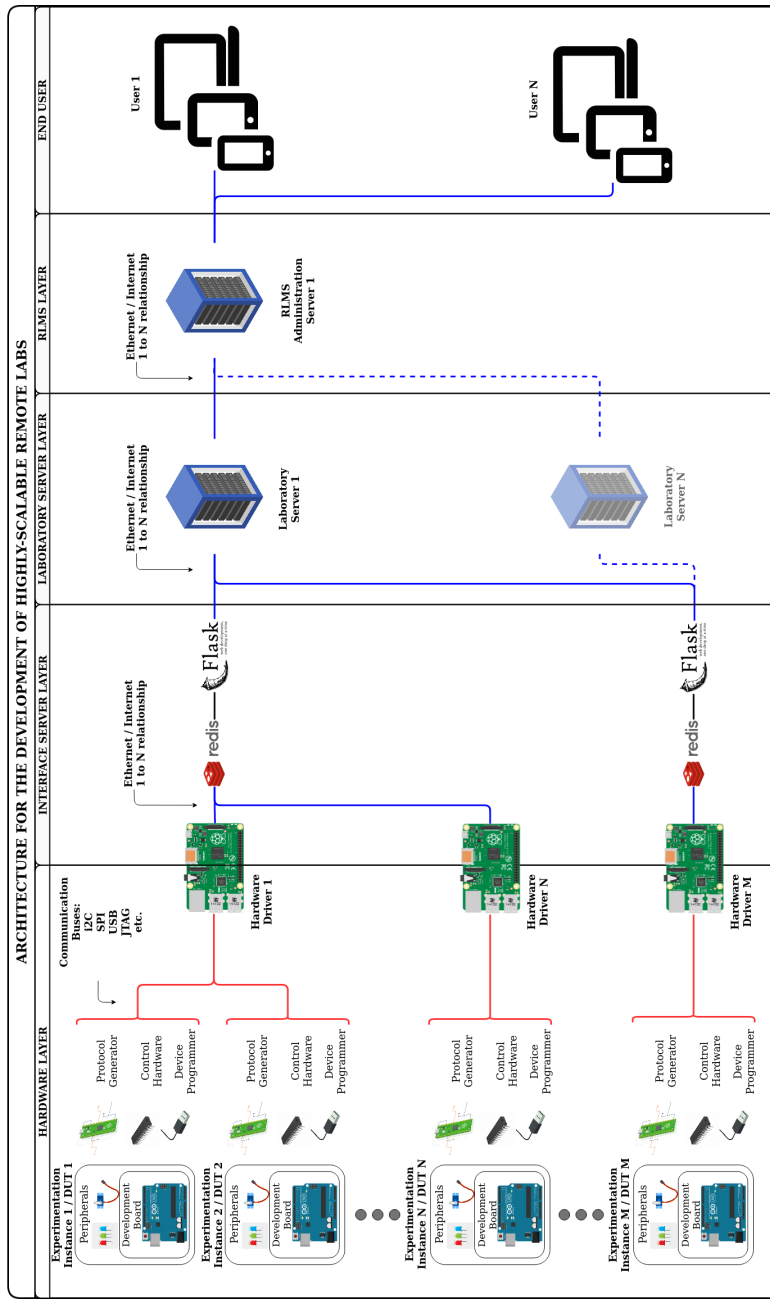


Figura 4.15: Vista detallada de la arquitectura multi-instancia dividida en varias capas. De (Villar-Martínez et al., 2019).

Primera capa (captura de excepciones)

Consiste en capturar todas las excepciones y, por lo general, asumir que las que no se manejan o las que no se pueden recuperar implican que la instancia particular no está funcionando. Algunos fallos que generan excepciones y que implican marcar el laboratorio como fallido son los dos siguientes:

- Fallo en la programación del microcontrolador Arduino.
- Fallo al modificar un valor de un periférico de entrada emulado (por ejemplo, la pulsación de un botón).

Segunda capa *healthchecks genéricos*

Se ejecutan periódicamente (cada 5-15 minutos). Se pueden configurar, pero no tienen ninguna lógica específica para el experimento. Por ejemplo, la comprobación de la cámara web podría probar la cámara web de este experimento específico, o la cámara web de cualquier otro.

- Se comprueba que la cámara web responde.
- Se comprueba que el dispositivo Raspberry Pi responde al *ping*.
- Se comprueba que el servidor HTTP dentro de la Raspberry Pi está respondiendo con un código de éxito 200.
- Se comprueba que la cámara web está proporcionando imágenes aparentemente válidas (archivos JPEG válidos).

Tercera capa (healthchecks específicos del experimento)

Estos también se ejecutan periódicamente, pero son específicos para cada experimento. Por ejemplo:

- Verificar que el Arduino que está conectado a través de USB a la Raspberry Pi es detectado como un host USB con el ID correcto.

Cuarta capa (Pruebas de funcionamiento)

Estos se ejecutan más raramente (una vez cada pocas horas). Se reserva la instancia del laboratorio, y se programa un binario específico de Arduino en la placa, que debe hacer que el robot siga la línea. Simultáneamente, el sistema basado en la visión por ordenador que se describió anteriormente (véase la Figura 4.12) realiza un seguimiento de la posición del robot en tiempo real. Así, el sistema verifica:

- Que el robot siga la línea de manera correcta.
- Que el circuito se complete en un tiempo determinado.
- Que el robot esté parado antes de que comience el programa, y se detenga después de que éste termine.

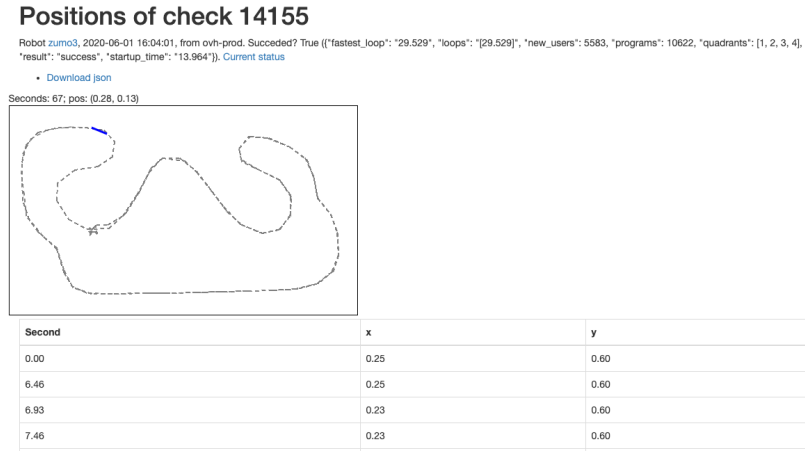


Figura 4.16: Panel de información de la prueba del Robot LabsLand Arduino.

La principal desventaja de este tipo de pruebas es que el experimento tiene que ser reservado y el robot tiene que ejecutarlo durante un tiempo. Además, implica el uso activo del robot, por lo que si se ejecutara miles de veces, esto podría suponer un desgaste extra del robot. Por este motivo, suele ejecutarse con menos frecuencia que otras pruebas. La ventaja, sin embargo, es que es muy completa. Si el robot no funciona como se espera, lo más probable es que

la prueba sea capaz de detectarlo, independientemente del componente preciso que esté causando el fallo. La Figura 4.16 muestra la interfaz de usuario que muestra el resultado de una prueba (exitosa). El gráfico de arriba muestra los movimientos del robot durante la prueba, seguidos automáticamente a través de la visión por ordenador.

4.3.3 **Arquitectura de laboratorio remoto orientada al despliegue de múltiples instancias de experimentación**

El segundo paso, tras detectar que un laboratorio remoto está fuera de línea, es la capacidad de redirigir a los usuarios sin problemas a otra instancia (una copia) del mismo laboratorio. Así, los usuarios siempre acceden a una instancia del laboratorio que funciona, no experimentan ningún error y perciben que el laboratorio es fiable. Para que este modelo sea posible es necesario disponer de varias instancias de laboratorio similares y de un sistema que pueda hacer un seguimiento de las instancias que están funcionando y dirigir a los usuarios que llegan a las que están funcionando y disponibles.

Para conseguir una fiabilidad basada en la replicabilidad (múltiples copias del mismo experimento gracias a la solución presentada de escalabilidad) es necesario utilizar arquitecturas que permitan controlar una serie de instancias de laboratorio de forma eficiente y con un coste limitado. Mantener un coste bajo facilita que haya más instancias. Estas instancias pueden utilizarse no sólo para proporcionar una mayor fiabilidad, sino también para dar servicio a un mayor número de usuarios de manera simultánea. La arquitectura propuesta (Villar-Martínez et al., 2019) en el Capítulo 3 de esta tesis, permite la creación de laboratorios remotos multi-instancia basados en los principios de alta escalabilidad, flexibilidad de experimentos remotos y bajo coste. Esta arquitectura se basa parcialmente en la solución propuesta en la tesis doctoral de Ignacio Angulo Martínez (Angulo et al., 2018; Angulo, 2015).

La citada arquitectura se divide en cuatro capas principales para poder controlar varias instancias de experimentación de un laboratorio remoto. La Figura 4.14 muestra una visión general de la arquitectura, con los detalles de las capas y los componentes. En la Figura 4.14 se muestra una placa de desarrollo de microcontroladores *Arduino UNO* con un conjunto de periféricos, aunque es adaptable a cualquier otro dispositivo programable. A continuación se detallan las capas que forman la arquitectura.

La capa de nivel más bajo de la arquitectura, que es la Capa de Hardware, engloba todos los dispositivos de hardware necesarios para gestionar adecuadamente el experimento remoto del laboratorio remoto. Normalmente un *single-board computer* acoplado a una serie de componentes hardware es capaz de interactuar correctamente con el experimento remoto. Para adaptarse a otras configuraciones de laboratorio, esta capa podría basarse en un ordenador tradicional o incluso en un chasis PXI. El propósito de esta capa es replicar las acciones físicas que el usuario realizaría en un laboratorio práctico. Este experimento remoto, que es el centro del laboratorio remoto, también forma parte de la capa de hardware.

En segundo lugar, la Capa de Servidor de Interfaz es una entidad encargada de abstraer toda lo relacionado con el control y el hardware a través de una API REST simplificada. Se trata de una capa de software que puede ser desplegada en el citado dispositivo hardware o en otros dispositivos específicos. Su objetivo es almacenar y distribuir todas las tareas de control del laboratorio, que representan acciones a realizar sobre el experimento remoto, a los componentes hardware existentes.

Estas tareas se reciben a través de un servidor web que expone una API REST, mediante peticiones GET y POST, y se guardan en una estructura de datos FIFO. A continuación, las tareas se distribuyen secuencialmente a los dispositivos disponibles de la capa de hardware para ser ejecutadas. Tanto el almacenamiento como la distribución de las acciones se basan en Redis (Carlson, 2013), un componente mixto de *in-memory data store /message broker*. Como se puede ver en la Figura 4.15, los diferentes objetos de la Capa de Servidor de Interfaz se conectan a través de una conexión TCP utilizando el protocolo HTTP. Esto amplía las capacidades de interconexión, permitiendo el uso de diferentes topologías y diferentes números de dispositivos para adaptarse mejor a los objetivos de despliegue, como la capacidad, la modularidad o el coste.

En tercer lugar, la Capa de Servidor de Laboratorio es un subsistema encargado de soportar la interacción requerida entre el usuario y el experimento. Este componente soporta el cliente basado en la web que se muestra al usuario, que incluye un *stream* de vídeo en tiempo real del experimento y los diferentes controles virtuales del experimento remoto. Esta entidad también se comunica con la Capa de Servidor de Interfaz para propagar las acciones del usuario hacia abajo y para recuperar el estado e información sobre el experimento.

En cuarto lugar, el RLMS es una entidad encargada de controlar todas las tareas administrativas necesarias para gestionar las diferentes sesiones de laboratorio concurrentes que puedan surgir. Estas tareas incluyen la autorización de usuarios, las reservas, las colas o el balanceo de carga, entre otras. También sirve como herramienta de integración, actuando como puente entre el laboratorio remoto y los diferentes sistemas de gestión del aprendizaje o LMS desde los que acceden los alumnos.

Por último, en paralelo a los otros cuatro subsistemas, se despliega la Plataforma Interactiva de Live-Streaming, WILSP ya descrita con anterioridad. Este componente, que también está formado por varias capas para garantizar la escalabilidad y la fiabilidad, se encarga de proporcionar al usuario del laboratorio remoto un *stream* de vídeo en tiempo real, que es capturado in-situ por una cámara. Esta plataforma hace abstracción de las peculiaridades específicas de la cámara y proporciona estas funcionalidades de forma fiable.

La capa Sistema de Gestión Remota del Laboratorio o RLMS, mencionada anteriormente, es la encargada de controlar a los usuarios y acceder a las instancias del laboratorio. Esta capa dispone de la información generada por los métodos de detección de fallos explicados en la Subsección 4.3.1, y en base a ella es capaz de redirigir a los usuarios a las diferentes instancias disponibles. Si todas las instancias están disponibles, los usuarios serán distribuidos entre ellas, de forma transparente. Si alguna de las instancias no está disponible, gracias a los métodos de detección activa, los usuarios sólo se distribuirán entre las instancias disponibles, permitiendo una experiencia de experimentación fluida. En el caso de que todas las instancias disponibles fallen, independientemente de la gravedad del fallo, el usuario sería avisado antes de acceder al laboratorio.

Este modelo escalable tiene, por tanto, dos grandes ventajas: en primer lugar ofrece fiabilidad, pero también hace posible que múltiples usuarios utilicen el laboratorio al mismo tiempo. La capacidad dependerá del número de instancias de experimentación disponibles. En el caso de un aula, si el número de instancias es lo suficientemente alto, permite a todos los alumnos utilizar el laboratorio remoto de manera simultánea. El número de instancias que se requiere para ello suele ser significativamente menor que el número total de alumnos de la clase, mediante el uso de las estrategias presentadas en la tesis doctoral de Pablo Orduña Fernández (Orduña, 2013).

Cuando todas las instancias funcionan correctamente, la capacidad del la-

boratorio en términos de posibles usuarios concurrentes es alta. Si alguna de las instancias del laboratorio falla, entonces la relación entre los usuarios en cola y las instancias disponibles se recalcula automáticamente. Esto mantiene el laboratorio disponible para todos los usuarios, que además no perciben que algunas instancias del laboratorio puedan estar defectuosas.

4.3.4 **Despliegue de la solución propuesta para detección de fallos en el laboratorio remoto de Arduino de LabsLand**

Un ejemplo de laboratorio remoto basado en la solución propuesta WebLabPRO es el laboratorio remoto de Arduino de LabsLand. Este laboratorio es escalable y tolerante a fallos. Se centra en proporcionar a los usuarios la capacidad de programar e interactuar con una placa de desarrollo Arduino UNO, que se basa en un microcontrolador AVR ATmega328P. Este laboratorio cuenta con multiples instancias de experimentación repartidas por el mundo, con el fin de mitigar puntos únicos de fallo, como las caídas de red o de alimentación.

Cada instancia de experimentación está formada por la placa Arduino UNO Rev3 y una serie de periféricos, que incluyen diodos LED, potenciómetros emulados, una consola de comunicaciones serie, interruptores emulados, botones emulados, una pantalla OLED y un servomotor. Estos periféricos están preconfigurados y ya conectados de una manera específica, ya que los usuarios no pueden conectar estos componentes por sí mismos porque el sistema se encuentra en un laboratorio remoto. El diagrama, que se muestra en la Figura 4.17, se da a los usuarios de antemano para que puedan programar la placa Arduino en consecuencia. Se emulan los periféricos de entrada, como interruptores, botones y potenciómetros. El usuario controla los periféricos virtuales en el panel de control y mediante el uso de dispositivos de hardware (por ejemplo, convertidores digital-analógico, expansores de puertos, etc.) se transmiten las señales eléctricas equivalentes hacia y desde la placa Arduino. Estos dispositivos de hardware forman parte de los dispositivos de la Capa de Hardware, que permiten al usuario reproducir esas señales eléctricas in-situ a través de Internet.

Para facilitar el despliegue del laboratorio en cada lugar se han agrupado las cuatro instancias del laboratorio en una única estructura física. En la Figura 4.18 se muestra la estructura física que contiene la PCB principal con las

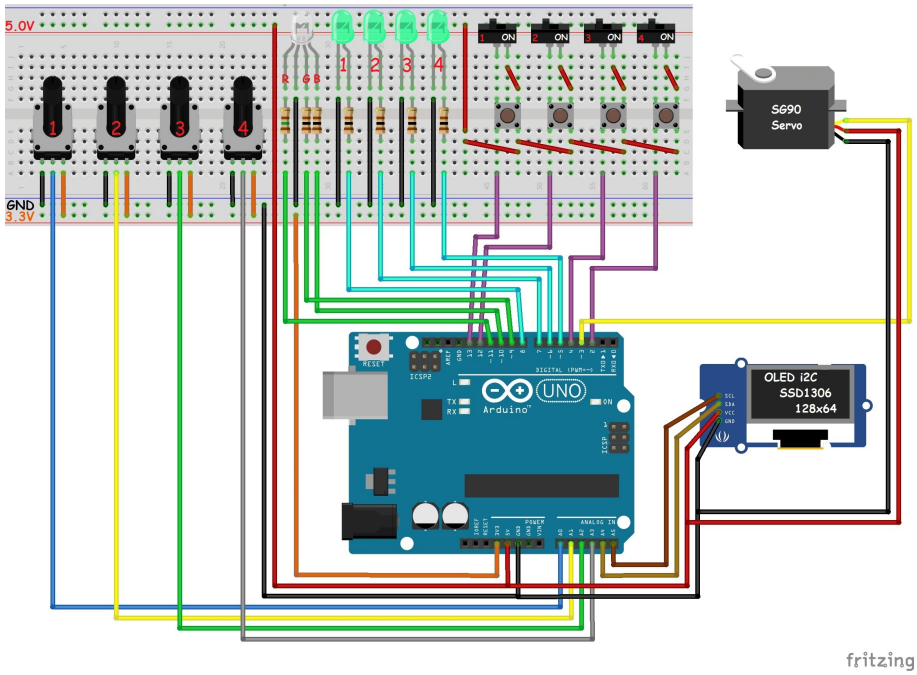


Figura 4.17: Esquema de Fritzing puesto a disposición de los usuarios del laboratorio remoto Arduino de LabsLand.

cuatro placas Arduino UNO, sus periféricos y las piezas de hardware necesarias para controlar cada montaje del laboratorio. La estructura también sirve de soporte para dos ordenadores monoplaca Raspberry Pi 3, una cámara, una fuente de alimentación y una tira de LED para la iluminación. El ordenador monoplaca Raspberry Pi 3 actúa como plataforma de hardware para soportar el *capa de servidor de interfaz*. La cámara única es capaz de grabar las cuatro instancias al mismo tiempo, y la plataforma de streaming es capaz de cortar la configuración de cada laboratorio y proporcionar a cada usuario el *stream* de vídeo correcto de cada laboratorio. La arquitectura del laboratorio remoto también proporciona flexibilidad y es capaz de compartir recursos, lo que permite un menor coste de despliegue. Actualmente existen numerosos laboratorios remotos similares al laboratorio remoto Arduino de LabsLand en cuanto a los dispositivos de capacidad de experimentación utilizados (placas de desarrollo basadas en Arduino y sistemas de control basados en Raspberry Pi), pero algunos de ellos (Alexander and Radhakrishnan, 2015; Mostefaoui and Benachenhou, 2015; Fernández-Pacheco et al., 2019; Costa et al., 2020) tienen una única instancia de experimentación, no están basados en una arquitectura con capacidad de compartir recursos y/o no utilizan un RLMS como herramienta de gestión desacoplada.

En relación con las capas de estructura mencionadas, este laboratorio remoto puede describirse de la siguiente manera:

- **Capa de Hardware:** Está formada tanto por el experimento remoto como por el hardware de control. En este caso, cuatro experimentos remotos (Las placas Arduino UNO más los diodos LED, el servomotor, la pantalla, etc.) conviven en una PCB común que proporciona la interconexión eléctrica entre los dispositivos, así como el soporte físico. La PCB se puede ver en detalle en la Figura 4.19. El hardware de control se compone de una serie de dispositivos, como expansores de puertos, convertidores digital a analógico, programador y otros dispositivos necesarios para propagar las interacciones entre el usuario y el experimento remoto, y viceversa.

- **Capa de Servidor de Interfaz:** Esta capa está limitada desde el punto de vista del hardware a un único *single-board computer* Raspberry Pi 3. Desde el punto de vista del software contiene tres componentes principales, que son el Controlador de Hardware, el *message broker* y

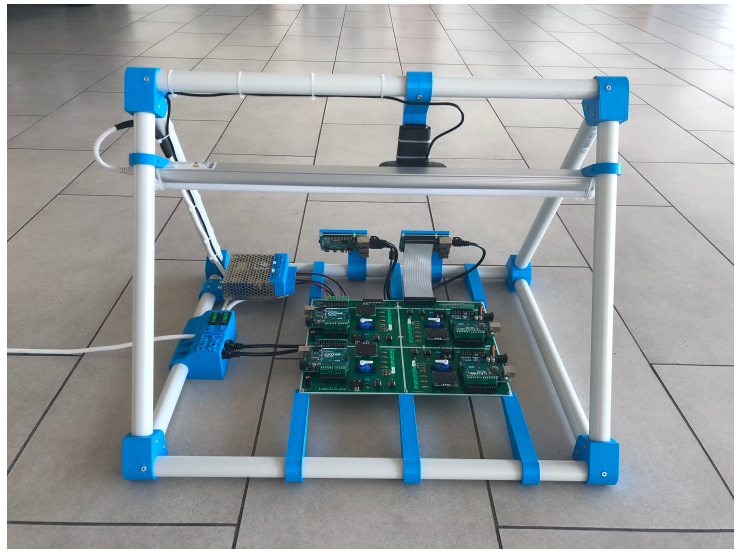


Figura 4.18: Vista general de la estructura del laboratorio para experimentación con Arduino.

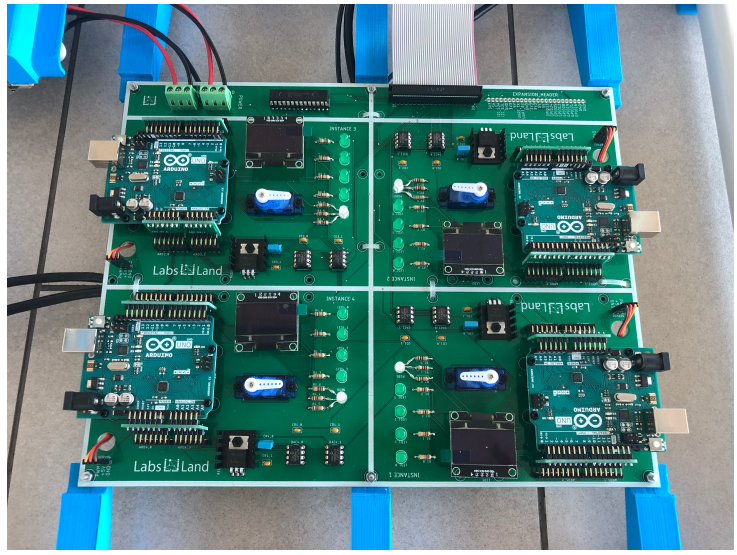


Figura 4.19: Vista detallada de la tarjeta de laboratorio Arduino Board.

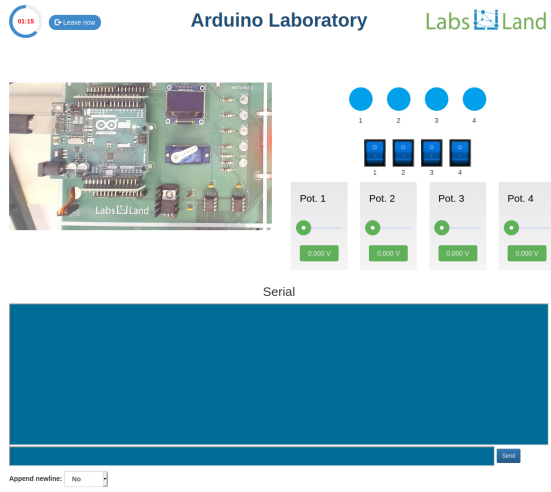


Figura 4.20: Panel de control del experimento de laboratorio de la placa Arduino.

el Servidor Web de Interfaz. Esta capa transforma las consultas HTTP en los comandos específicos necesarios para controlar los dispositivos de hardware desde la capa inferior. También captura los datos del experimento y los propaga hacia arriba. Redis [1] se utiliza para el *message broker* y el servidor de la interfaz se basa en Flask [2].

- **Capa de Servidor de Laboratorio:** Esta capa sólo de software se encarga de proporcionar al usuario el cliente web específico del laboratorio, con todas las tareas que ello conlleva. Sirve de soporte para todos los controles emulados (botones, interruptores, potenciómetros deslizantes), para el *stream* de vídeo en tiempo real del montaje, para la consola de comunicación serie y para el editor de código Arduino. En la Figura 4.20 se muestra el mencionado panel de control. Como se puede observar, el usuario no es consciente de que existen diferentes instancias de un mismo experimento, ya que el sistema RLMS en conjunción con esta capa hace transparente el control de cada instancia.

[1] https://redis.io/
[2] https://flask.palletsprojects.com/en/1.1.x/

- **Capa del sistema de gestión remota del laboratorio:** Al igual que en el ejemplo anterior del Robot Arduino, el RLMS para este laboratorio se encuentra en la nube. Diferentes ubicaciones utilizan diferentes componentes RLMS, ya que este componente se despliega en función de las ubicaciones y no de los laboratorios.
- **Plataforma interactiva de streaming en tiempo real:** Paralelamente, este componente multicapa también se despliega junto con el laboratorio. Desde el punto de vista del hardware, utiliza una cámara USB y una Raspberry Pi 3 como elemento de captura, así como diferentes servicios en servidores en la nube para procesar las imágenes.

El objetivo de esta distribución por capas es conseguir una herramienta de experimentación remota de doble uso. Disponer de múltiples instancias de un mismo experimento permite aumentar la capacidad del laboratorio en un momento dado, y proporciona al usuario una sesión de experimentación satisfactoria, incluso cuando una o varias instancias se encuentran inoperativas.

Para llevar a cabo lo anterior es importante detectar cuando una instancia concreta está fallando. Para ello este laboratorio también hace uso de algunas de las capas de detección de fallos explicadas en la Sección 4.3.1. En este caso se utilizan la primera capa (*captura de excepciones*), la segunda capa (*healthchecks genéricos*) y la tercera capa (*healthchecks específicos*). En este caso de uso, la cuarta capa de la solución propuesta no se implementa, ya que el laboratorio remoto cuenta con un menor numero de componentes hardware. Esto minimiza los potenciales casos de fallo.

4.4 Validación de la solución propuesta para fiabilidad en experimentación remota

En la Sección 4.2 se analizaron cinco laboratorios remotos y sus experimentos remotos asociados y se ha determinado que, efectivamente, existen carencias respecto a su fiabilidad. A continuación, se ha descrito una arquitectura de varios niveles orientada a detectar los problemas de fiabilidad y a mitigarlos mediante diferentes enfoques (Sección 4.3), el principal de los cuales se basa en la tolerancia a fallos mediante la replicación. Para validar ese enfoque se han introducido esas mejoras en dos experimentos remotos específicos para analizar su viabilidad. El objetivo de validación es, por tanto, asegurar que

mediante estas mejoras se puede conseguir una fiabilidad y disponibilidad significativamente mejoradas para estos dos experimentos, de forma que en el futuro otros investigadores y desarrolladores de laboratorios puedan también mejorar la fiabilidad de sus propios experimentos remotos mediante enfoques similares.

En esta sección se trata de evaluar desde un punto de vista técnico si se ha cumplido el objetivo de mejorar la fiabilidad de un laboratorio remoto utilizando las soluciones propuestas en la Seccion 4.3. Los resultados obtenidos en el periodo de evaluación de ambas soluciones sugieren que, efectivamente, el objetivo se ha cumplido. Para ello se han diseñado dos experimentos remotos adicionales, fuera del análisis incluido en la Sección 4.2, de acuerdo con la metodología indicada en la Sección 4.3. Ambos experimentos remotos fueron probados, durante un periodo de 16 días, desde el 12/07/2020 hasta el 27/07/2020 en el caso del laboratorio de robótica de LabsLand, y durante un periodo de 5 meses, desde el 01/05/2020 hasta el 30/09/2020 en el caso del laboratorio Arduino de LabsLand.

4.4.1 **Validación de la arquitectura de fiabilidad en el laboratorio remoto de robótica de LabsLand**

El primer laboratorio que utiliza las dos soluciones propuestas es el Laboratorio de robótica de LabsLand. Como se muestra en la Sección 4.3, este laboratorio permite al usuario controlar remotamente un robot basado en la plataforma Arduino, que puede ser controlado sobre una plataforma rectangular (ver Figura 4.21).

Este laboratorio utiliza las dos soluciones propuestas en esta tesis, escalabilidad y fiabilidad, para garantizar la fiabilidad para los estudiantes, los profesores y otros posibles usuarios del laboratorio. En primer lugar utiliza la arquitectura mencionada en la Sección 4.3 para gestionar diferentes instancias equivalentes de forma transparente, y en segundo lugar utiliza una serie de pruebas de detección de fallos en diferentes capas para detectar si una instancia específica falla.

Como se menciona en la Sección 4.3, se realizan diferentes tipos de pruebas dependiendo del tipo de laboratorio. En este caso, por sus características, es posible realizar pruebas periódicas en las cuatro capas descritas anteriormente. La prueba más exhaustiva de todas ellas es la prueba de funcionamiento. En

Figura 4.21: Laboratorio de robótica LabsLand Arduino desplegado en la Universidad de Fort Hare, en Sudáfrica.

esta prueba, un usuario virtual controlado por ordenador sigue todos los pasos que realizaría un usuario real. Este usuario virtual hace una reserva en el gestor del laboratorio, accede a él, programa el robot, ejecuta el código, que hace que el robot siga un determinado trazado de circuito, captura una serie de datos y abandona el laboratorio. La información obtenida permite al sistema recoger información de estado sobre el comportamiento de todas las capas del laboratorio, desde el sistema de gestión de colas hasta el funcionamiento del hardware específico del laboratorio. Además, cuando se detecta un fallo crítico (el robot no sigue el circuito correctamente, lo hace pero tarda más de lo debido, no es posible observar el robot o no es posible conectarse al laboratorio), desactiva automáticamente esa instancia concreta hasta la siguiente ejecución de la prueba con éxito. El resultado de cada prueba se recoge en un panel de información (mostrado en la Figura 4.16) que puede ser consultado por quienes desarrollan o mantienen el laboratorio. Para cada una de las instancias, se realizan tres pruebas diarias, a intervalos de ocho horas. Este número de pruebas diarias permite obtener suficiente información sobre el estado del robot y, al mismo tiempo, no causa fatiga para el hardware del experimento. Un número mayor de pruebas diarias acumularía demasiados usos, lo que podría dañar el robot prematuramente.

Durante los dieciséis días de evaluación, se probaron diariamente las cinco instancias operativas del robot. La Figura 4.22 muestra los resultados de cada prueba (T1, T2 y T3) para cada instancia. La primera conclusión significativa que se puede extraer es que todos los robots han fallado en algún momento. Los robots contienen motores, engranajes y otros componentes mecánicos, por lo que es de esperar que se produzcan fallos. La segunda conclusión significativa es que, a pesar de que todos los robots han fallado en el periodo de tiempo en cuestión, ha sido posible mantener el servicio en funcionamiento, aunque con una capacidad limitada. El peor día de este periodo de evaluación fue el 20/07/2020, cuando tres de las instancias estuvieron completamente inoperativas durante todo el día, y otras dos instancias estuvieron inoperativas durante al menos un tercio del día. Incluso en esta situación, el servicio no estaba inoperativo desde la perspectiva de los usuarios, ya que al menos una de las instancias funcionaba correctamente.

Por lo tanto, aunque en esta situación los usuarios podrían haber tenido que esperar más tiempo para acceder, incluso en este peor escenario, el servicio estaba en línea y funcionando, y los usuarios con clases planificadas utilizando

Date	Test run	Instance 1	Instance 2	Instance 3	Instance 4	Instance 6	Available instances	Laboratory capacity	Laboratory online
07/12/2020	T1	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T2	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T3	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
07/13/2020	T1	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T2	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T3	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
07/14/2020	T1	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T2	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T3	Pass	Pass	Pass	Pass	Fail	4 of 5	80.00%	Yes
07/15/2020	T1	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T2	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T3	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
07/16/2020	T1	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T2	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T3	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
07/17/2020	T1	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T2	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T3	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
07/18/2020	T1	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
	T2	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
	T3	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
07/19/2020	T1	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
	T2	Fail	Fail	Pass	Pass	Fail	2 of 5	40.00%	Yes
	T3	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
07/20/2020	T1	Fail	Fail	Pass	Pass	Fail	2 of 5	40.00%	Yes
	T2	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
	T3	Fail	Fail	Fail	Fail	Pass	1 of 5	20.00%	Yes
07/21/2020	T1	Fail	Fail	Pass	Pass	Fail	2 of 5	40.00%	Yes
	T2	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
	T3	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
07/22/2020	T1	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
	T2	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
	T3	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
07/23/2020	T1	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
	T2	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
	T3	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
07/24/2020	T1	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
	T2	Fail	Fail	Pass	Pass	Pass	3 of 5	60.00%	Yes
	T3	Fail	Pass	Pass	Pass	Pass	4 of 5	80.00%	Yes
07/25/2020	T1	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T2	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T3	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
07/26/2020	T1	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T2	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T3	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
07/27/2020	T1	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T2	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes
	T3	Pass	Pass	Pass	Pass	Pass	5 of 5	100.00%	Yes

Figura 4.22: Resultados de cada una de las instancias operativas para cada día del periodo de evaluación.

el laboratorio no habrían necesitado suspender sus actividades. Con una carga de usuarios suficientemente alta, el servicio en ese caso podría ser más lento, pero el laboratorio seguiría siendo fiable y se percibiría como tal.

Como puede verse en la parte derecha de la Figura 4.22, la capacidad del laboratorio fue del 80 % o más durante más de la mitad de las franjas horarias de 8 horas, y nunca estuvo completamente inoperativo. Este aspecto pone de manifiesto la importancia de disponer de varias instancias intercambiables del laboratorio. Normalmente, las instancias del laboratorio fallan de forma aleatoria, por lo que al tenerlas replicadas, es muy poco probable que todas las instancias fallen a la vez. Esto se vuelve más improbable a medida que aumenta el número total de instancias desplegadas.

En conclusión se puede considerar que las soluciones propuestas han mejorado la fiabilidad del servicio de laboratorio en su conjunto. No disponer de varias instancias habría provocado una indisponibilidad de varios días, lo cual es insostenible para el uso diario en clase por parte de profesores y alumnos. Poder detectar automáticamente los diferentes fallos en el robot habría dificultado las sesiones de experimentación de los usuarios y proporcionado una mala experiencia de uso, ya que los usuarios habrían estado expuestos a los fallos. Gracias al uso de estas soluciones, el laboratorio ha permanecido disponible y funcionando correctamente, y además, el equipo de mantenimiento del laboratorio ha sido notificado cuando se han producido los diferentes fallos, lo que minimiza el tiempo de reparación de la instancia de laboratorio.

4.4.2 **Validación de la arquitectura de fiabilidad en el laboratorio remoto Arduino de LabsLand**

Como se menciona en la Sección 4.3, el laboratorio remoto Arduino de LabsLand basa su fiabilidad mejorada en la combinación de técnicas de detección de fallos y la creación de varias instancias del mismo experimento. Algunas de las instancias comparten parte del hardware para reducir el coste, y el hardware que se comparte es, por diseño, piezas que son mayoritariamente fiables. El laboratorio está actualmente desplegado en cuatro lugares diferentes (Universidad de Deusto en Bilbao, España, oficinas de LabsLand en Bilbao, España, Universidad de Fort Hare en Alice, Sudáfrica), y en la UNED en Costa Rica, con cuatro instancias en cada lugar. Los datos utilizados para este análisis provienen de los dos despliegues ubicados en Bilbao, con un total de ocho

instancias.

En cuanto a las técnicas de detección de fallos, cabe mencionar que en este laboratorio sólo se utilizan las capas uno, dos y tres de las mencionadas en la Subsección 4.3.1. El montaje de experimentación de este laboratorio es más sencillo que el del laboratorio de robots y contiene menos piezas mecánicas, que son la principal causa de fallos en ese laboratorio. Además, el hardware de este laboratorio se ha diseñado teniendo en cuenta un posible mal uso por parte del usuario, lo que lo hace más robusto y menos propenso a los fallos.

Las pruebas de las capas uno, dos y tres, al estar basadas en el software, pueden realizarse con una alta frecuencia, lo que minimiza el tiempo necesario para detectar posibles fallos. En caso de detectarse un fallo en una instancia concreta, ésta se desconecta automáticamente y los usuarios se redistribuyen entre el resto de instancias activas de forma transparente. Además, se envía un correo electrónico al personal de mantenimiento para garantizar una comunicación eficaz.

Debido a que este laboratorio no dispone de dispositivos hardware tan delicados como en el caso del laboratorio de robots, el número de incidencias detectadas es menor. En la Figura 4.23 se recogen las incidencias detectadas a lo largo del periodo de pruebas. Cada fila de la tabla muestra la fecha en la que se produjo la incidencia, el motivo de la misma, las instancias afectadas por la misma y el tiempo en minutos durante el cual la instancia no estuvo disponible.

Como puede verse, un número importante de las incidencias duró tres minutos o menos. Por regla general, estos errores de tan corta duración suelen ser glitches, errores puntuales de comunicación o timeouts, y tienden a autocorregirse. La detección de estos errores es posible gracias a la alta frecuencia con la que se realizan las pruebas. Si las pruebas se hubieran realizado manualmente por el equipo de mantenimiento del laboratorio, muchos de estos errores habrían pasado desapercibidos. Durante el periodo de pruebas también surgieron otras incidencias que tardaron más tiempo en resolverse, cerca de una hora, pero fueron menos numerosas.

Las causas de las incidencias detectadas estaban relacionadas, en su mayoría, con la estabilidad de las cámaras, con la estabilidad de la red y, en unos pocos casos, con el despliegue de nuevo código en algunas de las capas. A lo largo del periodo de pruebas, no hubo errores causados por el hardware específico del laboratorio, y no hubo que sustituir ningún componente electrónico. A ello contribuye el hecho de que se trata de un laboratorio sencillo en

cuanto a componentes mecánicos (sólo utiliza un servomotor, que únicamente gira para dar respuesta visual al usuario) y su hardware se diseñó teniendo en cuenta el posible mal uso por parte del usuario durante las sesiones de experimentación.

En ningún caso el laboratorio estuvo totalmente indisponible. En los peores días, la capacidad del laboratorio fue de al menos el 50 %, y sólo en cuatro de los 17 días en los que hubo fallos, éstos duraron más de tres minutos. En los días restantes del periodo de pruebas (136 de un total de 153), el laboratorio mantuvo una capacidad total del 100 %. Asimismo, incluso en los momentos de menor capacidad, la disponibilidad del laboratorio en su conjunto se mantuvo en el 100 %, ya que siempre hubo al menos una instancia funcionando correctamente.

Estos resultados refuerzan la idea de que es importante elegir una arquitectura de laboratorio que permita escalar el laboratorio a diferentes niveles. Gracias a ello, el laboratorio se ha desplegado con éxito en diferentes ubicaciones, lo que mejora su resistencia frente a puntos de fallo difíciles de controlar, como la inestabilidad de la conexión de red con el proveedor de Internet o los cortes de energía.

4.5 Conclusiones de la arquitectura de fiabilidad en experimentación remota

A pesar de los importantes avances tecnológicos conseguidos en el campo de la experimentación remota y de su probada validez como herramienta educativa, la adopción de los laboratorios remotos en la enseñanza no ha supuesto una progresión proporcional.

Uno de los principales obstáculos para la adopción de los laboratorios remotos por parte de los profesores es, sin duda, la falta de fiabilidad de muchos de los experimentos remotos disponibles.

Muy a menudo los proveedores de los laboratorios remotos son los equipos de investigación encargados de su desarrollo, entre los que suelen encontrarse profesores que promueven su uso en cursos de su propia institución educativa y, en muchas ocasiones, los comparten con la comunidad de forma totalmente altruista.

La complejidad de los laboratorios remotos, que incluyen componentes de software y hardware, requiere un mantenimiento constante para garantizar

Date	Fault type	University of Deusto				LabsLand Offices			
		Instance 1	Instance 2	Instance 3	Instance 4	Instance 1	Instance 2	Instance 3	Instance 4
05/11/2020	Camera	1 min					1 min		
05/13/2020	Camera	2 min	2 min		2 min				
05/14/2020	Camera	1 min							
05/16/2020	Camera								
06/02/2020	Network	70 min	70 min	70 min	70 min				
06/08/2020	Network						1 min	1 min	
06/08/2020	Network						105 min	105 min	
06/12/2020	Network	1 min	1 min		1 min				
07/02/2020	Camera	2 min	2 min	2 min	2 min	2 min			2 min
07/08/2020	Network						2 min	2 min	
07/20/2020	Network				3 min		1 min	1 min	
07/21/2020	Development	3 min			60 min				
07/24/2020	Development	60 min	60 min	60 min		45 min	45 min	45 min	45 min
07/26/2020	Network								
08/14/2020	Network					1 min	1 min	1 min	1 min
08/18/2020	Network		2 min						
09/18/2020	Network		1 min	1 min					

Figura 4.23: Resumen de las incidencias registradas durante el periodo de pruebas del laboratorio remoto Arduino de LabsLand.

su correcto uso. La falta de control sobre los fallos experimentales perjudica la percepción de los usuarios del experimento al reducir la fiabilidad de la tecnología.

Aparte del coste, tanto en personal como en equipos, es necesario proporcionar técnicas que permitan la detección temprana de cualquier error de funcionamiento.

En este capítulo se han identificado los principales tipos de fallos derivados de las sesiones de experimentación remota, identificando la actuación óptima ante cada fallo para garantizar la fiabilidad de los laboratorios remotos.

Como resultado, esta tesis presenta una solución que da respuesta al problema de la fiabilidad de los laboratorios remotos mediante la integración de técnicas de detección de fallos y de escalabilidad. La primera solución comprende una arquitectura que soporta un conjunto de niveles de detección de fallos, que van desde el más bajo hasta el más alto de especificidad, diseñados para validar el correcto funcionamiento de los diferentes aspectos del experimento. Dada la heterogeneidad de los experimentos analizados, se han detectado fallos específicos derivados de la propia naturaleza de cada experimento, así como otros fallos genéricos que surgen globalmente con total independencia de las particularidades de los experimentos. Las capas menos específicas capturan las excepciones lanzando automáticamente peticiones HTTP para verificar que todos los recursos del laboratorio siguen en línea, mientras que las capas más específicas realizan pruebas de laboratorio, simulando sesiones reales de usuarios y validando los resultados. Estas pruebas permiten la detección automática de fallos en los laboratorios sin que éstos deban ser notificados por los usuarios.

Ambas soluciones se implementaron en dos laboratorios remotos de LabsLand para validar las mejoras en la fiabilidad de los experimentos. En el caso del laboratorio de robótica se observó que durante un periodo de dieciséis días se detectaron automáticamente numerosos fallos, sin comprometer la disponibilidad del laboratorio gracias al balanceo de carga entre las instancias de trabajo disponibles en cada momento.

A su vez, la implementación de las técnicas propuestas en el laboratorio remoto de Arduino permitió mantener el laboratorio en funcionamiento de forma continua durante un periodo de 5 meses, en los que se detectaron un total de 17 averías. La disposición distribuida de múltiples instancias en diferentes entidades de todo el mundo ha permitido que la disponibilidad del laboratorio

nunca haya caído por debajo del 50 % de su capacidad total.

Minimizar el tiempo de inactividad de un laboratorio remoto mejora sustancialmente la calidad del servicio y la experiencia del usuario. Se ha mostrado que las técnicas propuestas mejoran sustancialmente la calidad de los laboratorios remotos al aumentar la fiabilidad del laboratorio y la confianza de los usuarios.

La solución propuesta, e integrada en WebLabPRO, no evita ni reduce el número de fallos internos de los laboratorios remotos, pero sí reduce drásticamente su efecto sobre la disponibilidad del experimento remoto y reduce el número de fallos a los que están expuestos los usuarios, mejorando así la experiencia de éstos.

4.6 **Líneas futuras de fiabilidad en experimentación remota**

La detección de los defectos específicos de un laboratorio remoto ofrece una línea de investigación con muchas posibilidades. Los laboratorios remotos son sistemas que pueden analizarse utilizando las últimas tecnologías de mantenimiento predictivo desarrolladas en la industria para el análisis de los equipos utilizados en los procesos de producción. Las capacidades de diagnóstico de las tecnologías de mantenimiento predictivo han aumentado en los últimos años con los avances realizados en las tecnologías de detección.

Incluso en los laboratorios más complejos, con infinidad de variables que complican la repetibilidad de un experimento y dificultan la detección de errores en los datos aportados, la aplicación de técnicas de aprendizaje automático permite replicar automáticamente las sesiones experimentales y evaluar la validez de los resultados para detectar un error en el comportamiento del laboratorio.

Estas técnicas, combinadas con las capacidades de la visión artificial, proporcionan un campo de investigación abierto con vistas a desarrollar sistemas de detección de fallos adecuados para laboratorios remotos que impliquen el uso de cualquier tipo de dispositivo.

CAPÍTULO

5

Evaluación de la arquitectura WebLabPRO

UNA vez presentadas y validadas por separado las soluciones de escalabilidad y fiabilidad de WebLabPRO, en este capítulo se integran ambas para su validación mediante métricas ad-hoc. Esta validación, mas allá de una prueba de concepto, se ha llevado a cabo en un entorno real y no controlado facilitado por la empresa LabsLand. La arquitectura WebLabPRO se ha integrado en los laboratorios remotos de experimentación con FPGA de LabsLand, y mediante su uso se han obtenido los resultados de validación. Estos resultados se han obtenido utilizando los sistemas de trazabilidad y *learning analytics* proporcionados por la plataforma LabsLand. Esta validación transcurre durante un periodo superior a dos años, donde han participado centenares de estudiantes de diversos países y de distintos niveles educativos, que han llevado a cabo 72.377 sesiones de experimentación remota.

5.1 Introducción

Una vez diseñada e implementada la arquitectura WebLabPRO para mejorar la fiabilidad y la escalabilidad de los laboratorios remotos se debe evaluar su impacto en laboratorios reales.

El objetivo principal de este capítulo es evaluar si, aprovechando las aportaciones anteriormente mencionadas, es posible obtener laboratorios remotos que puedan ser realmente eficaces no sólo como prototipos de investigación, sino en entornos operativos educativos del mundo real, lo que eventualmente llevaría a su adopción generalizada.

La evaluación planteada en (Villar-Martínez et al., 2022) se llevará a cabo en laboratorios remotos desarrollados bajo la solución propuesta en esta tesis. Estos tienen múltiples copias y están desplegados en un mínimo de dos instituciones en diferentes países, apoyándose en una arquitectura de equilibrio de carga federada, como se define en (Orduña, 2013). Dado que el objetivo es evaluar si es posible la adopción generalizada para el uso multinstitucional en el mundo real, el estudio tendrá las siguientes características:

- Se centrará en un laboratorio remoto FPGA de LabsLand, el cual está desplegado en varias instituciones de todo el mundo y cuenta con docenas de instancias. Se utilizarán datos que abarca un periodo de más de dos años (736 días).

- Contará con la participación de estudiantes de múltiples universidades en varios países diferentes.

- Los profesores y alumnos implicados no serán contactados ni antes ni durante el proceso de evaluación.

Este capítulo se ha organizado de la siguiente manera: La Sección 5.2 describe el escenario de evaluación, mientras que la Sección 5.3 describe los criterios y métodos utilizados para analizar las soluciones para desarrollar laboratorios remotos escalables y fiables. Las Secciones 5.4 y 5.5 abordan la evaluación cuantitativa de WebLabPRO. La Sección 5.6 recoge y discute los resultados de un posterior análisis, centrado principalmente en la caracterización cualitativa del laboratorio remoto DE1-SoC FPGA de LabsLand. Por ultimo, la Sección 5.7 resume las conclusiones de este capítulo.

5.2 Escenario de la evaluación

La solución propuesta en esta tesis, por sus criterios de diseño, permite ser adaptada a cualquier laboratorio remoto, en especial si el objetivo de este es realizar prácticas con dispositivos programables.

Con el fin de lograr capturar datos de uso en condiciones de uso reales, la arquitectura diseñada se ha desplegado en múltiples laboratorios remotos de la empresa LabsLand. Específicamente, la evaluación aquí presentada se centra en el laboratorio remoto de FPGA, que será detallado en la siguiente subsección.

Es importante destacar que la evaluación se desarrolla en un escenario real, en absoluto controlado por el autor de esta tesis. Así pues, el análisis llevado a cabo se corresponde con escenarios reales de uso en aulas de distintas instituciones en diferentes países del mundo. En las subsecciones sucesivas se pormenorizan los detalles relativos a las dos versiones de laboratorios remotos de FPGA de LabsLand.

5.2.1 Laboratorio remoto de FPGA Intel DE1-SoC de LabsLand

Este laboratorio se basa en la placa de desarrollo Intel DE1-SoC, que está equipada con una FPGA Altera System-on-Chip Cyclone V. La propia placa de desarrollo cuenta con varios periféricos visuales, como LEDs o displays de 7 segmentos captadas por una cámara en tiempo real. LabsLand añade periféricos adicionales, como captura e inyección de audio, captura e inyección de vídeo, inyección de protocolos (teclado PS2, mando Nintendo N8, etc.), y botones y pulsadores virtualizados. Los usuarios de este laboratorio también pueden desarrollar su código VHDL o Verilog en el propio IDE de LabsLand sin tener que descargar, instalar o utilizar ningún software adicional. Sin embargo, pueden seguir sintetizando sus propios binarios a través de Quartus de Intel, si lo desean. En la actualidad, hay 62 instancias de este laboratorio en todo el mundo. Se encuentran en diversas instituciones de todo el mundo, en los siguientes países España (26) y Estados Unidos de América (36). En este caso, todo el hardware del laboratorio se fija en una estructura impresa en 3D, que se muestra en la Figura 5.1.

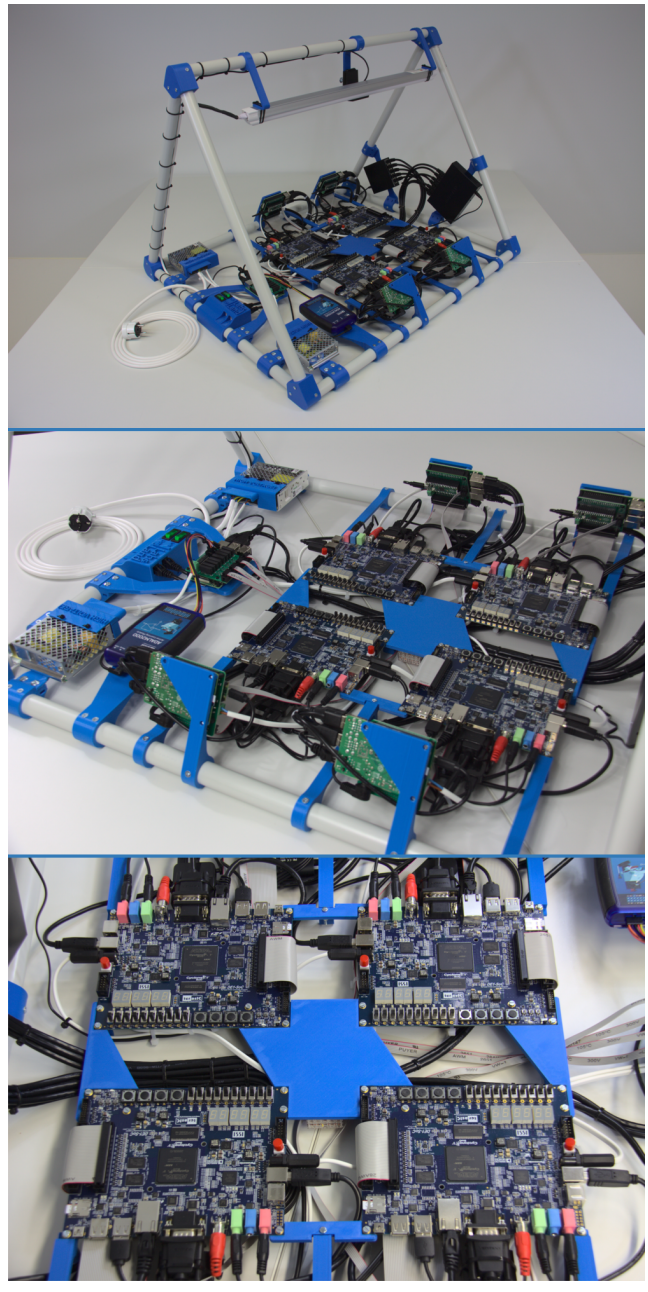


Figura 5.1: Múltiples vistas de la estructura física del laboratorio remoto Intel DE1-SoC FPGA de LabsLand.

5.2.2 Laboratorio remoto de FPGA Intel DE2-115 de LabsLand

El laboratorio LabsLand Intel FPGA DE2-115 permite a los usuarios programar y practicar con un Altera Cyclone IV en una placa de desarrollo Intel DE2-115. Los periféricos externos son similares a los del laboratorio SoC DE1, mientras que la propia placa de desarrollo tiene más LEDs, más displays de 7 segmentos y una pantalla de dos líneas de 16 caracteres de Hitachi LCD. Al igual que en el otro laboratorio FPGA, el código se puede desarrollar en el IDE LabsLand basado en la web.

En este caso, hay instancias distribuidas por todo el mundo en los siguientes países: España, Malasia, Brasil y Estados Unidos de América.

Con el fin de homogeneizar y mantener los costes de desarrollo de todos los laboratorios anteriores, se desarrolló un sistema de montaje modular que utiliza piezas impresas en 3D para soportar múltiples instancias de laboratorio en una única estructura física.

Esta estructura, que puede verse en la Figura 5.1, puede albergar 4 u 8 placas de desarrollo, dependiendo del tamaño, junto con fuentes de alimentación, switches de Ethernet, hardware de control, periféricos, iluminación y cámara. Estos componentes suelen compartirse entre las 4 u 8 placas de desarrollo, lo que minimiza los costes de desarrollo al no duplicar el hardware para cada instancia.

5.3 Metodología

El objetivo general de este capítulo es evaluar si los laboratorios remotos diseñados integrando las soluciones de escalabilidad y fiabilidad presentadas en esta tesis pueden convertirse en herramientas de enseñanza fiables y prácticas que están listas para su uso generalizado en el mundo real.

Adicionalmente, se presentarán los resultados de la evaluación cualitativa de la satisfacción de un grupo de alumnos, utilizando un cuestionario de satisfacción. Esta evaluación será explicada en detalle en la Seccion 5.6 ya que la evaluación más relevante para el objetivo de esta tesis es la cuantitativa.

La metodología se divide en dos fases. En primer lugar se comprueba si los laboratorios creados bajo la arquitectura WebLabPRO son capaces de escalar el número de instancias sin comprometer su funcionalidad y manteniendo los costes bajo control. En segundo lugar, si los laboratorios creados bajo la arquitectura WebLabPRO son realmente fiables.

Además, para asegurar que los resultados son realmente representativos y pueden confirmar si el uso generalizado en el mundo real es posible y efectivo, el estudio aplicará las características mencionadas en la Sección 5.1:

- Se centrará en un laboratorio remoto (el laboratorio LabsLand FPGA) que está desplegado en varias instituciones de todo el mundo y con docenas de instancias desplegadas, y se basará en datos que abarcan más de dos años (736 días).
- Contará con la participación de estudiantes de múltiples universidades en varios países diferentes.
- Los profesores que utilicen los laboratorios con sus alumnos no conocen que forman parte de este estudio.

5.4 Análisis cuantitativo del laboratorio

El análisis cuantitativo de usos del laboratorio remoto FPGA de Labsland examina si la arquitectura WebLabPRO, y los laboratorios remotos que la adoptan, cumplen los objetivos de escalabilidad y fiabilidad. Para ello, es necesario analizar si el laboratorio es capaz de soportar un elevado número de sesiones experimentales dentro de un periodo de tiempo determinado y cómo lo hace.

Para ello, se registraron y almacenaron diariamente varios parámetros relacionados con cada sesión experimental durante un periodo de 24 meses. Estos parámetros son:

- Fecha
- Número de instancias de experimentación disponibles
- Número de sesiones de experimentación realizadas
- Número de usuarios atendidos
- Posición máxima alcanzada en la cola de espera
- Tiempo máximo alcanzado en la cola de espera, en segundos

Es importante evaluar tanto la capacidad como la fiabilidad del laboratorio. Un laboratorio de gran capacidad es aquel que es capaz de realizar un gran número de experimentos en un tiempo determinado. Además, un laboratorio fiable no es el que no falla, sino el que, aunque fallen varias instancias de experimentos, es capaz de adaptarse automáticamente al nuevo número de instancias disponibles sin que el usuario se dé cuenta del fallo.

Un número excesivo de instancias defectuosas o un laboratorio infradimensionado tendrán el mismo resultado: un usuario potencialmente frustrado debido a los excesivos tiempos de espera en la cola y/o las altas posiciones en la cola. Unos tiempos de espera bajos o un número reducido de posiciones de cola máxima alcanzado hacen que el usuario perciba el laboratorio remoto como fiable, sabiendo que puede comprobar el rendimiento de su código de forma casi instantánea cada vez que accede al laboratorio, tal y como lo haría en un laboratorio clásico con su propio equipo.

En los casos en que el tiempo de espera o la posición máxima alcanzada en la cola son elevados, es necesario comparar el volumen de usuarios y sesiones de experimentos con el número de instancias de experimentos disponibles. De este modo, es posible determinar si se trata de una situación inusual (pico de carga en un momento determinado, por ejemplo, durante una clase, o cuando parte de las instancias de experimentación disponibles fallan, por ejemplo, debido a un corte de energía) o si se trata de una situación habitual que indicaría un subdimensionamiento del laboratorio.

Los parámetros descritos anteriormente dan lugar a las siguientes métricas:

- Número de instancias de experimentación disponibles.
- Número de días del subperiodo.
- Número de días del subperiodo en los que hubo al menos un uso.
- Número total de sesiones de experimentación realizadas en el subperiodo.
- Número máximo de sesiones de experimentación realizadas en un día.
- Número medio de sesiones de experimentación realizadas en un día.
- Percentil 90 del número de sesiones de experimentación realizadas en un día.

- Número máximo de usuarios atendidos en un día.
- Número medio de usuarios atendidos en un día.
- Percentil 90 del número de usuarios atendidos en un día.
- Posición máxima alcanzada en la cola en un día.
- Posición media alcanzada en la cola en un día.
- Percentil 90 de la posición alcanzada en la cola en un día.
- Percentil 99 de la posición alcanzada en la cola en un día.
- Tiempo máximo alcanzado en la cola de espera, en segundos, en un día.
- Tiempo medio alcanzado en la cola de espera, en segundos, en un día.
- Percentil 90 del tiempo alcanzado en la cola de espera, en segundos, en un día.

Los valores máximos ayudan a identificar las peores situaciones del laboratorio, mientras que la media y el percentil 90 de los datos ayudan a comprender el rendimiento del laboratorio en el 90 % de las ocasiones.

Los resultados de este análisis se presentan y discuten en la siguiente sección.

5.5 Discusión de los resultados cuantitativos del laboratorio remoto FPGA de Labsland

En esta sección se presentan los resultados del análisis realizado a través de la metodología antes mencionada.

Como se ha descrito, el análisis cuantitativo se centra en dos aspectos clave: la escalabilidad y la fiabilidad. Para ello, se analiza un conjunto de datos sobre el uso del laboratorio recogidos durante un periodo de 736 días en múltiples instituciones. Los datos de uso proceden de un uso de producción en el mundo real. Como tal, los profesores y estudiantes que utilizan el laboratorio no están directamente implicados en este estudio ni asociados a los autores, y utilizan el laboratorio libremente, sin seguir ninguna directriz específica. Esto

es lo que se pretende, ya que el objetivo es precisamente evaluar la idoneidad para el uso multinstitucional en el mundo real.

El periodo consta de cuatro subperiodos a lo largo de los cuales el número total de instancias de experimentación disponibles aumentó. En el subperiodo 1, había 10 instancias de experimentación en línea en el laboratorio, mientras que en los subperiodos siguientes había 18, 34 y 62 instancias de experimentación, respectivamente. Esta variación puede ser útil ya que permite evaluar cómo varía el rendimiento del modelo con diferentes números de instancias replicadas y diferentes cargas. No obstante, cabe destacar que no es un resultado del diseño experimental en sí, sino más bien una consecuencia del hecho de que se están utilizando datos de uso del mundo real, y a lo largo de este período relativamente largo (casi dos años) nuevas instituciones adquirieron nuevas copias del laboratorio, las desplegaron y se integraron en el clúster de la red de LabsLand.

La Figura 5.2 resume los datos obtenidos durante el análisis. Las filas 1, 2, 3 y 4 proporcionan el contexto, caracterizando el subperiodo. Las filas 5, 6 y 7 recogen los resultados relacionados con el número total de sesiones de experimentación realizadas en el laboratorio para cada día del subperiodo. Las filas 8, 9 y 10 recogen los resultados relativos al número total de usuarios para cada día del subperiodo. Las filas 11, 12, 13 y 14 muestran los resultados en términos de la posición máxima en la cola que permite el acceso al laboratorio para cada día del subperiodo. La posición 0 en la cola indica que no tienen a nadie por delante y que accederán al laboratorio en cuanto una de las instancias de experimentación existentes esté disponible. Esto ultimo, puede ser una acción inmediata. Las filas 15, 16 y 17 muestran los resultados en términos del tiempo máximo de espera en la cola alcanzado cada día (en segundos). Este tiempo de espera puede verse incrementado por diversas causas, como el tiempo necesario para restaurar el equipo del laboratorio entre sesiones, el tiempo de sesión de los usuarios anteriores o el tiempo de asignación de sesiones por parte del sistema de gestión del laboratorio.

La Figura 5.3 muestra gráficamente la posición máxima en la cola (en rojo) y el tiempo máximo de espera (en azul) para cada día dentro del periodo en el que se produjo al menos un uso del laboratorio. El eje izquierdo representa el tiempo en segundos, con un rango entre 0 y 240 segundos, mientras que el eje derecho representa la posición máxima alcanzada en la cola, con valores entre 0 y 4 (incluidos). La línea verde claro es una línea de tendencia lineal,

		Subperiod 1	Subperiod 2	Subperiod 3	Subperiod 4
1	**Number of available experimentation instances**	**10**	**18**	**34**	**62**
2	**Number of days of the subperiod**	361	115	205	55
3	**Number of days of the subperiod in which there was at least one use**	252	107	196	55
4	**Total number of experimentation sessions carried out in the subperiod**	22635	10033	24835	14874
5	**Maximum number of experimentation sessions carried out in one day**	807	1393	1242	1104
6	**Average number of experimentation sessions carried out in one day**	89.82	93.77	126.71	270.43
7	**90th percentile of number of experimentation sessions carried out in one day**	232.60	202.00	287.50	653.20
8	**Maximum number of users served in one day**	75	84	98	118
9	**Average number of users served in one day**	15.88	16.36	19.19	34.72
10	**90th percentile of number of users served in one day**	38.00	40.80	44.50	58.60
11	**Maximum position reached in the queue in one day**	4	0	2	1
12	**Average position reached in the queue in one day**	0.030	0.000	0.025	0.018
13	**90th percentile of position reached in the queue in one day**	0.00	0.00	0.00	0.00
14	**99th percentile of position reached in the queue in one day**	0.00	0.00	1.05	0.46
15	**Maximum waiting time reached in the queue, in seconds, in one day**	98	31	35	226
16	**Average waiting time reached in the queue, in seconds, in one day**	8.73	8.29	9.34	15.018
17	**90th percentile waiting time reached in the queue, in seconds, in one day**	12.00	13.00	16.50	19.60

Figura 5.2: Resultados del análisis cuantitativo del laboratorio LabsLand DE1-SoC FPGA.

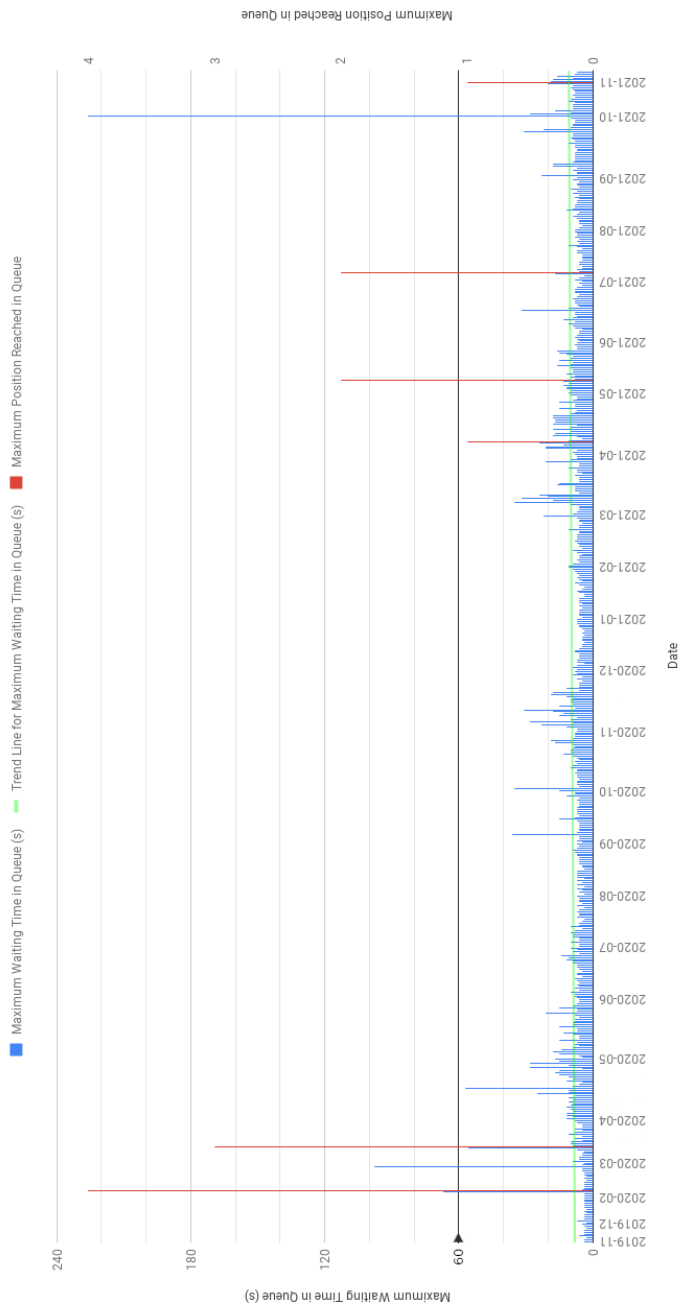


Figura 5.3: Posición máxima de las colas (en rojo) y tiempo máximo de espera registrado (en azul) para cada día del período de análisis.

calculada a partir de los datos del tiempo máximo de espera en la cola. Es una línea prácticamente horizontal, con una ligera tendencia ascendente. Cabe decir que la línea de color rojo es una línea continua, pero debido a que su valor representado es 0 en prácticamente la totalidad del periodo, queda no visible por situarse sobre el eje horizontal.

Un rápido análisis visual revela dos cosas: una, que hay algunos días en los que los valores se alejan de los valores típicos, y dos, que en la mayoría de los días, los valores se mantienen por debajo del umbral de 60 segundos que se ha fijado para el tiempo máximo de espera en cola y prácticamente se alcanzan 0 posiciones de espera en la cola. Sin embargo, la Figura 5.3 carece de la precisión necesaria para ver esos valores comunes en detalle debido a los valores máximos. Por ello, se ha incluido la Figura 5.4, que muestra una versión ampliada en la que se han recortado los casos de pico. La línea horizontal de la Figura 5.3, establecida en el eje horizontal en el punto de 60 segundos de tiempo máximo de espera, es un umbral arbitrario y también el nuevo límite que se utiliza para recortar los datos de la Figura 5.4.

Los casos máximos son aquellos en los que los valores se desvían significativamente de la media. Para identificar estos casos, se han elegido como valores de corte 45 segundos de tiempo de espera en la cola y una posición en la cola superior a 1. Los casos identificados se han recogido en una nueva figura, Figura 5.5. Estos casos no han sido frecuentes, pero no por ello dejan de ser analizados en detalle. Los datos adicionales asociados a estos días se han recogido en la Figura 5.6.

En general, los resultados del análisis cuantitativo son positivos e indican que se cumple el objetivo original. El laboratorio remoto es realmente escalable y fiable, y ha sido capaz de proporcionar el servicio a nivel de producción previsto para múltiples instituciones y miles de estudiantes y sesiones de experimentación, manteniendo una alta QoS la gran mayoría del tiempo a lo largo del periodo de 736 días.

Se realizaron más de 70.000 experimentos durante todo el periodo de pruebas. En el primer subperiodo, hubo días en los que el laboratorio no se utilizó, mientras que en los otros subperiodos, el uso fue significativamente más regular, ocurriendo casi todos los días e incluyendo los fines de semana.

Si se observa el número de sesiones realizadas en el laboratorio, se puede ver que el número máximo de sesiones por día se ha mantenido cerca de las 1000 sesiones diarias. Observando la media y el percentil 90 de esta métrica, se

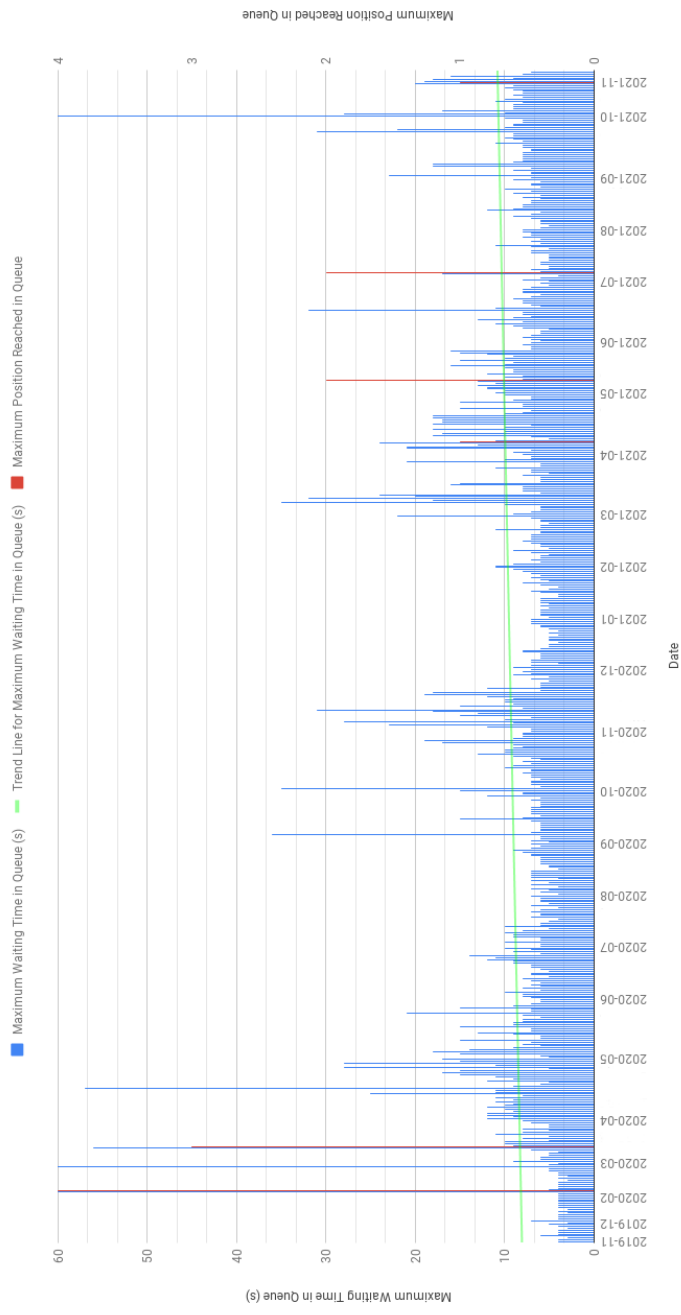


Figura 5.4: Versión ampliada del gráfico de la Figura 5.3.

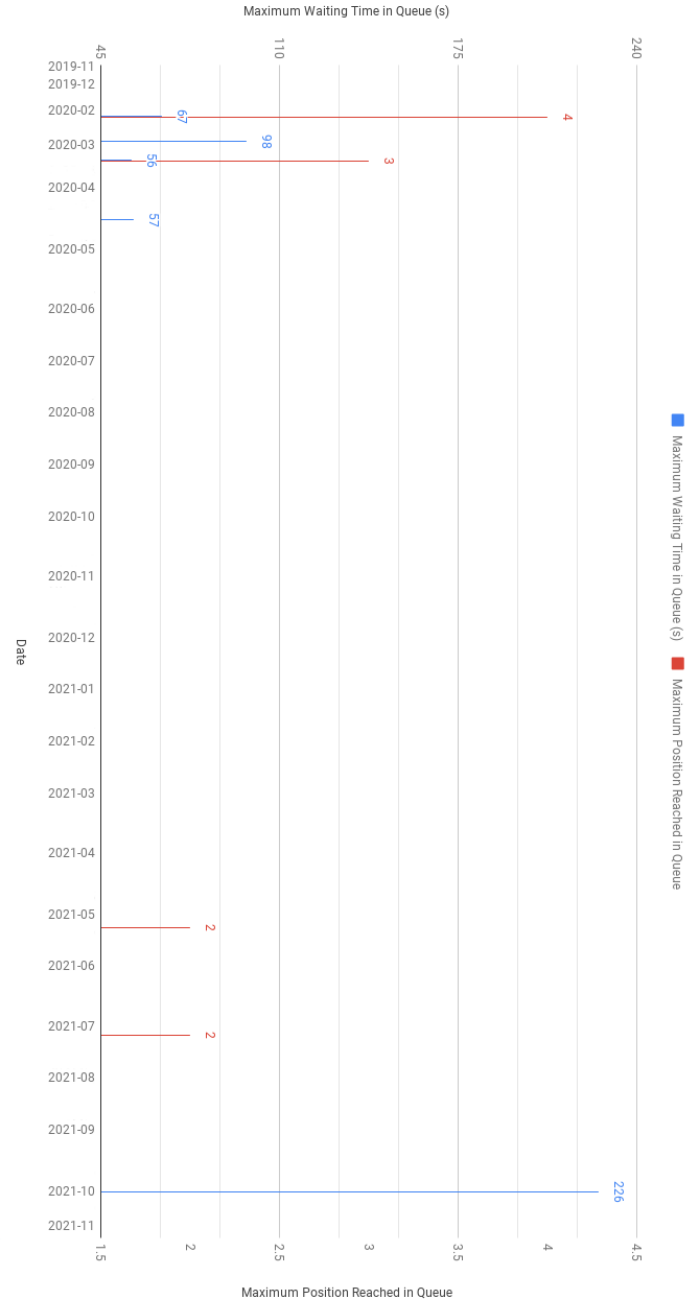


Figura 5.5: Versión ampliada del gráfico de la Figura 5.3.

Day	Date	Available Experimentation Instances	Experimentation Sessions Carried Out	Users Served	Maximum Position Reached in Queue	Maximum Waiting Time Reached in Queue
1	**2020-02-13**	10	168	55	4	67
2	**2020-03-18**	10	25	12	0	98
3	**2020-03-30**	10	53	30	3	56
4	**2020-04-30**	10	305	47	0	57
5	**2021-05-24**	34	363	58	2	13
6	**2021-07-19**	34	80	25	2	17
7	**2021-10-18**	62	643	105	0	226

Figura 5.6: Casos de pico detectados durante el análisis cuantitativo del laboratorio DE1-SoC FPGA de LabsLand.

constata que el uso del laboratorio se triplicó en el último subperiodo, mientras que se mantuvo casi constante a lo largo de los tres primeros subperiodos. En el primer y segundo subperiodo, el 90 % de los días, se atendieron hasta 232 sesiones de experimentación, mientras que en el cuarto subperiodo, durante el 90 % de los días, se atendieron hasta 653 sesiones diarias.

Si se observa el número de usuarios atendidos en el laboratorio, los resultados muestran una tendencia diferente. Tanto el número total de usuarios atendidos como el percentil 90 aumentaron sólo un 50 %. Si el número de sesiones servidas aumentó un 300 %, pero el número de usuarios servidos sólo aumentó un 50 %, es razonable suponer que los usuarios estaban utilizando el laboratorio de forma mucho más intensa.

En cuanto a la posición alcanzada en la cola, los resultados muestran que en el 90 % de los casos los usuarios no encontraron a nadie por delante de ellos en la cola.

Los valores máximos son especialmente bajos y, de hecho, son 0 durante el subperiodo 2. Como el percentil 90 es 0,00 en todos los subperiodos (el 90 % de los usuarios pudieron acceder sin esperar a nadie en la cola), se calculó el percentil 99. En los dos primeros subperiodos, ningún usuario se encontró en segundo lugar en la cola el 99 % de las veces, mientras que en los subperiodos 3 y 4, sólo el 1 % de los usuarios encontraron a alguien por delante de ellos en la cola.

Es importante recordar que, a pesar de no tener a nadie delante en la cola, los usuarios pueden tener que esperar poco tiempo. En este caso, no es posible discernir si el usuario ha accedido directamente al laboratorio o está esperando a que se libere la primera instancia. Estos tiempos de espera pueden verse incrementados por factores como los retrasos en la conexión o los procesos de recuperación entre sesiones de experimentación, por lo que no siempre se pueden controlar directamente. Por este motivo, los resultados no son completamente regulares en cuanto al tiempo de espera.

Hay algunos valores máximos bastante elevados, pero son pocos de los valores totales. El valor medio se acerca a los 9 segundos en los tres primeros subperiodos, en los que el laboratorio se utilizó con menos intensidad. El cuarto subperiodo tiene un valor medio de unos 15 segundos. Teniendo en cuenta el percentil 90, se puede asegurar que en el 90 % de los casos los usuarios tuvieron que esperar como máximo 19,60 segundos en el subperiodo en el que el uso del laboratorio fue más extenso, lo cual es un dato positivo que refuerza

la sensación de fiabilidad del laboratorio.

A partir de los datos gráficos mostrados en la Figura 5.3, es posible observar el comportamiento del laboratorio a lo largo del periodo de prueba. En general, tanto los tiempos de espera como las posiciones de cola alcanzadas se mantienen bajos, salvo en algunos casos de pico.

La Figura 5.6 muestra en detalle los datos relativos a los días en los que se detectaron picos. Se pueden observar estos picos en 7 días diferentes. Tres de ellos tienen tiempos de espera en cola relativamente altos (un usuario pasó 98, 57 y 226 segundos esperando, respectivamente, mostrados en azul) mientras que, sin embargo, tienen una posición máxima en cola de 0 (mostrada en rojo). Se trata de circunstancias poco frecuentes, ya que normalmente un tiempo de espera largo iría acompañado de una posición de cola superior a cero, y además el número de sesiones y usuarios no se aleja especialmente de la media. Es probable que esos raros casos (en 3 días de los 736) se debieran a problemas temporales en el clúster basado en la nube o a problemas temporales de la red.

En los días del 5º y 6º pico (en rojo), las posiciones máximas alcanzadas en la cola fueron superiores a 1 (2 para ambos días). Esto es más alto de lo habitual y está causado por las altas cargas de usuarios concurrentes. Implica que durante esos días, hubo momentos en los que todas las instancias en línea disponibles para el laboratorio estaban atendiendo a un estudiante en un momento determinado. En esos días, había 34 instancias del laboratorio, por lo que se trata de una carga importante. Sin embargo, los tiempos máximos de espera fueron de sólo 13 y 17 segundos, respectivamente. Por ello, la línea de la figura ampliada 5.5 es sólo roja, ya que esos tiempos están por debajo del umbral. Esto indica que, incluso en esos días concretos con una carga tan elevada, pudieron acceder con éxito al laboratorio tras esperar un tiempo razonable.

Por último, los días con el primer y tercer pico representan una situación en la que se alcanzó la capacidad del laboratorio y la calidad del servicio se vio parcialmente comprometida. La posición máxima en la cola era elevada, y los tiempos máximos de espera reales también eran relativamente altos. El día del primer pico, un usuario se encontraba en la cuarta posición de la cola y un usuario (probablemente, pero no necesariamente el mismo) tuvo que esperar hasta 67 segundos para acceder al laboratorio. El día del tercer pico, ocurrió algo similar: un usuario estaba en la tercera posición de la cola y un usuario tuvo que esperar hasta 56 segundos para acceder a su sesión de experimentación.

En el periodo de esos dos picos, el número de instancias de experimenta-

ción disponibles fue de 10. Estas 10 instancias dieron servicio a 55 usuarios el primer día y a 30 usuarios el segundo. Como no estaban repartidos a lo largo del día, sino que se concentraban en una ventana de tiempo más pequeña (probablemente porque estaban utilizando el laboratorio simultáneamente en clase), esta situación era más probable. Esta situación se relaja con un mayor número de instancias (como en los siguientes periodos) ya que es menos probable que clases de diferentes instituciones estén utilizando el laboratorio al mismo tiempo. También cabe destacar que esta situación sólo se produjo en 2 de los 736 días del periodo si estudio, no se repitió en los subperiodos en los que había un mayor número de instancias disponibles, y sí afectó a la QoS, pero probablemente de forma poco significativa (tuvieron que esperar como mucho 67 segundos para acceder al laboratorio).

5.6 Análisis cualitativo del laboratorio remoto de FPGA de LabsLand

Si bien la anterior validación cuantitativa es la más importante en relación con los objetivos técnicos de esta tesis, el análisis cualitativo de la satisfacción del alumno es también una parte importante, ya que el usuario final del laboratorio remoto es el propio alumno, y conocer su opinión es relevante.

5.6.1 Cuestionarios de evaluación de la experimentación remota

En la literatura, existen numerosos cuestionarios para conocer la satisfacción de los alumnos (Hu et al., 2016; Lang et al., 2007; Dormido et al., 2011; Corter et al., 2004). En este campo, la Universidad de Deusto, junto con el IQS (Institut Químic de Sarrià) ha desarrollado el cuestionario UXQ4RL (Cuadros et al., 2021). Este cuestionario de escala Likert 1-7 cuenta con nueve cuestiones que se recogen en la Tabla 5.1. Este cuestionario está validado estadísticamente, situación que no se da en los cuestionarios anteriormente nombrados.

5.6.2 Escenario de uso y resultados

El cuestionario UXQ4RL fue completado por los alumnos de la asignatura de Electrónica del curso 2021/2022 del Grado de Ingeniería en Tecnologías

Tabla 5.1: Cuestionario UXQ4RL. Adaptado de (Cuadros et al., 2021).

	Pregunta	**Categoría**
Q1	*El laboratorio remoto es fácil de usar*	Usabilidad
Q2	*Usar el laboratorio remoto se parece a estar en un laboratorio real*	Inmersión
Q3	*Mis interacciones con el laboratorio remoto parecen reales*	Inmersión
Q4	*El laboratorio remoto me ayuda a aprender*	Utilidad
Q5	*Cuando uso el laboratorio remoto, me concentro en las tareas encomendadas*	Inmersión
Q6	*Puedo predecir el resultado que se obtiene al usar cada elemento de la interfaz del laboratorio remoto*	Usabilidad
Q7	*El diseño del laboratorio remoto es suficientemente intuitivo de forma que no se requiere ayuda para usarlo*	Usabilidad
Q8	*El laboratorio remoto cumple con mis requisitos*	Utilidad
Q9	*El laboratorio remoto me ayuda a aprobar*	Utilidad

Industriales de la Universidad de Deusto. Dentro de esta asignatura los alumnos aprenden a implementar sistemas digitales básicos en VHDL FPGA. Los alumnos comprueban los diseños utilizando el laboratorio remoto FPGA de LabsLand anteriormente mencionado. Este laboratorio remoto es utilizado por los alumnos tanto en clase durante las lecciones, como fuera de la misma para completar sus tareas personales.

Los alumnos de esta asignatura fueron 40, y 35 de ellos completaron el cuestionario UXQ4RL en formato online. A continuación se analizan los resultados obtenidos.

La Figura 5.7 muestra gráficamente los resultados obtenidos en el cuestionario UXQ4RL, aunque en una escala Likert 1-5, donde 1 significa nada de acuerdo, y 5 significa muy de acuerdo.

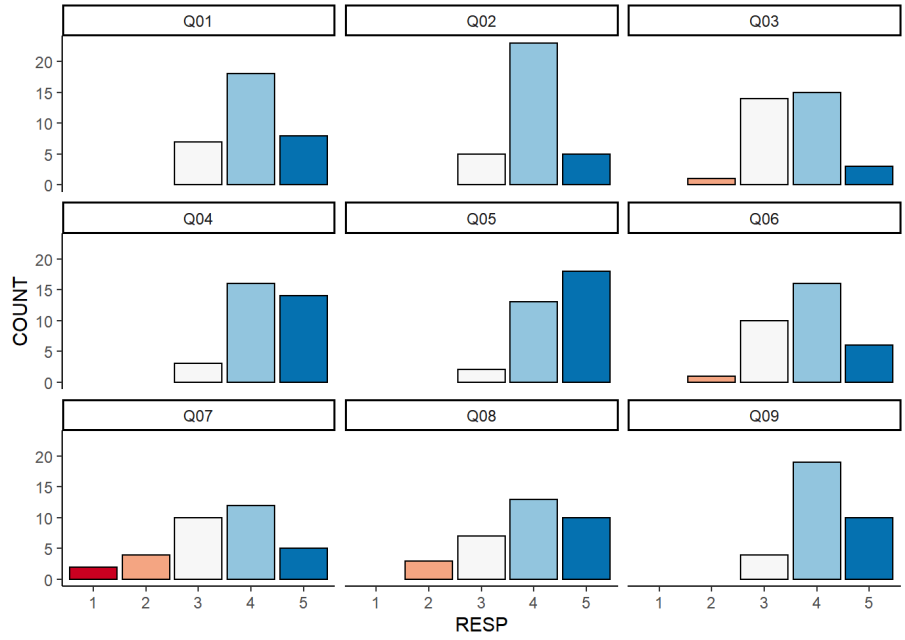


Figura 5.7: Resultados globales del cuestionario UXQ4RL.

Una simple inspección visual indica que los alumnos tienen una opinión positiva del laboratorio remoto de FPGA de LabsLand, donde solo para algunas cuestiones, aparecen valores por debajo de 3.

En la Figura 5.8 se muestran los mismos resultados respecto de su frecuencia relativa. Para todas las cuestiones planteadas al alumno, su opinión es mayoritariamente satisfactoria, con un porcentaje que supera el 70 % en todos los casos. Las dos cuestiones que suscitan un mayor apoyo por parte de los alumnos son las cuestiones Q4 y Q5, *El laboratorio remoto me ayuda a aprender* y *Cuando uso el laboratorio remoto, me concentro en las tareas encomendadas*. La cuestión Q7, *El diseño del laboratorio remoto es suficientemente intuitivo, de forma que no se requiere ayuda para usarlo*, es la que

cuenta con una valoración más negativa por parte del alumnado, aunque este porcentaje de alumnos no supera el 20 % del total (6 alumnos). En este caso hay que tener en cuenta que es posible que el alumno confunda la sencillez de la interfaz con la sencillez de las tareas que debe de completar.

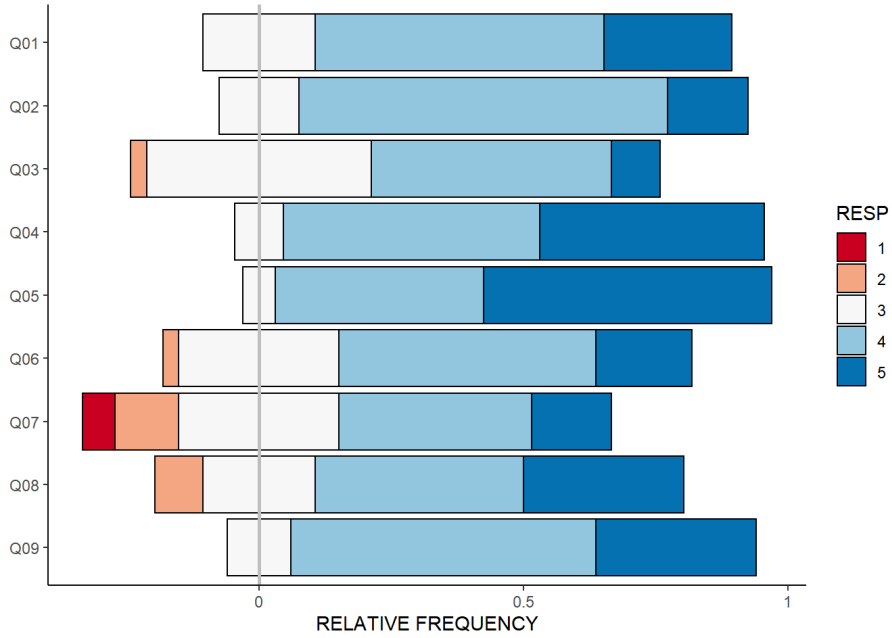


Figura 5.8: Resultados del cuestionario UXQ4RL en términos de frecuencia relativa.

Además, el cuestionario UXQ4RL permite medir la usabilidad (cuestiones Q1, Q6, Q7), la utilidad (cuestiones Q4, Q8 y Q9) y la inmersión (cuestiones Q2, Q3 y Q5), entendida esta ultima como la sensación del alumno de estar operando con un equipamiento real.

Al igual que antes, la Figura 5.9 muestra el número de veces que cada una de estas tres categorías alcanza un valor entre 3 y 15, ya que cada una de las tres preguntas de cada categoría puede tener un valor de 1 a 5, y por tanto el valor global, estará entre 3 y 15.

De nuevo, una simple inspección visual indica que los alumnos valoran el

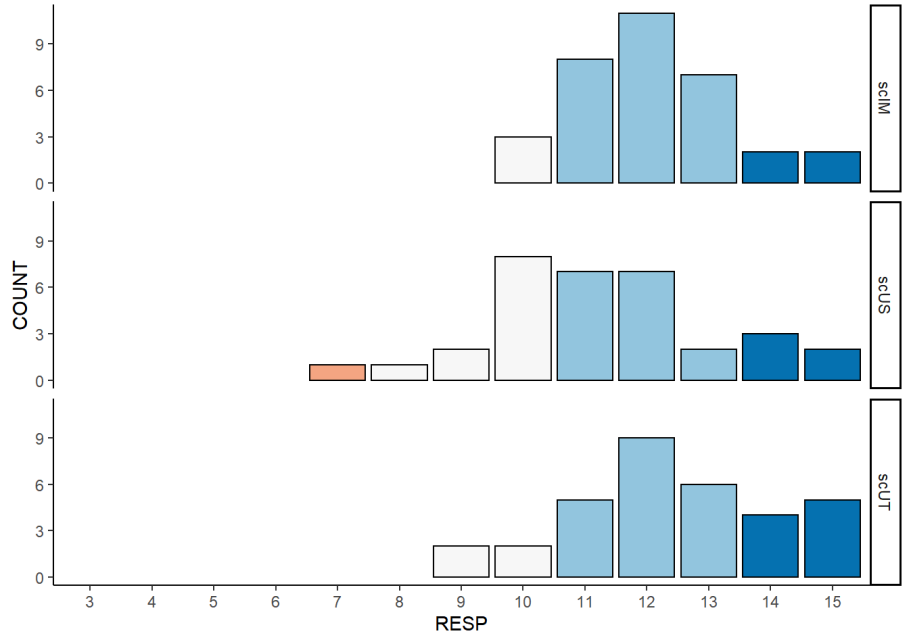


Figura 5.9: Resultados del cuestionario UXQ4RL por categorías.

laboratorio remoto FPGA de LabsLand como un entorno usable y útil, y que además favorece la inmersión.

La Figura 5.10 muestra los anteriores resultados centrados respecto del valor medio. Esta figura muestra que el 100 % de los alumnos considera que este laboratorio remoto es inmersivo, y prácticamente el 100 % lo considera útil. En el caso de la usabilidad, solo 3 alumnos (menos de un 10 %) tienen una opinión no positiva.

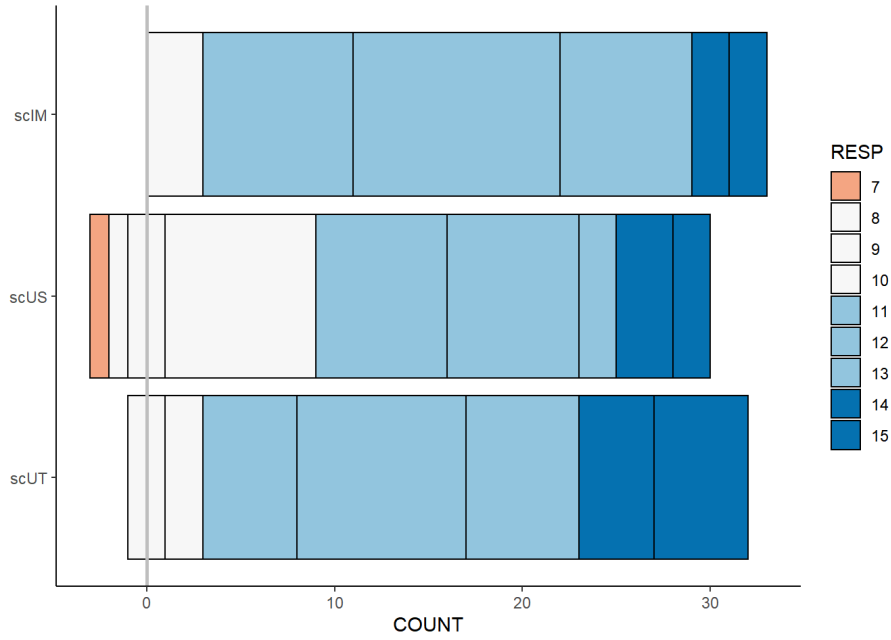


Figura 5.10: Resultados del cuestionario UXQ4RL por categorías, en terminos de frecuencia relativa.

En conclusión, el laboratorio remoto FPGA de LabsLand no solo es escalable y fiable, sino que además, los alumnos tienen una opinión positiva del mismo. Es decir, este laboratorio remoto es tanto técnica como didácticamente útil en el aula.

5.7 Conclusión

El objetivo principal de esta sección es evaluar WebLabPRO para conocer si realmente es posible desplegar laboratorios remotos que puedan ser eficaces en entornos educativos en el mundo real, abarcando múltiples instituciones. Ello podría conducir a su adopción generalizada.

Como se ha comentado, para que un laboratorio remoto se convierta en una herramienta generalizada y estándar en el entorno de una institución educativa, sus usuarios potenciales deben confiar en él y percibirlo como una herramienta fiable. En un laboratorio clásico esto también es importante: el equipo debe funcionar. Aunque, como los usuarios mantienen una cercanía física con el equipo, a menudo pueden solucionar problemas que serían mucho más difíciles de solucionar a distancia. Además, aunque se produzcan problemas, es menos probable que los usuarios sientan que no tienen el control del experimento, lo cual es una percepción muy perjudicial para la calidad del servicio. Por lo tanto, para que se confíe en ellos y se les perciba como fiables, el nivel de exigencia es aún mayor para un laboratorio remoto.

El estudio realizado, que abarca un periodo de algo más de 2 años (736 días), sugiere que el objetivo principal se ha cumplido y que, efectivamente, es posible desplegar laboratorios a nivel de producción para un uso multinstitucional en el mundo real aplicando la arquitectura WebLabPRO desarrollada en esta tesis. Los profesores y sus clases, de múltiples instituciones diferentes, utilizaron con éxito el laboratorio remoto FPGA para sus cursos, dando lugar a miles de sesiones de experimentación, con 72.377 accesos por parte de los alumnos a lo largo de los cuatro períodos descritos. En todo momento, a lo largo de ese periodo de 736 días, el laboratorio permaneció disponible y proporcionó un acceso rápido y eficaz y una alta calidad de servicio. Sólo en 2 días de los 736 un usuario experimentó pequeñas degradaciones en la QoS, al tener que esperar hasta 67 segundos para acceder al laboratorio. Por tanto, se concluye que en general, los objetivos de disponibilidad y QoS se cumplen claramente.

Aunque no forma parte de este estudio, cabe destacar que los informes individuales de los profesores que utilizaron el laboratorio a lo largo de este periodo y que comunicaron su experiencia fueron positivos. En un estudio independiente que uno de los profesores realizó utilizando el laboratorio FPGA LabsLand, por ejemplo, los resultados de aprendizaje fueron de hecho superiores a los del curso práctico tradicional de su año anterior (Hussein and Wilson,

2021).

Desde un punto de vista cualitativo, el laboratorio remoto FPGA de LabsLand, ha cumplido con las expectativas de los usuarios en términos de usabilidad, utilidad e inmersión, siendo estas dos ultimas las mas valoradas por los alumnos.

En resumen, puede concluirse que aplicando la arquitectura WebLabPRO propuesta es posible crear laboratorios remotos que sean realmente útiles para el uso multinstitucional en el mundo real y que puedan cumplir con los estándares de calidad de servicio. Cumplir con estos estándares y ser confiables es un paso importante para que los laboratorios remotos se conviertan en una herramienta verdaderamente ubicua para la educación científica y técnica en las universidades y otras instituciones de aprendizaje y, por lo tanto, alcancen su potencial y se generalicen realmente.

CAPÍTULO

6

Conclusiones finales y líneas futuras

A lo largo de los capítulos anteriores se han reflejado los esfuerzos llevados a cabo para proporcionar dos características fundamentales a la experimentación remota: escalabilidad y fiabilidad. Estas características conforman el núcleo de la arquitectura WebLabPRO diseñada e implementada. WebLabPRO se ha utilizado para el despliegue de múltiples laboratorios remotos de sistemas programables que han sido validados en entornos académicos reales. Todos estos esfuerzos y resultados serán resumidos y explicitados en el presente capítulo que finalizará con la propuesta de un conjunto de líneas futuras de investigación en el ámbito de la experimentación remota.

6.1 Conclusiones y resultados

Una vez introducido el presente trabajo doctoral, y establecidos la hipótesis y los objetivos del mismo, en el Capítulo 2 *La experimentación remota y los laboratorios remotos* se establece la definición de laboratorio remoto analizando

los principales estudios que demuestran su validez como herramienta formativa y su relevancia en el actual contexto educativo. Además, en este capítulo se caracterizan los diferentes experimentos remotos, se identifican los principales laboratorios remotos desplegados internacionalmente y se establecen las causas que históricamente han frenado la adopción de esta tecnología por parte de la comunidad docente. Paralelamente, se describen en detalle las características principales de los laboratorios remotos y se desarrollan las dos características principales que dan lugar a esta tesis: escalabilidad y fiabilidad, estableciendo la relevancia de ambas en la evolución de la experimentación remota, fundamentalmente en campo de los sistemas electrónicos programables.

El Capítulo 3 *Diseño, implementación y validación de una arquitectura de alta escalabilidad para experimentación remota* se centra en la consecución del objetivo específico OE2, *Diseñar, implementar y validar una arquitectura para el desarrollo de laboratorios remotos que facilite la escalabilidad de los mismos*. Para ello se presenta una arquitectura enfocada en facilitar la integración de múltiples instancias de experimentación en laboratorios remotos sobre sistemas electrónicos programables que recibe el nombre de WebLabPRO. Esta arquitectura establece una taxonomía organizada sobre cuatro capas completamente desacopladas que abordan las diferentes tareas requeridas para desarrollar la experimentación de forma progresiva desde la propia plataforma de experimentación hasta el usuario. De esta forma, la Capa de Hardware, que ocupa el nivel inferior de la arquitectura, integra las diferentes instancias de experimentación junto con los componentes hardware que posibilitan la interactuación física con las mismas. Posteriormente, la Capa de Servidor de Interfaz proporciona una interfaz de programación de aplicaciones que permite controlar el experimento proporcionando un nivel total de abstracción hardware. En el siguiente nivel, la Capa de Servidor de Laboratorio se encarga de proporcionar la interacción necesaria entre el usuario y el experimento y finalmente, la Capa de Sistema de Gestión de Laboratorios Remotos o RLMS proporciona los recursos de administración comunes a los distintos laboratorios remotos. Una vez definida la arquitectura de escalabilidad, el Capítulo 3 incluye la evaluación de la misma mediante el desarrollo, despliegue y validación de un laboratorio remoto diseñado bajo las especificaciones establecidas en WebLabPRO. El laboratorio remoto resultante, AduinoRL, es analizado de forma pormenorizada y comparado con dos laboratorios remotos desplegados internacionalmente por instituciones con experiencia en la experimentación re-

mota en términos de escalabilidad soportada, adaptabilidad a nuevas plataformas de experimentación y eficiencia de coste en el despliegue. Hay que reseñar que el laboratorio remoto Arduino de LabsLand ha sido validado en un escenario educativo real demostrando la validez de la arquitectura WebLabPRO en términos de escalabilidad .

Cumplidos los objetivos identificados para la escalabilidad, el Capítulo 4 *Diseño, implementación y validación de una arquitectura tolerante a fallos para experimentación remota* se centra en añadir a la arquitectura WebLabPRO un conjunto de recursos enfocados a optimizar la fiabilidad de los laboratorios remotos. Este capítulo aúna los esfuerzos desarrollados en el trabajo doctoral para la consecución del OE3 *Diseñar, implementar y validar una arquitectura para el desarrollo de laboratorios remotos que facilite la fiabilidad de los mismos*. Una vez identificada la falta de fiabilidad en los laboratorios remotos como uno de los principales obstáculos en la adopción de estos sistemas entre los profesores, el capítulo comienza con un detallado análisis de fiabilidad en los que cinco laboratorios remotos, ampliamente reconocidos por la comunidad internacional, son accedidos durante un periodo de tiempo para establecer, de forma cuantitativa y cualitativa, la fiabilidad de los mismos. Fruto de este análisis se establecen los principales errores que provocan la falta de accesibilidad de los laboratorios remotos y los diferentes grados en los que afectan a la experimentación. Esta clasificación sirve como base para el diseño de una arquitectura destinada a optimizar la fiabilidad de los laboratorios remotos basada en el desarrollo de pruebas organizadas en cuatro capas diferentes que abordan progresivamente desde problemas de fiabilidad genéricos a la experimentación remota, hasta errores específicos resultantes de la propia naturaleza de una experimentación concreta. De esta forma la primera Capa de Excepciones consiste en detectar los posibles errores experimentados por los usuarios durante una sesión de experimentación. La segunda Capa de Controles Genéricos realiza una serie de comprobaciones genéricas de forma automática que validan fundamentalmente la comunicación con las interfaces de programación de aplicaciones. Además una tercera Capa de Controles Específicos realiza una serie de comprobaciones que han sido diseñadas específicamente para cada laboratorio atendiendo a los subsistemas específicos de cada uno y especialmente a aquellos con mayor probabilidad de fallo o rotura. Finalmente la cuarta Capa de Pruebas de Alta Especificidad, incorpora de forma automatizada sesiones reales de experimentación y valida los resultados obte-

nidos pudiendo garantizar el correcto funcionamiento del sistema. Tal y como se hizo en el capítulo anterior la validación de la arquitectura WebLabPRO de fiabilidad se ha llevado a cabo en un escenario real implementando las cuatro capas de la arquitectura sobre dos laboratorios remotos desplegados por la empresa LabsLand que se encuentran en fase producción. Estos dos laboratorios, el laboratorio remoto de robótica de LabsLand y el laboratorio remoto Arduino de LabsLand que son utilizados por distintas instituciones educativas a lo largo del mundo, han sido monitorizados a lo largo de un periodo de tiempo superior a 5 meses permitiendo validar la fiabilidad ofrecida por la arquitectura WebLabPRO. Esta validación ha concluido que anque la arquitectura WebLabPRO no evita ni reduce el número de errores en los laboratorios remotos, permite anticipar los mismos y mejora la experiencia de los usuarios evitando experiencias frustrantes.

Finalmente, el Capítulo 5 *Evaluación de la arquitectura WebLabPRO* permite validar conjuntamente las soluciones de escalabilidad y fiabilidad planteadas de forma individual en los capítulos anteriores, para alcanzar el objetivo OE4 *Integrar las soluciones de escalabilidad y fiabilidad en una única arquitectura para su despliegue en un entorno industrial y comercial de experimentación remota*. Siguiendo la metodología establecida, para la validación integrada de la arquitectura WebLabPRO se ha diseñado, desarrollado e implementado un nuevo laboratorio remoto siguiendo las normas de diseño indicadas en los Capítulos 3 y 4, para demostrar que las soluciones de escalabilidad y fiabilidad desarrolladas y validadas individualmente mantienen sus resultados cuando se integran en un mismo laboratorio. Para ello, contando con la colaboración de la empresa LabsLand se ha desarrollado el laboratorio remoto de FPGA de LabsLand siguiendo la arquitectura planteada en el presente trabajo doctoral. Este laboratorio, descrito en detalle en el Capítulo 5, ha permitido desarrollar un escenario real de validación en el que han participado numerosas instituciones de múltiples países sin tener conocimiento de estar participando en la validación de una tesis doctoral. A lo largo de los 25 meses de duración del análisis el laboratorio ha ido integrando paulatinamente nuevas instancias de experimentación hasta superar las 60 unidades y soportando una media de más de 250 sesiones diarias. El análisis detallado del uso del experimento remoto muestra que únicamente en tres situaciones el tiempo de espera ha sido superior a 60 segundos, tiempo máximo propuesto como indicador de QoS. En cuanto a la posición máxima alcanzada en la cola, esta en ningún caso ha si-

do superior a cuatro. Queda por tanto validada la escalabilidad ofrecida por la arquitectura WebLabPRO. Además, la fiabilidad queda demostrada dado que durante todo el periodo de prueba el sistema se mantuvo activo sin disminuir significativamente la calidad del servicio. Cualitativamente, los alumnos expresan sentirse satisfechos con el uso del laboratorio remoto en terminos de usabilidad, utilidad e inmersión.

6.2 Hipótesis y validación de objetivos

Al comienzo de esta tesis se planteó la hipótesis fundamental de la misma:

> *Es posible crear una arquitectura para el desarrollo de laboratorios remotos que facilite el escalado y dé soporte a un elevado número de usuarios concurrentes, con coste contenido, y que permita garantizar la fiabilidad del proceso de experimentación remota a través de técnicas de detección automática de fallos.*

La comprobación de la anterior hipótesis dará lugar a la siguiente meta:

> *Diseñar e implementar una arquitectura (WebLabPRO) capaz de permitir el despliegue efectivo de laboratorios remotos escalables y fiables en un entorno real de uso con fines comerciales e industriales bajo un enfoque profesional.*

La comprobación de esta hipótesis se sustanció en la consecución de varios objetivos específicos. A continuación se contrasta cada uno de estos objetivos iniciales con los resultados obtenidos:

- *OE1: Analizar las arquitecturas de hardware de los laboratorios remotos en el estado del arte en cuanto a su escalabilidad, potencial de concurrencia, eficiencia de costes y fiabilidad.*

  En los Capítulos 3, 4 y 5 distintas soluciones técnicas a los problemas de escalabilidad y fiabilidad han sido analizadas cuantitativamente atendiendo a criterio de potencial de concurrencia, calidad de servicio (QoS) y coste, siendo necesario para ello establecer criterios cuantitativos claros. Estos criterios se convierten en los indicadores necesarios para analizar la arquitectura WebLabPRO propuesta y comparar estos resultados con las soluciones propuestas por el estado del arte.

- *OE2: Diseñar, implementar y validar una arquitectura para el desarrollo de laboratorios remotos que facilite la escalabilidad de los mismos*.

  En el Capítulo 3 la arquitectura WebLabPRO aporta soluciones técnicas para que los laboratorios remotos desplegados sobre ella sean altamente escalables gracias a la integración de distintas tecnologías como Redis, API REST y contenedores o Dockers para el diseño de las distintas capas aisladas que garantizan la escalabilidad. La solución implementada permite su despliegue mediante dispositivos de bajo coste, facilitando su utilización en entornos reales y comerciales (LabsLand) y no solo como una prueba de concepto. La escalabilidad implementada permite el despliegue de cientos de instancias de un experimento remoto, lo que lleva al estado del arte a un estadio totalmente nuevo y da este objetivo por cumplido.

- OE3: *Diseñar, implementar y validar una arquitectura para el desarrollo de laboratorios remotos que facilite la fiabilidad de los mismos.*

  En el Capítulo 4 la arquitectura WebLabPRO aporta soluciones técnicas que aumentan la fiabilidad del experimento remoto, que es considerado el principal requisito para un despliegue exitoso en un entorno real y comercial. El uso de técnicas procedentes del área del QoS y de la escalabilidad obtenida en OE2 permite que los laboratorios remotos desplegados usando WebLabPRO sean capaces de determinar qué instancias no son operativas y cómo restaurarlas de forma automática en caso posible. Operativamente la solución implementada se basa en distintas capas, aisladas entre sí, responsable cada una de ellas de un conjunto de fallos específicos que dan lugar a distintas soluciones técnicas de *healthcheck*, utilizando técnicas de visión artificial y servicios de testeo automático. Al desplegar varios laboratorios remotos en un entorno real y comercial (LabsLand) y observar que la QoS obtenida es aceptable, el objetivo se puede dar cumplido.

- *OE4: Integrar las soluciones de escalabilidad y fiabilidad en una única arquitectura para su despliegue en un entorno industrial y comercial de experimentación remota.*

  El Capítulo 5 muestra la integración exitosa en la arquitectura WebLabPRO de técnicas novedosas de escalabilidad y fiabilidad me-

diante una validación extensiva en el tiempo, más de dos años, y en el conjunto de usuarios, millares, en el entorno real y comercial de la empresa LabsLand. Los indicadores obtenidos avalan la consecución de la meta original y permiten la explotación de WebLabPRO.

6.3 Publicaciones

El desarrollo de esta tesis ha dado lugar a diversas publicaciones en revistas y presentaciones en conferencias con el afán de divulgar resultados y contrastarlos con la comunidad científica.

La Tabla 6.1 resume solo las principales contribuciones de esta tesis, relacionándolas con los objetivos previamente planteados, así como los capítulos que incluyen resultados de dichas publicaciones. A continuación se aporta información específica de cada publicación, así como su cita completa y el factor de impacto para las publicaciones en revistas científicas.

Tabla 6.1: Resultados de divulgación

Publicación	Tipo	Estado	Capítulo	Objetivo
P1	Revista (Q1)	Publicado	3	OE1, OE2
P2	Revista (Q2)	Publicado	4	OE1, OE3
P3	Revista (Q2)	Publicado	5	OE4
P4	Conferencia	Publicado	3	OE1, OE2
P5	Conferencia	Publicado	3,5	OE1, OE2
P6	Conferencia	Publicado	5	OE4
P7	Conferencia	Publicado	5	OE4

- **Publicación P1** - *Improving the Scalability and Replicability of Embedded Systems Remote Laboratories through a Cost-effective Architecture*. Publicada en la revista *IEEE Access*. Esta publicación analiza el estado del arte de los laboratorios remotos en términos de escalabilidad, e identifica la falta de escalabilidad como uno de los principales problemas para la popularización de los laboratorios remotos. Como solución a dicho problema, propone una arquitectura mixta hardware-software para

la creación de laboratorios remotos de alta escalabilidad con un coste contenido. Esta arquitectura es validada técnicamente mediante la creación y posterior análisis de un laboratorio remoto con múltiples instancias de experimentación, capaz de soportar múltiples usuarios de manera concurrente. Los resultados obtenidos sugieren que la arquitectura propuesta cumple con los objetivos propuestos al inicio de la publicación.

> A. Villar-Martínez, L. Rodríguez-Gil, I. Angulo, P. Orduña, J. García-Zubía y D. López-De-Ipiña, *Improving the Scalability and Replicability of Embedded Systems Remote Laboratories Through a Cost-Effective Architecture*, en IEEE Access, vol. 7, pág. 164164-164185, 2019, doi: 10.1109/ACCESS.2019.2952321.
>
> "IEEE Access" JCR IF (2019): 3.745.
> Q1 in Computer Science, Information Systems.

- **Publicación P2** - *Towards Reliable Remote Laboratory Experiences: A Model for Maximizing Availability Through Fault-Detection and Replication*. Publicada en la revista *IEEE Access*. Esta publicación analiza el estado del arte de los laboratorios remotos en términos de fiabilidad, e identifica la falta de fiabilidad como uno de los principales problemas para la popularización de los laboratorios remotos. Como solución a dicho problema propone un modelo de detección de fallos. Este modelo es validado técnicamente mediante la integración del mismo en dos laboratorios remotos multiinstancia ya existentes. Los resultados obtenidos sugieren que el modelo desarrollado cumple con los objetivos propuestos al inicio de la publicación.

> A. Villar-Martínez, J. García-Zubía, I. Angulo y L. Rodríguez-Gil, *Towards Reliable Remote Laboratory Experiences: A Model for Maximizing Availability Through Fault-Detection and Replication*, en IEEE Access, vol. 9, pág. 45032-45054, 2021, doi: 10.1109/ACCESS.2021.3065742.
>
> "IEEE Access" JCR IF (2021): 3.476.
> Q2 in Computer Science, Information Systems.

- **Publicación P3** - *Towards Widespread Remote Laboratories: Evaluating the Effectiveness of a Replication-based Architecture for Real-world Multi-institutional Usage*. Publicado en la revista *IEEE Access*. Este trabajo evalúa si, aplicando una arquitectura orientada a la replicación de instancias rentable y un modelo orientado a la detección de fallos, es posible crear laboratorios que puedan proporcionar una alta QoS y que, por tanto, puedan ser utilizados en un entorno real en múltiples instituciones. Los resultados sugieren que la calidad del servicio es adecuada y que este enfoque puede conducir a laboratorios remotos confiables, apropiados para el uso educativo en el mundo real, y su eventual adopción generalizada.

  A. Villar-Martínez, J. García-Zubía, I. Angulo y L. Rodríguez-Gil, *Towards Widespread Remote Laboratories: Evaluating the Effectiveness of a Replication-based Architecture for Real-world Multi-institutional Usage*, en IEEE Access, vol. 10, pág. 86298-86317, 2022, doi: 10.1109/ACCESS.2022.3198961.

  "IEEE Access" JCR IF (2021): 3.476.
  Q2 in Computer Science, Information Systems.

- **Publicación P4** - *WebLabLib: New Approach for Creating Remote Laboratories*. Presentado en la conferencia *International Conference on Remote Engineering and Virtual Instrumentation, REV2019*. Esta publicación presenta el framework *WebLabLib*, una herramienta software para la integración y federación de laboratorios remotos. Esta comunicación recibió en 2019 el premio *Online Laboratory Award* del *Global Online Laboratory Consortium (GOLC)* al mejor laboratorio remoto visualizado.

  P. Orduña, L. Rodríguez-Gil, I. Angulo, U. Hernández-Jayo, A. Villar-Martínez y J. García-Zubía, *WebLabLib: New Approach for Creating Remote Laboratories*, en International Conference on Remote Engineering and Virtual Instrumentation REV2019, pág. 477-488, 2019, Springer, Cham, doi: 10.1007/978-3-030-23162-0_43.

- **Publicación P5** - *WEBLAB-ARM: A Scalable, Replicable, Reliable Remote Laboratory For Distance Learning Of ARM-microcontroller-based Embedded Systems*. Presentado en la conferencia *14th annual International Conference of Education, Research and Innovation ICERI2021*. Esta publicación presenta la creación de un laboratorio remoto para la experimentación con tarjetas de desarrollo de microcontroladores ARM, utilizando la arquitectura de creación de laboratorios remotos WebLabPRO.

  A. Villar-Martínez, I. Angulo, L. Rodríguez-Gil y J. García-Zubía, *WEBLAB-ARM: A Scalable, Replicable, Reliable Remote Laboratory For Distance Learning Of Arm-microcontroller-based Embedded Systems*, en 14th annual International Conference of Education, Research and Innovation ICERI2021, pág. 5799-5808, 2021, IATED, doi: 10.21125/iceri.2021.1308.

- **Publicación P6** - *Mobile Arduino Robot Programming Using a Remote Laboratory in UNAD: Pedagogic and Technical Aspects*. Presentado en la conferencia *17th International Conference on Remote Engineering and Virtual Instrumentation, REV2020*. Esta publicación analiza los aspectos pedagógicos y técnicos de un laboratorio remoto para experimentación con un robot Arduino. El análisis se realiza con estudiantes de la Universidad Nacional Abierta y a Distancia (UNAD) en Costa Rica. Dicho laboratorio remoto está desarrollado sobre la arquitectura WebLabPRO.

  P. Andrea Buitrago, R. Camacho, H. Enseider Pérez, O. Jaramillo, A. Villar-Martínez, L. Rodríguez-Gil y P. Orduña, *Mobile Arduino Robot Programming Using a Remote Laboratory in UNAD: Pedagogic and Technical Aspects*, en Cross Reality and Data Science in Engineering: Proceedings of the 17th International Conference on Remote Engineering and Virtual Instrumentation REV2020, pág. 171-183, 2021, Springer, Cham, doi: 10.1007/978-3-030-52575-0_14.

- **Publicación P7** - *FPGA Remote Laboratory: Experience in UPNA and UNIFESP*. Presentado en la conferencia *17th International Conferen-*

ce on Remote Engineering and Virtual Instrumentation, REV2020. En esta publicación se desarrolla un laboratorio remoto para experimentación con FPGAs. Este laboratorio es desplegado utilizando múltiples instancias de experimentación (17) a nivel global, y utiliza la arquitectura WebLabPRO para como base.

> C. Aramburu Mayoz, A.L. da Silva Beraldo, A. Villar-Martínez, L. Rodríguez-Gil, W. F. Moireira de Souza Seron, T. de Oliveira y P. Orduña, *FPGA Remote Laboratory: Experience in UPNA and UNIFESP*, en Cross Reality and Data Science in Engineering: Proceedings of the 17th International Conference on Remote Engineering and Virtual Instrumentation REV2020, pág. 112-127, 2021, Springer, Cham, doi: 10.1007/978-3-030-52575-0_9.

El autor de esta trabajo ha participado, durante el periodo de investigación en el que se enmarca esta tesis, en otros trabajos científicos, que si bien no están relacionados de manera directa con ésta tesis, si pertenecen al ámbito de los laboratorios remotos. La Tabla 6.2 recoge éstos trabajos.

Tabla 6.2: Otros resultados de divulgación del autor en el ámbito de los laboratorios remotos

Publicación	**Tipo**
(Orduña et al., 2018b)	Conferencia
(Buitrago et al., 2018)	Conferencia
(Rodríguez-Gil et al., 2019)	Revista (Q1)
(García-Zubía et al., 2019)	Conferencia
(Angulo et al., 2019)	Conferencia

6.4 Líneas futuras de investigación

Necesariamente la consecución de los objetivos anteriores y de la meta final ha permitido fijar nuevas líneas de investigación que han podido ser abordadas en esta tesis. A continuación se reflejan aquellas que se muestran como más incisivas.

En la consecución del requisito de escalabilidad ha quedado fuera otra característica muy importante: la adaptabilidad. La arquitectura WebLabPRO, gracias al aislamiento de sus capas, favorece la adaptabilidad y permite que nuevos sistemas programables se integren con facilidad en los laboratorios remotos de LabsLand, como ya se ha comprobado con el despliegue de laboratorios basados en dispositivos concretos. Sin embargo este proceso exige que el diseñador conozca con suficiente detalle la arquitectura WebLabPRO y que tenga habilidades hardware y software. Una línea futura que se abre es el diseño de una arquitectura que aporte modularidad a cualquier laboratorio remoto bajo el paradigma *plug&play* y con la mínima intervención de un diseñador. De esta forma los centros educativos no solo serían usuarios finales de laboratorios remotos, sino que también podrían ser proveedores de los mismos.

Desde un punto de vista más técnico, la escalabilidad puede ser mejorada mediante el uso intensivo de contenedores software para la implementación de las diferentes capas de la arquitectura WebLabPRO utilizando Docker y herramientas similares. Esta tarea tendría como meta global la estandarización del diseño de laboratorios remotos superando el estándar IEEE 1876-2019, actualmente propuesto y con muy bajo índice de adopción.

En cuanto a la fiabilidad, un campo de trabajo abierto es utilizar las técnicas predictivas industriales en el ambiente educativo de WebLabPRO para, en la medida de lo posible, predecir si un experimento remoto va a tener problemas de funcionamiento en un tiempo concreto. La implementación de estas técnicas preventivas bien puede basarse en la utilización de herramientas de inteligencia artificial, máxime cuando la experiencia acumulada en esta tesis ha dado lugar a un *tracking* considerable de situaciones y fallos.

Además de las técnicas anteriores un campo con mucho potencial es el uso de técnicas de visión artificial para reconocer situaciones de fallos en los experimentos. Así, la integración de estas técnicas de visión artificial con los modelos propios de IA ofrecen un campo de investigación cuya meta sea la automatización de pruebas de fiabilidad.

Por último en cada uno de los dos requisitos exigidos a WebLabPRO para alcanzar la meta final, esta tesis ha tenido que definir sus indicadores. Un campo de investigación en experimentación remota alejado de la técnica es aquel que tenga por objeto la definición de indicadores QoS para experimentación remota desde una perspectiva general.

Bibliografía

Al-Busaidi, K. A. and Al-Shihi, H. (2010). Instructors acceptance of learning management systems: A theoretical framework. *Communications of the IBIMA*, 2010(2010):1–10. 71

Albiol, A., Corbi, A., and Burgos, D. (2017). Design of a Remote Signal Processing Student Lab. *IEEE Access*, 5:16068–16076. 42

Alexander, P. and Radhakrishnan, N. (2015). Remote lab implementation on an embedded web server. In *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, pages 1–5. IEEE. 42, 127

Alexeev, V., Domashnev, P., Lavrukhina, T., and Nazarkin, O. (2019). The design principles of intelligent load balancing for scalable websocket services used with grid computing. *Procedia Computer Science*, 150:61–68. 62

Angulo, I. (2015). *Arquitectura abierta para el despliegue de laboratorios remotos sobre tecnologías de desarrollo de sistemas embebidos*. PhD thesis, University of Deusto. 122

Angulo, I., García-Zubía, J., Hernández-Jayo, U., Uriarte, I., Rodríguez-Gil, L., Orduña, P., and Pieper, G. M. (2017). Roboblock: A remote lab for robotics and visual programming. In *2017 4th Experiment@ International Conference (exp. at'17)*, pages 109–110. IEEE. 40, 116

Angulo, I., García-Zubía, J., Orduña, P., Rodriguez-Gil, L., and Villar, A. (2019). Integral remote laboratory for programmable logic. In *2019 5th Experiment International Conference (exp.at'19)*, pages 253–255. 177

Angulo, I., Rodríguez-Gil, L., and García-Zubía, J. (2018). Scaling up the lab: An adaptable and scalable architecture for embedded systems remote labs. *IEEE Access*, 6:16887–16900. 13, 40, 122

Anzhelika, P., Olga, G., Ivanov, E., Sokolyanskii, A., and Kurson, S. (2015). Development and application of remote laboratory for embedded systems design. In *Proceedings of 2015 12th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pages 69–73. xiii, 42, 43, 74

Aydogmus, Z. and Aydogmus, O. (2008). A web-based remote access laboratory using scada. *IEEE Transactions on education*, 52(1):126–132. 13

Banks, A. D. J., Beardall, G. D., Dennis, P. S., Dick, A. D., and Vanstone, I. C. (2013). Improving scalability and throughput of a publish/subscribe network. US Patent 8,495,127. 62

Batanero, C., de Marcos, L., Holvikivi, J., Hilera, J. R., and Otón, S. (2019). Effects of new supportive technologies for blind and deaf engineering students in online learning. *IEEE Transactions on Education*, 62(4):270–277. 12

Bencomo, S. D. (2002). Control learning: Present and future. *IFAC Proceedings Volumes*, 35(1):71 – 93. 15th IFAC World Congress. xvii, 23, 24

Benmohamed, H., Leleve, A., and Prevot, P. (2005). Generic framework for remote laboratory integration. In *2005 6th International Conference on Information Technology Based Higher Education and Training*, pages T2B/11–T2B/16. 17

Bermúdez-Ortega, J., Besada-Portas, E., López-Orozco, J., Bonache-Seco, J., and de la Cruz, J. (2015). Remote web-based control laboratory for mobile devices based on EJsS, Raspberry Pi and Node.js. *IFAC-PapersOnLine*, 48(29):158–163. IFAC Workshop on Internet Based Control Education IBCE15. 16

Bohus, C., Crowl, L. A., Aktan, B., and Shor, M. H. (1996). Running control engineering experiments over the internet. *IFAC Proceedings Volumes*, 29(1):2919–2927. 10

Bower, J. L. and Christensen, C. M. (1995). Disruptive technologies: catching the wave. *Harvard Business Review*. 10

Brinson, J. R. (2015). Learning outcome achievement in non-traditional (virtual and remote) versus traditional (hands-on) laboratories: A review of the empirical research. *Computers & Education*, 87:218 – 237. 11, 18

Brom, P., Lustig, F., Kuriščák, P., and Dvořák, J. (2018). DIY hands-on-remote experiment in physics with Arduino. In *Proceedings of the 15th International Conference on Remote Engineering and Virtual Instrumentation (REV 2018)*, pages 617–624, University of Applied Science Düsseldorf (HSD). 14, 96

Buitrago, P., Camacho, R., Orduña, P., Villar, A., Rodríguez-Gil, L., Angulo, I., and García-Zubía, J. (2018). Use of remote laboratories in engineering as an alternative to pedagogical mediation and social inclusion in distance education. In *2018 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI)*, pages 1–6. 177

Buitrago, P. A., Camacho, R., Pérez, H. E., Jaramillo, O., Villar-Martinez, A., Rodríguez-Gil, L., and Orduna, P. (2020). Mobile arduino robot programming using a remote laboratory in UNAD: Pedagogic and technical aspects. In *International Conference on Remote Engineering and Virtual Instrumentation*, pages 171–183. Springer. 116

Carlson, J. (2013). *Redis in Action*. Manning Publications, Greenwich, CT, USA. 62, 123

Chen, S., Tang, X., Wang, H., Zhao, H., and Guo, M. (2016). Towards scalable and reliable in-memory storage system: A case study with Redis. In *2016 IEEE Trustcom/BigDataSE/ISPA*, pages 1660–1667. IEEE. 62

Corbi, A. and Burgos, D. (2016). OERaaS: open educational resources as a service with the help of virtual containers. *IEEE Latin America Transactions*, 14(6):2927–2933. 89

Cordeiro, A., Fernao Pires, V., and Foito, D. (2018). Combining local and remote laboratories for the interactive learning of industrial automation. *Computer Applications in Engineering Education*, 26(3):675–687. 14

Corter, J., Nickerson, J., Esche, S., and Chassapis, C. (2004). Remote versus hands-on labs: a comparative study. In *34th Annual Frontiers in Education, 2004. FIE 2004.*, pages F1G–17. 158

Corter, J. E., Nickerson, J. V., Esche, S. K., Chassapis, C., Im, S., and Ma, J. (2007). Constructing reality: A study of remote, hands-on, and simulated laboratories. *ACM Trans. Comput.-Hum. Interact.*, 14(2):7–es. 11, 14

Costa, R., Pérola, F., and Felgueiras, C. (2020). µLAB A remote laboratory to teach and learn the ATmega328p µC. In *2020 IEEE Global Engineering Education Conference (EDUCON)*, pages 12–13. 127

Crompton, H., Burke, D., and Gregory, K. H. (2017). The use of mobile learning in pk-12 education: A systematic review. *Computers & Education*, 110:51–63. 13

Cuadros, J., Serrano, V., García-Zubía, J., and Hernandez-Jayo, U. (2021). Design and Evaluation of a User Experience Questionnaire for Remote Labs. *IEEE Access*, 9:50222–50230. xvii, 158, 159

Da Silva, J. B., Bilessimo, S. M. S., and Nardi da Silva, K. C. (2016). Mobile Remote Experimentation Workgroup (GTMRE) strategy for integrating technology into education [A estratégia de integração de tecnologia na educação do Grupo de Trabalho em Experimentação Remota Móvel (GTMRE). *Revista Tecnologias na Educação*, 17:1–14. 15, 97

De Jong, T., Linn, M. C., and Zacharia, Z. C. (2013). Physical and virtual laboratories in science and engineering education. *Science*, 340(6130):305–308. 11, 13, 18

De La Torre, L., Neustock, L. T., Herring, G. K., Chacon, J., García Clemente, F. J., and Hesselink, L. (2020). Automatic generation and easy deployment of digitized laboratories. *IEEE Transactions on Industrial Informatics*, 16(12):7328–7337. 11

de Lima, J. P. C., Carlos, L. M., Schardosim Simão, J. P., Pereira, J., Mafra, P. M., and da Silva, J. B. (2016). Design and implementation of a remote lab for teaching programming and robotics. *IFAC-PapersOnLine*, 49(30):86–91. 15, 42, 43, 45

de Vries, P., Klaasen, R., Ceulemans, D., and Ionnides, M. (2017). *Emerging Technologies in Engineering Education: Do we need them and can we make them work*. 4TU. Center for Engineering Education. 11

Dormido, S., Sánchez–Moreno, J., Vargas, H., de la Torre Cubillo, L., and Heradio, R. (2011). *UNED Labs: a network of virtual and remote laboratories*, pages 253–270. Editorial Deusto. 158

El-Abd, M. (2017). A review of embedded systems education in the Arduino age: lessons learned and future directions. *International Journal of Engineering Pedagogy*, 7(2):79–93. 41

Esche, S. K., Chassapis, C., Nazalewicz, J. W., Hromin, D. J., et al. (2003). An architecture for multi-user remote laboratories. *Dynamics (with a typical class size of 20 students)*, 5(6). 16

Fäth, T., Henke, K., Hutschenreuter, R., Seidel, F., and Wuttke, H.-D. (2018). On effective maintenance of distributed remote laboratories. In *International Conference on Remote Engineering and Virtual Instrumentation*, pages 80–89. Springer. 16

Fernandes, H., Leal, S., and Leal, J. P. (2010). E-lab: O laboratório online. *Gazeta Da Física*, 33(3):37–40. 15

Fernández-Pacheco, A., Martin, S., and Castro, M. (2019). Implementation of an Arduino Remote Laboratory with Raspberry Pi. In *2019 IEEE Global Engineering Education Conference (EDUCON)*, pages 1415–1418. IEEE. 42, 127

Flaño, T. (2020). Una startup permite a los colegios acceder online y gratis a laboratorios reales [A startup allows schools to access real laboratories online for free]. *Diario Vasco*. Accessed: 2020-09-01. 11

Fotopoulos, V., Spiliopoulos, A. I., and Fanariotis, A. (2016). Preparing a remote conducted course for microcontrollers based on Arduino. *Διεθνές Συνέδριο για την Ανοικτή & εξ Αποστάσεως Εκπαίδευση*, 7(5B). 42

Froyd, J. E., Wankat, P. C., and Smith, K. A. (2012). Five major shifts in 100 years of engineering education. *Proceedings of the IEEE*, 100(Special Centennial Issue):1344–1360. 10

Garcia-zubia, J. (2021). *Remote Laboratories: Empowering STEM Education With Technology*. World Scientific Publishing Company. 10, 11, 12

Garcia-Zubia, J., Hernandez, U., Angulo, I., Orduña, P., and Irurzun, J. (2009). Acceptance, usability and usefulness of weblab-deusto from the students point of view. *International Journal of Online Engineering*, 5(1):1–7. 13

García-Zubia, J., Orduna, P., López-de Ipiña, D., and Alves, G. R. (2009). Addressing software impact in the design of remote laboratories. *IEEE Transactions on Industrial Electronics*, 56(12):4757–4767. 49

García-Zubía, J. and Alves, G. R. (2012). *Using Remote Labs in Education: Two Little Ducks in Remote Experimentation*, volume 8. University of Deusto. 10

García-Zubía, J., Angulo, I., Martínez-Pieper, G., Orduña, P., Rodríguez-Gil, L., and Hernandez-Jayo, U. (2018). Learning to program in k12 using a remote controlled robot: Roboblock. In *Online Engineering & Internet of Things*, pages 344–358. Springer. 116

García-Zubía, J., Cuadros, J., Serrano, V., Hernández-Jayo, U., Angulo-Martínez, I., Villar, A., Orduña, P., and Alves, G. (2019). Dashboard for the visir remote lab. In *2019 5th Experiment International Conference (exp.at'19)*, pages 42–46. 177

García-Zubía, J. and Hernández-Jayo, U. (2020). Los laboratorios remotos revolucionan el aprendizaje desde casa [Remote labs revolutionize learning from home]. *The Conversation*. Accessed: 2020-09-01. 11

Gustavsson, I., Zackrisson, J., Håkansson, L., Claesson, I., and Lagö, T. L. (2007). The VISIR project - an open source software initiative for distributed online laboratories. In *REV 2007*. 16

Gómez, S., Zervas, P., Sampson, D. G., and Fabregat, R. (2014). Context-aware adaptive and personalized mobile learning delivery supported by uolmp. *Journal of King Saud University - Computer and Information Sciences*, 26(1, Supplement):47–61. Current Advances in Digital Learning Technologies. 13

Harward, V. J., Del Alamo, J. A., Lerman, S. R., Bailey, P. H., Carpenter, J., DeLong, K., Felknor, C., Hardison, J., Harrison, B., Jabbour, I., et al. (2008). The iLAB shared architecture: A web services infrastructure to build communities of Internet accessible laboratories. *Proceedings of the IEEE*, 96(6):931–950. 13, 25

Henke, K., Vietzke, T., Hutschenreuter, R., and Wuttke, H.-D. (2016a). The remote lab cloud GOLDi-labs.net. In *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pages 37–42. 16

Henke, K., Vietzke, T., Wuttke, H.-D., and Ostendorff, S. (2016b). GOLDi-Grid of online lab devices Ilmenau. *Int. J. Online Eng.*, 12(4):11–13. 14, 16

Henke, K., Wuttke, H., Hutschenreuter, R., and Streitferdt, D. (2019). Interactive Content Objects as GOLDi-Lab Services : Interactive Demonstration of ICOs within the Hybrid Online Online Lab GOLDi. In *2019 5th Experiment International Conference (exp.at'19)*, pages 229–230. 97

Henriques, R., Pereira, J., Dias, F., Fernandes, H., Duarte, A., Pereira, T., and Fortunato, J. (2015). A generic communication protocol for remote laboratories: An implementation on e-lab. In *2015 4th International Conference on Advancements in Nuclear Instrumentation Measurement Methods and their Applications (ANIMMA)*, pages 1–6. IEEE. 16

Hu, M., Hu, W., and Zhou, H. (2016). Ncslab remote experimentation for control engineering education in wuhan university. In *2016 35th Chinese Control Conference (CCC)*, pages 7416–7421. 158

Hussein, R. and Wilson, D. (2021). Remote versus in-hand hardware laboratory in digital circuits courses. In *2021 ASEE Virtual Annual Conference Content Access*. 11, 164

Jara, C. A., Candelas, F. A., and Torres, F. (2008). Virtual and remote laboratory for robotics e-learning. In *Computer Aided Chemical Engineering*, volume 25, pages 1193–1198. Elsevier. 67

Jona, K., Walter, A. D., and Pradhan, S. N. (2015). Designing remote labs for broader adoption. In *Proceedings of 2015 12th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pages 74–80. 16

Kwinana, P. M., Nomnga, P., Rani, M., and Lekala, M. L. (2020). Real laboratories available online: Establishment of ReVEL as a conceptual framework for implementing remote experimentation in South African higher education institutions and rural-based schools–a case study at the University of Fort Hare. In *International Conference on Remote Engineering and Virtual Instrumentation*, pages 128–142. Springer. 116

Lahoud, H. A. and Krichen, J. P. (2010). Networking labs in the online environment: Indicators for success. *Journal of Technology Studies*, 36(2):31–40. 12, 13, 16, 18, 92

Lang, D., Mengelkamp, C., Jäger, R., Geoffroy, D., Billaud, M., and Zimmer, T. (2007). Pedagogical evaluation of remote laboratories in emerge project. *European Journal of Engineering Education*, 32:57–72. 158

Lavayssière, C., Larroque, B., and Luthon, F. (2022). Laborem Box: A scalable and open source platform to design remote lab experiments in electronics. *HardwareX*, 11:e00301. 15

Leal, S. and Leal, J. P. (2013). A new chemistry e-lab experiment chemical equilibirum reaction. In *2013 2nd Experiment@ International Conference (exp.at'13)*, pages 154–155. 15

Letowski, B., Lavayssière, C., Larroque, B., and Luthon, F. (2019). An open source remote laboratory network based on a ready to use solution: Laborem. In *ICERI 2019*, Seville, Spain. 15

Liggesmeyer, P. and Trapp, M. (2009). Trends in embedded software engineering. *IEEE software*, 26(3):19–25. 47

Lima, N., Viegas, C., and Garcia-Peñalvo, F. (2019). Didactical use of a remote lab: A qualitative reflection of a teacher. In *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, page 99–108, New York, NY, USA. Association for Computing Machinery. 16

Linden, A. (2020). UFH ReVEL online technology prepares to provide support during lockdown. *University of Fort Hare News*. Accessed: 2020-09-01. 11

Lindsay, E. and Stumpers, B. (2011). Remote laboratories: enhancing accredited engineering degree programs. In *Proceedings of the 22nd Annual Conference for the Australasian Association for Engineering Education, AAEE 2011*, pages 588–593. Engineers Australia. 14

Lowe, D., Murray, S., Weber, L., de la Villefromoy, M., Johnston, A., Lindsay, E., Nageswaran, W., and Nafalski, A. (2009). LabShare: Towards a national approach to laboratory sharing. In *Proceedings of the 20th Annual Conference for the Australasian Association for Engineering Education*, pages 458–463. The School of Mechanical Engineering, University of Adelaide. 13, 17

Lustig, F., Brom, P., Kuriscak, P., and Dvorak, J. (2018). "Hands-on-Remote" laboratories. In *International Conference on Remote Engineering and Virtual Instrumentation*, pages 118–127. Springer. 14

Ma, J. and Nickerson, J. V. (2006). Hands-on, simulated, and remote laboratories: A comparative literature review. *ACM Computing Surveys (CSUR)*, 38(3):7–es. 11, 18

Maiti, A., Maxwell, A. D., and Kist, A. A. (2014). Features, trends and characteristics of remote access laboratory management systems. *International Journal of Online Engineering*, 10(2):30–37. 13, 17

Maksimović, M., Vujović, V., Davidović, N., Milošević, V., and Perišić, B. (2014). Raspberry Pi as Internet of Things hardware: Performances and constraints. In *Proceedings of the 1st International Conference on Electrical, Electronic and Computing Engineering, IcETRAN2014*, volume 3. 48

Marchisio, S. T., Lerro, F. G., et al. (2015). The FCEIA-UNR remote laboratory: Integration of resources and collaborative networking for Engineering Education [El laboratorio remoto FCEIA-UNR: Integración de recursos y trabajo en redes colaborativas para la enseñanza de la ingeniería]. In *Quinta conferencia de directores de tecnología de información TICAL 2015 Gestión de las TICs para la investigación y la colaboración*. 17

Mostefaoui, H. and Benachenhou, A. (2015). Design of a remote electronic laboratory. In *2015 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL)*, pages 160–162. IEEE. 42, 127

Müller, D. and Erbe, H.-H. (2007). Collaborative remote laboratories in engineering education: Challenges and visions. *Advances on remote laboratories and e-learning experiences*, pages 35–59. xiii, 27, 28

Muzaffar, A. W., Tahir, M., Anwar, M. W., Chaudry, Q., Mir, S. R., and Rasheed, Y. (2021). A systematic review of online exams solutions in e-learning: Techniques, tools, and global adoption. *IEEE Access*, 9:32689–32712. 12

Nedic, Z., Machotka, J., and Nafalski, A. (2003). Remote laboratories versus virtual and real laboratories. In *33rd Annual Frontiers in Education, 2003. FIE 2003.*, volume 1, pages T3E–T3E. 11, 14

Nickerson, J. V., Corter, J. E., Esche, S. K., and Chassapis, C. (2007). A model for evaluating the effectiveness of remote engineering laboratories and simulations in education. *Comput. Educ.*, 49(3):708–725. 11, 14

Nuci, K. P., Tahir, R., Wang, A. I., and Imran, A. S. (2021). Game-based digital quiz as a tool for improving students' engagement and learning in online lectures. *IEEE Access*, 9:91220–91234. 12

Orduña, P., Rodriguez-Gil, L., Garcia-Zubia, J., Angulo, I., Hernandez, U., and Azcuenaga, E. (2018). Increasing the value of remote laboratory federations through an open sharing platform: Labsland. In Auer, M. E. and Zutin, D. G., editors, *Online Engineering & Internet of Things*, pages 859–873, Cham. Springer International Publishing. 13, 18, 67

Orduña, P. (2013). *Transitive and Scalable Federation Model for Remote Laboratories*. PhD thesis, University of Deusto. 66, 124, 142

Orduña, P., Garcia-Zubia, J., Rodriguez-Gil, L., Angulo, I., Hernandez-Jayo, U., Dziabenko, O., and López-de Ipiña, D. (2018a). The weblab-deusto remote laboratory management system architecture: Achieving scalability, interoperability, and federation of remote experimentation. In *Cyber-Physical*

Laboratories in Engineering and Science Education, pages 17–42. Springer. 16, 17, 66

Orduña, P., Irurzun, J., Rodriguez-Gil, L., García-Zubía, J., Gazzola, F., and López-de Ipiña, D. (2011). Adding new features to new and existing remote experiments through their integration in WebLab-Deusto. *International Journal of Online and Biomedical Engineering (iJOE)*, 7(S2):33–39. 13, 97

Orduña, P., Rodriguez-Gil, L., Angulo, I., Hernandez, U., Villar, A., and Garcia-Zubia, J. (2019). WebLabLib: new approach for creating remote laboratories. In *International conference on remote engineering and virtual instrumentation*, pages 477–488. Springer. 17, 66

Orduña, P., Rodriguez-Gil, L., Angulo, I., Martinez, G., Villar, A., Hernandez, U., Buitrago, P., Camacho, R., Marmolejo, P., and Garcia-Zubia, J. (2018b). Addressing technical and organizational pitfalls of using remote laboratories in a commercial environment. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–7. 177

Orduña, P., Rodriguez-Gil, L., Garcia-Zubia, J., Angulo, I., Hernandez, U., and Azcuenaga, E. (2016). LabsLand: A sharing economy platform to promote educational remote laboratories maintainability, sustainability and adoption. In *2016 IEEE Frontiers in Education Conference (FIE)*, pages 1–6. IEEE. 12, 18, 92, 97

Orduña, P., Rodriguez-Gil, L., López-de Ipiña, D., and García-Zubia, J. (2012). Sharing the remote laboratories among different institutions: A practical case. In *2012 9th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pages 1–4. 13

Ožvoldová, M. and Schauer, F. (2016). *Remote Laboratories: in Research-based education of real world phenomena*. Peter Lang Publishers. 97

Paravati, G., Celozzi, C., Sanna, A., and Lamberti, F. (2010). A feedback-based control technique for interactive live streaming systems to mobile devices. *IEEE Transactions on Consumer Electronics*, 56(1):190–197. 13

Pennisi, E. (2020). During the pandemic, students do field and lab work without leaving home. *SCIENCE Magazine*. Accessed: 2020-09-01. 11

Potkonjak, V., Gardner, M., Callaghan, V., Mattila, P., Guetl, C., Petrović, V. M., and Jovanović, K. (2016). Virtual laboratories for education in science, technology, and engineering: A review. *Computers & Education*, 95:309–327. 11

Pretz, K. (2020). German university opens up its hands-on remote FPGA lab during the coronavirus pandemic. *IEEE Spectrum*. Accessed: 2020-09-01. 11

Ramo, S. (1957). A new technique for education. *Engineering and Science*, 21:17 – 22. 10

Rodriguez-Sánchez, M. C., Torrado-Carvajal, A., Vaquero, J., Borromeo, S., and Hernandez-Tamames, J. A. (2016). An embedded systems course for engineering students using open-source platforms in wireless scenarios. *IEEE Transactions on Education*, 59(4):248–254. 41

Rodríguez-Gil, L. (2017). *Improving the remote laboratory experience through augmented characteristics beyond the experiment core*. PhD thesis, University of Deusto. 52, 67

Rodríguez-Gil, L., García-Zubia, J., Orduña, P., and López-de Ipiña, D. (2017a). An open and scalable web-based interactive live-streaming architecture: The WILSP platform. *IEEE Access*, 5:9842–9856. 24, 66, 67

Rodríguez-Gil, L., García-Zubia, J., Orduña, P., and López-de Ipiña, D. (2017b). Towards new multiplatform hybrid online laboratory models. *IEEE Transactions on Learning Technologies*, 10(3):318–330. xvii, 24

Rodríguez-Gil, L., García-Zubia, J., Orduña, P., Villar-Martinez, A., and López-De-Ipiña, D. (2019). New approach for conversational agent definition by non-programmers: A visual domain-specific language. *IEEE Access*, 7:5262–5276. 177

Sáenz, J., Chacón, J., De La Torre, L., Visioli, A., and Dormido, S. (2015). Open and low-cost virtual and remote labs on control engineering. *IEEE Access*, 3:805–814. 13

Salzmann, C. and Gillet, D. (2007). Challenges in remote laboratory sustainability. In *Proceedings of the International Conference on Engineering Education ICEE 2007*. 18

Samuelsen, D. A. and Graven, O. H. (2013). Design of a general purpose platform for easy setup of low-cost remote laboratories in electronics. In *2013 10th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pages 1–6. 15

Sarik, J. and Kymissis, I. (2010). Lab kits using the Arduino prototyping platform. In *2010 IEEE Frontiers in Education Conference (FIE)*, pages T3C–1–T3C–5. 42

Sauter, M., Uttal, D. H., Rapp, D. N., Downing, M., and Jona, K. (2013). Getting real: the authenticity of remote labs and simulations for science learning. *Distance Education*, 34(1):37–47. 11

Sawahel, W. (2020). Covid-19 drives development of online laboratories. *IEEE Spectrum*. Accessed: 2020-09-01. 11

Schauer, F., Krbecek, M., Beno, P., Gerza, M., Palka, L., and Spilakovà, P. (2014). REMLABNET-open remote laboratory management system for e-experiments. In *2014 11th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pages 268–273. IEEE. 13, 14, 17

Schauer, F., Krbecek, M., Beno, P., Gerza, M., Palka, L., and Spilaková, P. (2015). REMLABNET II - open remote laboratory management system for university and secondary schools research based teaching. In *Proceedings of 2015 12th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pages 109–112. 17

Schauer, F., Krbecek, M., Beno, P., Gerza, M., Palka, L., Spilaková, P., and Tkac, L. (2016). REMLABNET III — federated remote laboratory management system for university and secondary schools. In *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pages 238–241. 14, 17

Schauer, F., Kuřitka, I., Lustig, F., et al. (2006). Creative laboratory experiments for basic physics using computer data collection and evaluation exem-

plified on the intelligent school experimental system (ISES). *Innovations*, pages 305–312. 14

Schauer, F., Ožvoldová, M., Gerža, M., Krbeček, M., Beňo, P., Komenda, T., Das, S., and Archibong, M. I. (2017). REMLABNET IV-LTI federated remote laboratory management system with embedded multiparameter simulations. *International Journal of Online Engineering*. 17

Schwandt, A. and Winzker, M. (2019). Make it open - improving usability and availability of an FPGA remote lab. In *2019 IEEE Global Engineering Education Conference (EDUCON)*, pages 232–236. 17

Spilakova, P. and Schauer, F. (2015). Remote laboratory management system RemLabNet and its booking system. In *2015 Forth International Conference on e-Technologies and Networks for Development (ICeND)*, pages 1–5. 17

Sáenz, J., de la Torre, L., Chacón, J., and Dormido, S. (2021). A study of strategies for developing online laboratories. *IEEE Transactions on Learning Technologies*, 14(6):777–787. 16, 19

Tawfik, M., Lowe, D., Murray, S., de la Villefromoy, M., Diponio, M., Sancristobal, E., Albert, M. J., Díaz, G., and Castro, M. (2013). Grid Remote Laboratory Management System. In *2013 10th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pages 1–9. 13

Thoms, L.-J. and Girwidz, R. (2017). Experiments on optical spectrometry in the VirtualRemoteLab. In *2017 4th Experiment@ International Conference (exp. at'17)*, pages 147–148. IEEE. 15

Vargas, H., Farias, G., Sanchez, J., Dormido, S., and Esquembre, F. (2013). Using augmented reality in remote laboratories. *International Journal of Computers Communications & Control*, 8(4):622–634. 67

Villar-Martínez, A., García-Zubía, J., Angulo, I., and Rodríguez-Gil, L. (2021). Towards reliable remote laboratory experiences: A model for maximizing availability through fault-detection and replication remote laboratory experiences: A model for maximizing availability through fault-detection and replication. *IEEE Access*, 9:45032–45054. 15, 21

Villar-Martínez, A., García-Zubía, J., Angulo, I., and Rodríguez-Gil, L. (2022). Toward widespread remote laboratories: Evaluating the effectiveness of a replication-based architecture for real-world multiinstitutional usage. *IEEE Access*, 10:86298–86317. 142

Villar-Martínez, A., Rodríguez-Gil, L., Angulo, I., Orduña, P., García-Zubía, J., and López-De-Ipiña, D. (2019). Improving the scalability and replicability of embedded systems remote laboratories through a cost-effective architecture. *IEEE Access*, 7:164164–164185. xiv, 16, 21, 118, 119, 122

Wang, N., Chen, X., Lan, Q., Song, G., Parsaei, H., and Ho, S.-C. (2016). A novel wiki-based remote laboratory platform for engineering education. *IEEE Transactions on Learning Technologies*, 10:1–1. 16

Weintrop, D. and Wilensky, U. (2017). Comparing Block-based and Text-based Programming in High School Computer Science Classrooms. *ACM Transactions on Computing Education (TOCE)*, 18(1):1–25. 65

Werner, S., Lauber, A., Becker, J., and Sax, E. (2016). Cloud-based remote virtual prototyping platform for embedded control applications: cloud-based infrastructure for large-scale embedded hardware-related programming laboratories. In *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pages 168–175. IEEE. 42

Winzker, M., Kiessling, R., Schwandt, A., Paez, C. S., and Shanab, S. A. (2018). Teaching across the ocean with video lectures and remote-lab. In *2018 IEEE World Engineering Education Conference (EDUNINE)*, pages 1–4. 17

Winzker, M. and Schwandt, A. (2019). Open education teaching unit for low-power design and FPGA image processing. In *2019 IEEE Frontiers in Education Conference (FIE)*. 17

Yazidi, A., Henao, H., Capolino, G.-A., Betin, F., and Filippetti, F. (2011). A web-based remote laboratory for monitoring and diagnosis of ac electrical machines. *IEEE Transactions on Industrial Electronics*, 58(10):4950–4959. 13, 67

Yeung, H., Lowe, D. B., and Murray, S. (2010). Interoperability of remote laboratories systems. *International Journal of Online Engineering*. 17

Esta tesis fue terminada en Bilbao el 1 de septiembre de 2022