

*A mi madre
y a Isi*

Resumen

La relevancia hoy en día de las comunicaciones móviles, su alto grado de penetración en sectores de actividad económica muy diversos y las limitaciones presentes dentro de esta área, debido a las propias limitaciones de los dispositivos móviles, de las redes inalámbricas y de la movilidad, han motivado la realización de esta tesis.

La comunidad científica estudia actualmente posibles soluciones a estos inconvenientes pero muy pocas exploran como solución el camino de la unificación de las dos grandes áreas tratadas en esta tesis: Soporte a las aplicaciones móviles y gestión de los recursos de los dispositivos móviles. Esta unificación posibilita el crecimiento de los servicios móviles más específicos y complejos proporcionados para este tipo de entornos tan limitados. Así, definimos una solución de movilidad que gestiona de manera inteligente estas limitaciones y las aplicaciones móviles a las que da soporte.

El carácter innovador de la misma lo aporta el hecho de que se basa en las necesidades de las aplicaciones y en el estado de los recursos para lograr una gestión más óptima tanto de las aplicaciones como de los recursos. Para ello, hemos procedido a la definición de un modelo de soporte para las aplicaciones móviles y a la definición de un lenguaje de representación de las necesidades de las aplicaciones basado en XML que sirva como base para la especificación de sus necesidades. Destacar además que nos hemos centrado en la gestión de dos recursos: las comunicaciones y la energía, definiendo diversas técnicas a nivel de middleware que logran optimizar el uso de dichos recursos. Por otra parte, es reseñable la aportación de una nueva metodología de movilización de aplicaciones para la solución propuesta orientada tanto al diseño de nuevas aplicaciones como a la movilización de aplicaciones existentes.

Todas las bondades enunciadas han sido probadas experimentalmente, por medio de una serie de escenarios y de una plataforma de medida definida de manera explícita para entornos tan limitados como una PDA, consiguiendo así probar que se puede lograr una gestión más óptima de las aplicaciones y de los recursos buscando el equilibrio entre las necesidades de las mismas y las limitaciones de los recursos.

Abstract

This dissertation has been motivated by the current high relevance of mobile communications, their high degree of penetration in many economic sectors and the still unresolved limitations associated to this area, such as the limitations of mobile devices, wireless networks and mobility.

The scientific community is currently studying possible solutions to these drawbacks but very few have considered as a solution the unification of the two major areas dealt within this dissertation: support for mobile applications and resource management for mobile devices. This unification enables the growth of the most specific and complex mobile services provided for this kind of environments. In this work, a mobile solution is proposed which manages intelligently these limitations and the ones associated to the mobile applications supported by it.

Its main novelty is the use of both the application needs and the status of resources to obtain a more optimal management of applications and resources. For that, a support model for mobile applications together with an XML-based language to represent the needs of applications has been defined. A special emphasis has been paid to the management of two particular resources: communications and energy, defining several techniques at the middleware level which optimise the usage of such resources. On the other hand, it is worth noting the definition of a new methodology to mobilize applications oriented both to the design of new applications and to the mobilization of already existing ones.

All the advantages mentioned above have been tested experimentally, by means of a set of scenarios and a measurement platform defined in an explicit manner for environments so limited as a PDA, proving that it is possible to obtain a more optimal management of applications and resources still finding a balance between their needs and resource limitations.

Agradecimientos

Quiero agradecer desde estas líneas a todas las personas que han contribuido en mayor o menor medida a la realización de esta tesis doctoral.

En primer lugar, debo dar las gracias a mi director de tesis, José Luis del Val Román, cuyo estímulo constante, apoyo incondicional y su comprensión han sido decisivos.

Gracias también a mis compañeros de la facultad, en especial a Asier Perallos, Iñaki Vázquez, Pablo G. Bringas, David Buján y Diego López de Ipiña, con quienes he pasado momentos inolvidables y considero parte de mi familia, por su apoyo en el día a día y porque desde que los conocí han estado ahí con un consejo o alguna palabra de aliento.

Gracias a Iker Larizgoitia por su colaboración entusiasta en ésta y otras investigaciones, por su optimismo, pasión y compromiso, y porque sin él habría sido imposible conseguir un acabado tan digno.

Gracias a Roberto Trueba, Ignacio Uria, Nacho Angulo, Jonathan Ruíz de Garibay y Nekane Sainz, por el trabajo inmejorable realizado en algunas de las tareas de la tesis. Muchas gracias por vuestro tiempo.

Finalmente, dar gracias a mi familia, en especial, a mi marido por su constante presencia y aliento durante estos últimos años, donde ha tenido que sufrir mi inevitable estrés, y a mi hija recién nacida, que desde que tuve conocimiento de su venida, ha sido determinante para afrontar con ganas e ilusión la última parte de la realización de esta tesis.

¡¡¡Muchas gracias!!!

Tabla de Contenidos

1. INTRODUCCIÓN.....	1
1.1 Computación móvil.....	2
1.1.1 Limitaciones de la Computación móvil	4
1.1.1.1 Limitaciones de los dispositivos móviles	5
1.1.1.2 Limitaciones de las comunicaciones inalámbricas	5
1.1.1.3 Limitaciones de la movilidad.....	9
1.1.2 Computación móvil y sus ámbitos de aplicación	11
1.1.3 Conclusiones.....	14
1.2 Hipótesis y Objetivos	14
1.2.1 Hipótesis.....	14
1.2.2 Objetivos.....	15
1.2.2.1 Objetivo general.....	15
1.2.2.2 Objetivos específicos y operacionales	17
1.3 Metodología de la investigación.....	18
1.4 Estructura de la memoria	21
2. SOLUCIONES DE MOVILIDAD Y GESTIÓN DE RECURSOS	25
2.1 Introducción	25
2.2 Tipos de aplicaciones móviles.....	26
2.3 Modelos de adaptación a la movilidad.....	29
2.4 Requisitos de una solución de movilidad	32

2.5 Operación online/offline	33
2.5.1 Enfoque orientado a datos	34
2.5.1.1 Base de datos móviles	35
2.5.1.2 Estrategias de diseño de base de datos móviles	36
2.5.1.3 Sincronización	39
2.5.2 Enfoque orientado a servicios	55
2.5.2.1 Soluciones de integración de procesos de negocio	56
2.5.2.2 Arquitectura orientada a servicio	57
2.5.2.3 Servicios web	59
2.5.2.4 Integración con servicios web	60
2.5.2.5 Integración en entornos móviles	61
2.6 Gestión de la energía en la computación móvil	65
2.6.1 Relación entre potencia y energía	67
2.6.2 Fuentes de energía: las baterías	68
2.6.2.1 Baterías Li-ion	70
2.6.3 Consumidores	72
2.6.3.1 Procesadores	72
2.6.3.2 Disco duro	73
2.6.3.3 Comunicación	74
2.6.3.4 Interfaces de usuario	75
2.6.4 Gestión de la energía en todos los niveles	76
2.6.4.1 Hardware	77
2.6.4.2 Sistema Operativo	80
2.6.4.3 Gestión de energía en redes inalámbricas	87
2.6.4.4 Nivel de aplicación y nivel de middleware	95
2.7 Gestión inteligente de las comunicaciones	99
3. ESTUDIO DE LAS SOLUCIONES DE MOVILIDAD Y GESTIÓN DE RECURSOS PREVIOS	105
3.1 Aspectos a analizar	106

3.2 Soluciones analizadas.....	109
3.2.1 Puppeteer	109
3.2.2 Odyssey/Spectra.....	112
3.2.3 ECOSystem	113
3.2.4 Mohapatra.....	114
3.2.5 Pagoda.....	117
3.2.6 Framework Grace.....	119
3.2.7 PARM	121
3.2.8 STPM I.....	123
3.2.9 STPM II.....	124
3.2.10 Cooperative I/O	126
3.2.11 Poellabauer	126
3.2.12 PAWP	129
3.2.13 DSM Framework.....	130
3.2.14 Framework Thor	131
3.2.15 FarGo-DA.....	133
3.2.16 Twin Agents.....	134
3.2.17 Efstratiou	135
3.2.18 Replets System.....	135
3.2.19 WebPADS.....	137
3.2.20 Framework de IBM.....	139
3.2.21 Domint Platform	140
3.2.22 IMMS Framework.....	141
3.2.23 SodaSync Framework.....	141
3.3 Comparativa.....	143
3.4 Solución propuesta	147
4. MOSE: UNA SOLUCIÓN PARA LA GESTIÓN DE APLICACIONES MÓVILES Y RECURSOS	
 SENSIBLE A LAS APLICACIONES	149
4.1 Requisitos de la solución	149

4.2	Introducción a la arquitectura de MOSE	151
4.3	Modelo teórico	156
4.3.1	Soporte a la Operación online/offline.....	156
4.3.1.1	Estudio sobre las necesidades de movilidad empresariales	157
4.3.1.2	Modelo teórico	161
4.3.2	Núcleo.....	166
4.3.2.1	El lenguaje ANE-SML	166
4.3.2.2	Clasificación de las invocaciones a servicios.....	176
4.3.2.3	Algoritmo de gestión inteligente de los recursos y aplicaciones basado en las necesidades de las aplicaciones.....	181
4.4	Diseño detallado	212
4.4.1	Consideraciones de diseño generales.....	213
4.4.2	Soporte a la Operación online/offline.....	214
4.4.2.1	Gestión de datos.....	215
4.4.2.2	Gestión de servicios.....	221
4.4.3	Núcleo.....	234
4.4.3.1	Diseño del gestor central	235
4.4.3.2	Diseño del gestor de aplicaciones.....	236
4.4.4	Gestión de recursos.....	245
4.4.4.1	Gestor de batería	246
4.4.4.2	Gestor de comunicaciones	248
5.	M2A: METODOLOGÍA DE MOVILIZACIÓN DE APLICACIONES.....	251
5.1	Definición de la Metodología de Movilización de Aplicaciones basada en el middleware MOSE (M2A).....	251
5.2	Fases de M2A	252
5.2.1	Clasificación de la aplicación a movilizar.....	252
5.2.2	Adaptación de la aplicación al enfoque orientado a datos.....	253
5.2.2.1	Conceptos previos al diseño de la base de datos.....	254
5.2.2.2	Proceso de adaptación de la base de datos a la sincronización.....	259

5.2.3	Modelizar el comportamiento de la aplicación mediante el lenguaje ANE-SML	265
5.2.3.1	Definición de la política con respecto a la gestión de datos.....	265
5.2.3.2	Definición de la política con respecto a la gestión de servicios.....	267
5.2.3.3	Definición de la política con respecto a la configuración global de la aplicación.....	269
5.3	Aplicación de la metodología M2A al desarrollo de una aplicación móvil concreta	269
5.3.1	Descripción del sistema de O&M y de la aplicación a movilizar	270
5.3.1.1	Propósito del sistema	270
5.3.2	Aplicación de la metodología M2A a la aplicación de O&M.....	271
5.3.2.1	Adaptación de la aplicación al enfoque orientado a datos	272
5.3.2.2	Modelizar el comportamiento de la aplicación mediante el lenguaje ANE-SML.....	283
6.	EXPERIMENTACIÓN Y VALIDACIÓN DE LA HIPÓTESIS	287
6.1	Implementación de la plataforma.....	287
6.1.1	Configuración de la plataforma de experimentación.....	287
6.1.2	Consideraciones de implementación	289
6.1.2.1	El proceso de implementación	289
6.1.2.2	Entorno de desarrollo.....	290
6.1.2.3	Distribución de los módulos sobre la plataforma de experimentación....	291
6.1.3	Incidencias encontradas	293
6.2	Evaluación	294
6.2.1	Aplicaciones desarrolladas	294
6.2.1.1	Aplicación móvil Ddetective	294
6.2.2	Plataforma de medida	297
6.2.3	Pruebas	301
6.2.3.1	Cálculo del idleTime	305
6.2.3.2	Escenario 1	306
6.2.3.3	Escenario 2	312

6.2.3.4 Escenario 3.....	319
6.2.3.5 Escenario 4.....	325
6.2.3.6 Escenario 5.....	330
6.2.4 Conclusiones generales.....	336
7. CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO.....	339
7.1 Conclusiones.....	339
7.1.1 Cumplimiento de los requisitos.....	340
7.1.2 Consecución de los objetivos operacionales.....	340
7.1.3 Proyectos de investigación relacionados y publicaciones.....	343
7.2 Líneas de investigación futuras.....	343
REFERENCIAS BIBLIOGRÁFICAS.....	347
APÉNDICE A.....	375

Índice de Figuras

Figura 1.1 Técnicas incluidas en la Computación móvil [Satyanarayana01].	4
Figura 1.2 Comparativa de tecnologías de acceso inalámbrico en Europa.	6
Figura 1.3 Movilidad entre <i>hot-spots</i> .	10
Figura 1.4 Movilidad entre operadores WiFi.	11
Figura 1.5 Aspectos principales abordados en la tesis.	15
Figura 1.6 Clasificación de los requisitos operacionales en áreas.	16
Figura 1.7 Etapas generales del proceso de investigación.	19
Figura 1.8 Fases y carácter cíclico del método investigación-acción.	21
Figura 2.1 Clasificación de la adaptación.	31
Figura 2.2 Cluster y jerarquía.	41
Figura 2.3 Arquitectura de sincronización.	42
Figura 2.4 Comunicación síncrona y asíncrona.	46
Figura 2.5 Sincronización Store-and-Forward.	47
Figura 2.6 Ejemplo de diseño de datagroups.	48
Figura 2.7 Integración B2B basada en servicios web.	60
Figura 2.8 Integración EAI basada en servicios web.	61
Figura 2.9 Arquitectura de WSMQ.	63
Figura 2.10 Evolución de las baterías. Fuente: Boston Consulting Group.	70
Figura 2.11 Características de descarga de Li-Ion con cobre y con grafito.	71
Figura 2.12 Tendencias de consumo de energía para los microprocesadores Intel [Gunther+01].	73
Figura 2.13 Consumo de diferentes interfaces inalámbricas [Simunic05].	74
Figura 2.14 Consumo de los diferentes componentes de una iPAQ [Zhong05].	74
Figura 2.15 Consumo de energía de Bluetooth [Zhong05].	75

Figura 2.16 Gestión de la energía en desde el punto de vista de la carga de trabajo, el dispositivo y los estados de energía [Lu+01].	82
Figura 2.17 Consumo de un dispositivo en estado working vs. en estado sleep.	83
Figura 2.18 Gestor de energía.	84
Figura 2.19 Nivel de aplicación de la gestión de energía [Jones+01].	88
Figura 3.1 Arquitectura de Puppeteer.	109
Figura 3.2 Arquitectura de Spectra.	112
Figura 3.3 Framework ECOSystem.	114
Figura 3.4 Adaptaciones a bajo nivel y a alto nivel para optimizar la energía y el rendimiento en streaming de vídeo.	116
Figura 3.5 Modelo del sistema.	116
Figura 3.6 Arquitectura de Pagoda.	117
Figura 3.7 Adaptación basándose en la velocidad de la CPU, la asignación de CPU y la calidad de vídeo.	119
Figura 3.8 Framework de adaptación Grace.	120
Figura 3.9 Arquitectura de PARM.	121
Figura 3.10 Arquitectura de STPM.	123
Figura 3.11 Arquitectura de STPM II.	124
Figura 3.12 Localización del Proxy PAWP.	129
Figura 3.13 Arquitectura de PAWP.	130
Figura 3.14 Reglas de funcionamiento de PAWP.	130
Figura 3.15 Arquitectura del framework DSM.	131
Figura 3.16 Arquitectura de Twin Agents.	134
Figura 3.17 Arquitectura del sistema propuesto por Efstratiou et al.	135
Figura 3.18 Arquitectura del sistema WebPADS.	138
Figura 4.1 Clasificación de los requisitos operacionales en áreas.	150
Figura 4.2 Localización de MOSE.	152
Figura 4.3 Distribución general de la arquitectura de la plataforma MOSE.	152
Figura 4.4 Arquitectura de MOSE en el cliente junto con su relación con los RO y RA.	153
Figura 4.5 Arquitectura de MOSE en el servidor.	155
Figura 4.6 Presencia de PYMES en la CAPV y en sus territorios [Eustat06].	157
Figura 4.7 Presencia de las PYMES por sectores en la CAPV [Eustat06].	158
Figura 4.8 Componentes del enfoque orientado a datos.	161
Figura 4.9 Componentes del enfoque orientado a servicios.	163

Figura 4.10 Componentes del enfoque orientado a servicios con Servidor Proxy.	164
Figura 4.11 Componentes del enfoque orientado a servicios con técnica de cacheo.	165
Figura 4.12 Arquitectura de MOSE.....	167
Figura 4.13 Esquema del lenguaje ANE-SML.	169
Figura 4.14 Ejemplo de especificación de necesidades de la gestión de datos basada en el lenguaje ANE-SML.	171
Figura 4.15 Tiempos a considerar en las invocaciones a servicios.....	173
Figura 4.16 Ejemplo de especificación de necesidades de la gestión de servicios basada en el lenguaje ANE-SML.	174
Figura 4.17 Aspectos desarrollados en el algoritmo junto con su localización.....	184
Figura 4.18 Descomposición en módulos del algoritmo.....	189
Figura 4.19 Ejemplo del cálculo de Tcom.	191
Figura 4.20 Ejemplo de replanificación de Tcom.....	191
Figura 4.21 Componentes del gestor de aplicaciones.....	193
Figura 4.22 Notificación de cambio en la base de datos.....	195
Figura 4.23 Gestión de la energía en desde el punto de vista de la carga de trabajo, el dispositivo y los estados de energía [Lu+01].	199
Figura 4.24 Consumo de un recurso inactivo en el estado working.	200
Figura 4.25 Consumo de un recurso al cambiar al estado sleep.	201
Figura 4.26 Periodo de inactividad.....	202
Figura 4.27 Técnica para maximizar el ahorro de energía.....	204
Figura 4.28 Ejemplo de la técnica de ahorro de energía basada en garantizar el cambio de estado.....	207
Figura 4.29 Funcionamiento del algoritmo.....	208
Figura 4.30 Ejemplo de aplicación del tiempo de reintento.	213
Figura 4.31 Comunicación entre cliente y servidor.	214
Figura 4.32 Submódulos del módulo “Soporte a la operación online/offline”.	215
Figura 4.33 Diagrama de clases del mecanismo de acceso a los datos.....	216
Figura 4.34 Diagrama de clases del mecanismo para el mantenimiento de los datos.	218
Figura 4.35 Diagrama de clases del Mecanismo de mantenimiento de datos para SQL Server CE.....	219
Figura 4.36 Módulos de soporte del servidor.....	220
Figura 4.37 Arquitectura de un sistema de sincronización.	220
Figura 4.38 Diagrama de clases del sistema de almacenamiento.	222
Figura 4.39 Diagrama de estados de las invocaciones a servicios.	223
Figura 4.40 Diagrama de clases del mecanismo de almacenamiento de invocaciones.....	224

Figura 4.41 Diagrama de clases del mecanismo de envío y recepción de invocaciones.	225
Figura 4.42 Arquitectura del servidor Proxy.	227
Figura 4.43 Diagrama de clases del procesador de invocaciones.	231
Figura 4.44 Diagrama de clases de los Adaptadores de invocación.	233
Figura 4.45 Componentes de la solución propuesta.	235
Figura 4.46 Diagrama de clases del Gestor Central.	236
Figura 4.47 Diagrama de clases del gestor de aplicaciones.	237
Figura 4.48 Diagrama de clases del gestor de políticas de datos.	238
Figura 4.49 Diagrama de clases de la gestión de las invocaciones.	240
Figura 4.50 Diagrama de clases de la gestión del ciclo de vida de las invocaciones.	241
Figura 4.51 Diagrama de estados de las invocaciones a servicios.	242
Figura 4.52 Diagrama de clases del Cálculo y notificación del siguiente tiempo de invocación.	245
Figura 4.53 Diagrama de clases del Gestor de batería.	247
Figura 4.54 Diagrama de clases del Gestor de comunicaciones.	248
Figura 5.1 Aplicación de M2A dentro del ciclo de desarrollo de una aplicación.	253
Figura 5.2 Ficha de análisis de la base de datos a sincronizar.	261
Figura 5.3 Diagrama de flujo de preparación de las tablas para la sincronización.	263
Figura 5.4 Ficha de diseño de la base de datos a sincronizar.	264
Figura 5.5 Ficha de estimación de tiempos de sincronización.	266
Figura 5.6 Ficha de estimación de tiempos de invocación de servicios.	268
Figura 5.7 Arquitectura general del sistema.	271
Figura 5.8 Esquema inicial de la base de datos a movilizar.	272
Figura 5.9 Ficha de análisis de la base de datos del Sistema de Gestión de Incidencias.	273
Figura 5.10 Ficha de Diseño de la Base de Datos para la tabla Prioridad.	274
Figura 5.11 Ficha de Diseño de base de datos para la tabla Incidencia.	275
Figura 5.12 Ficha de Diseño de base de datos par a la tabla Material.	280
Figura 5.13 Esquema final de la base de datos consolidada.	282
Figura 5.14 Esquema de la base de datos móvil.	283
Figura 5.15 Ficha de estimación de tiempos de sincronización para la tabla Prioridad.	284
Figura 5.16 Ficha de estimación de tiempos de sincronización para la tabla Incidencia.	284
Figura 5.17 Ficha de estimación de tiempos de invocación de servicios para el servicio Worker- ManagerWS.	284
Figura 5.18 Ficha de estimación de tiempos de invocación de servicios para el servicio Incident- ManagerWS.	285

Figura 5.19 Fragmento de la política de la aplicación de O&M.....	285
Figura 6.1 Plataforma de experimentación.....	288
Figura 6.2 Proceso incremental de implementación y prueba en MOSE.....	291
Figura 6.3 Distribución de los módulos sobre la plataforma de experimentación.....	291
Figura 6.4 Arquitectura de SQL Server 2005.....	292
Figura 6.5 Arquitectura de SQL Anywhere.....	292
Figura 6.6 Esquema de base de datos de la aplicación Ddetective.....	296
Figura 6.7 Arquitectura interna de la aplicación Ddetective.....	297
Figura 6.8 Plataforma de medición.....	299
Figura 6.9 Foto de la plataforma de medición.....	300
Figura 6.10 Foto del circuito diseñado.....	301
Figura 6.11 Arquitectura de MOSE en el cliente junto con su relación con los RO y RA.....	302
Figura 6.12 Potencia consumida en cada estado de la interfaz WLAN.....	305
Figura 6.13 Línea de ejecución del escenario 1.....	307
Figura 6.14 Fichero de configuración del escenario 1.....	307
Figura 6.15 Línea de ejecución real del escenario 1 – prueba A.....	308
Figura 6.16 Consumo de energía del escenario 1 – prueba A.....	308
Figura 6.17 Línea de ejecución real del escenario 1 – prueba B.....	309
Figura 6.18 Consumo de energía del escenario 1 – prueba B.....	309
Figura 6.19 Línea de ejecución real del escenario 1 – prueba C.....	310
Figura 6.20 Consumo de energía del escenario 1 – prueba C.....	310
Figura 6.21 Línea de ejecución del escenario 2.....	313
Figura 6.22 Fichero de configuración del escenario 2.....	314
Figura 6.23 Línea de ejecución real del escenario 2 – prueba A.....	314
Figura 6.24 Consumo de energía del escenario 2 – prueba A.....	315
Figura 6.25 Línea de ejecución real del escenario 2 – prueba B.....	315
Figura 6.26 Consumo de energía del escenario 2 – prueba B.....	316
Figura 6.27 Línea de ejecución real del escenario 2 – prueba C.....	316
Figura 6.28 Consumo de energía del escenario 2 – prueba C.....	317
Figura 6.29 Línea de ejecución del escenario 3.....	319
Figura 6.30 Fichero de configuración del escenario 3.....	320
Figura 6.31 Línea de ejecución real del escenario 3 – prueba A.....	320
Figura 6.32 Consumo de energía del escenario 3 – prueba A.....	321
Figura 6.33 Línea de ejecución real del escenario 3 – prueba B.....	321

Figura 6.34 Consumo de energía del escenario 3 – prueba B.	322
Figura 6.35 Línea de ejecución real del escenario 3 – prueba C.	322
Figura 6.36 Consumo de energía del escenario 3 – prueba C.	323
Figura 6.37 Línea de ejecución real del escenario 4.	325
Figura 6.38 Fichero de configuración del escenario 4.	326
Figura 6.39 Línea de ejecución real del escenario 4 – prueba A.	327
Figura 6.40 Consumo de energía del escenario 4 – prueba A.	327
Figura 6.41 Línea de ejecución real del escenario 4 – prueba B.	328
Figura 6.42 Consumo de energía del escenario 4 – prueba B.	328
Figura 6.43 Línea de ejecución del escenario 5.	331
Figura 6.44 Fichero de configuración del escenario 5.	331
Figura 6.45 Línea de ejecución real del escenario 5 – prueba A.	332
Figura 6.46 Consumo de energía del escenario 5 – prueba A.	332
Figura 6.47 Línea de ejecución real del escenario 5 – prueba B.	333
Figura 6.48 Consumo de energía del escenario 5 – prueba B.	333
Figura 6.49 Línea de ejecución real del escenario 5 – prueba C.	334
Figura 6.50 Consumo de energía del escenario 5 – prueba C.	334
Figura 6.51 Resumen de la potencia media consumida en cada escenario.	337
Figura 6.52 Comparativa del porcentaje de ahorro obtenido en las pruebas realizadas.	338
Figura 7.1 Arquitectura de MOSE en el cliente junto con su relación con los RO y RA.	340

Índice de Tablas

Tabla 1.1 Distribución de los puntos de acceso comerciales y privados por ciudad [IWE-X06].	2
Tabla 1.2 Objetivos operacionales más relevantes y objetivo específico al que contribuyen.	17
Tabla 1.3 Relación entre los requisitos y los objetivos operacionales.	18
Tabla 2.1 Comparativa de baterías recargables [Israelsohn01].	69
Tabla 2.2 Comparativa de consumo de dos discos duros.	73
Tabla 2.3 Consumo de energía de las interfaces de usuario.	76
Tabla 2.4 Caracterización del consumo de energía de Bluetooth y 802.11b [Simunic+05].	83
Tabla 2.5 Tiempo de vida de la batería ante diversas configuraciones.	89
Tabla 2.6 Análisis de la solución de Mohapatra sobre la gestión de las comunicaciones.	101
Tabla 2.7 Análisis de la solución de SMTP sobre la gestión de las comunicaciones.	101
Tabla 2.8 Análisis de la solución de Poellabauer sobre la gestión de las comunicaciones.	102
Tabla 2.9 Análisis de la solución de PAWP sobre la gestión de las comunicaciones.	103
Tabla 2.10 Análisis de la solución de Twin Agents sobre la gestión de las comunicaciones.	104
Tabla 3.1 Características generales de Puppeteer y Odyssey/Spectra.	111
Tabla 3.2 Características principales de ECOSystem y del framework de Mohapatra.	115
Tabla 3.3 Características generales de Pagoda y Grace.	119
Tabla 3.4 Características principales de PARM y STPM I.	122
Tabla 3.5 Características principales de STPM II y Cooperative I/O.	125
Tabla 3.6 Características principales de Poellabauer y PAWP.	128
Tabla 3.7 Características principales de DSM y Thor.	132
Tabla 3.8 Características principales de FarGo-DA y Twin Agents.	134
Tabla 3.9 Características principales de la solución propuesta por Efstraïou y Replets System.	137
Tabla 3.10 Características principales de WebPADS e IBM.	139
Tabla 3.11 Características principales de la plataforma Domint e IMMS.	141

Tabla 3.12 Características principales de SodaSync.....	142
Tabla 3.13 Características de la solución propuesta en esta tesis.....	148
Tabla 3.14 Comparación de las soluciones existentes frente a la solución propuesta.....	148
Tabla 4.1 Clasificación de las invocaciones a servicios.....	176
Tabla 4.2 Cálculo de los tiempos de las invocaciones en función de sus tipos.....	181
Tabla 4.3 Interfaces de ADO.NET.....	218
Tabla 5.1 Posibles situaciones anómalas de la sincronización.....	259
Tabla 5.2 Relación entre la naturaleza de sincronización y los atributos del lenguaje.....	267
Tabla 6.1 Especificaciones técnicas de la PDA.....	288
Tabla 6.2 Especificaciones técnicas de los servidores.....	289
Tabla 6.3 Descripción del servicio de mensajería.....	296
Tabla 6.4 Descripción del servicio de Blogger.....	297
Tabla 6.5 Parámetros evaluados en el escenario 1.....	306
Tabla 6.6 Resultados obtenidos sobre consumo de energía del escenario 1.....	312
Tabla 6.7 Parámetros evaluados en el escenario 2.....	313
Tabla 6.8 Resultados obtenidos sobre consumo de energía del escenario 2.....	318
Tabla 6.9 Parámetros evaluados en el escenario 3.....	319
Tabla 6.10 Resultados obtenidos sobre consumo de energía del escenario 3.....	324
Tabla 6.11 Parámetros evaluados en el escenario 4.....	325
Tabla 6.12 Resultados obtenidos sobre consumo de energía del escenario 4.....	330
Tabla 6.13 Parámetros evaluados en el escenario 5.....	330
Tabla 6.14 Resultados obtenidos sobre consumo de energía del escenario 5.....	335
Tabla 6.15 Parámetros evaluados en los escenarios.....	337
Tabla 7.1 Objetivos operacionales más relevantes y el apartado donde se abordan.....	340
Tabla 7.2 Relación entre los objetivos específicos y los operacionales.....	342
Tabla 7.3 Comparación de las soluciones existentes frente a la solución propuesta.....	342

- **Ancho de banda**

El ancho de banda de las conexiones inalámbricas puede diferir dramáticamente dependiendo del protocolo inalámbrico usado. Como mostramos en la Figura 1.2, algunas de las redes inalámbricas tienen tasas de transferencia de datos muy bajas, alrededor de 9,6 kbps, lo cual hace que sea inviable el uso de ciertas aplicaciones móviles. Por ello, es necesario optimizar las comunicaciones de forma que se pueda ahorrar tráfico.

- **Latencia**

Algunas veces incluso es más importante la latencia de la red que el propio ancho de banda. La latencia es definida como el tiempo que tarda una petición en alcanzar el servidor. Muchos usuarios culpan del bajo rendimiento al ancho de banda, cuando en realidad la latencia de la red es la que lleva la mayoría del tiempo de la conexión total. Para llevar a cabo una petición a un servidor empresarial, la latencia puede añadir entre 1 y 30 segundos al tiempo de conexión total. La cantidad de tráfico afecta a la latencia con lo que en ciertos periodos de gran cantidad de tráfico, la latencia puede ser muy grande.

- **Fiabilidad**

Las desconexiones son comunes en las redes inalámbricas. Este aspecto debe ser tomado en cuenta a la hora de desarrollar aplicaciones móviles y soluciones de movilidad. La fiabilidad está relacionada con la cobertura y la penetración de la red usada.

- **Coste**

El coste de la conexión inalámbrica puede ser algunas veces prohibitivo. En algunas redes, el coste se calcula en base al tiempo de conexión, mientras que en otras se calcula basándose en la cantidad de datos transmitidos.

- **Falta de unificación de estándares**

Cada proveedor de red mantiene diferentes protocolos de red para conectarse a los dispositivos móviles. Algunos de ellos se basan en IP, mientras que otros no. Este aspecto es más crítico en Norteamérica y Asia donde los proveedores de red implementan varios protocolos distintos. En Europa, hay más consenso en los protocolos de red, lo cual reduce la complejidad.

- **Seguridad**

La seguridad en las redes inalámbricas presenta una problemática especial debido a varios factores [Gaptel03]. Por un lado, el usuario no puede estar seguro de que el punto de acceso al que se conecta pertenezca realmente a la organización que opera el 'hot-spot'⁶. Podría tratarse de un punto de acceso falso instalado en las proximidades con propósito fraudulento. Y por otro, al utilizarse la transmisión por radio, la comunicación puede estar siendo escuchada por terceros que dispongan de un receptor adecuado.

Esto hace que la seguridad sea un elemento muy importante, que hoy por hoy no está bien resuelto en muchas redes.

Las redes cableadas y las redes inalámbricas tienen los mismos objetivos de seguridad (confidencialidad, integridad, control de acceso y disponibilidad) y tienen que hacer frente a los mismos ataques a alto nivel (Denial of service, eavesdropping, man-in-the-middle,...). Los ataques entre redes inalámbricas de área local, típicamente, indican que el atacante accede al enlace entre dos estaciones o una estación y un punto de acceso. La principal diferencia entre proteger una estación LAN cableada y una inalámbrica es la facilidad de interceptar e inyectar comunicaciones de red. En general, desarrollar un entorno seguro inalámbrico es más complicado que desarrollar un entorno seguro cableado debido principalmente, a las limitaciones de los propios dispositivos móviles que no poseen la capacidad suficiente para ejecutar algunos sistemas de seguridad, por ejemplo, algoritmos de cifrado complejos.

Los peligros en las redes inalámbricas son igual a la suma de los peligros en redes cableadas más los nuevos peligros introducidos por la debilidad de los protocolos inalámbricos. A continuación, enumeramos algunos de los ataques y vulnerabilidades de los sistemas inalámbricos [Karygiannis+02]:

- Entidades maliciosas pueden obtener acceso no autorizado a ordenadores de empresas a través de conexiones inalámbricas, saltándose las protecciones de los firewalls.
- La información sensible que no es cifrada y que es transmitida entre dos dispositivos inalámbricos puede ser interceptada y leída.

⁶ "Los hotspots son lugares públicos (hoteles, aeropuertos, estaciones, restaurantes, etc.) donde los usuarios pueden conectarse utilizando la tecnología WiFi" [Carbonell+03] (pp. 103).

"El modelo de hot-spots se corresponde con la creación de redes de comunicaciones electrónicas inalámbricas para la prestación de servicios, fundamentalmente acceso a Internet, en ubicaciones específicas donde se concentra un gran número de potenciales clientes, en lugares de tránsito o vía pública." [Gaptel03](pp. 22).

realizar un estudio sobre las necesidades de movilidad empresariales. Al final de esta memoria se incluye un CD con material adicional entre el que se encuentra una copia de la presente memoria, el código de la solución de movilidad y de los prototipos e información sobre las pruebas realizadas.

Capítulo 2

Soluciones de movilidad y Gestión de recursos

La Computación móvil y, particularmente, el desarrollo de una solución de movilidad que dé soporte a las aplicaciones móviles empresariales junto con la gestión inteligente de los recursos limitados de los dispositivos donde residen, son las áreas de conocimiento en las que se ubica el trabajo de investigación descrito en esta memoria.

El propósito de este capítulo es precisamente proveer al lector de una visión detallada del estado del arte relativo a estas dos áreas. Sin llegar a existir una relación inmediata y directa con las motivaciones, desarrollo y resultados de la solución propuesta, el contenido del presente capítulo sí fija los pilares conceptuales del mismo.

Inicialmente, abordamos los requisitos que toda solución de movilidad debería cumplir para solventar las limitaciones de la computación móvil debidas a los dispositivos móviles, a la naturaleza de las comunicaciones inalámbricas y a la movilidad, detallados en el capítulo uno. De cada requisito concretaremos su definición y analizaremos las distintas técnicas usadas para su cumplimiento.

2.1 Introducción

En muchas ocasiones en el contexto de movilidad se plantea la necesidad de construir aplicaciones en las que su correcto funcionamiento no dependa de una conexión permanente (*online*) con los sistemas centrales que contienen la lógica de negocio y los datos. Esta necesidad surge por distintos motivos:

- El nivel de disponibilidad de las redes móviles, el cual está muy por debajo del nivel de las redes fijas. Todavía se está muy lejos del sueño de disponer de una conexión móvil en todo momento y lugar.
- La existencia de posibles zonas de sombra en la cobertura.
- El bajo ancho de banda y latencia de las redes móviles, lo que hace que las salidas al servidor sean costosas en tiempo.
- La necesidad de optimizar las comunicaciones de forma que se pueda ahorrar tráfico y consumo de energía.
- La existencia de zonas de prohibición de uso de comunicaciones inalámbricas (gasolineras, alrededores de edificios oficiales...).

Las razones anteriores implican la necesidad de disponer aplicaciones móviles que funcionen en modo *offline*, esto es, aplicaciones en las que se garantice el funcionamiento mientras se disponga del terminal móvil, independientemente de que, en un momento dado, no exista conexión al sistema central.

2.2 Tipos de aplicaciones móviles

Giguère [Giguère01] clasificó las aplicaciones según su modelo de conectividad a los datos definiendo dos tipos de aplicaciones:

- **Siempre conectadas (*always connected*)**

Este tipo de aplicaciones requieren de conexión permanente con el servidor central. En caso de no haber conexión, el error sería fatal y la aplicación no podría funcionar. Este modelo es usado normalmente en ordenadores que están conectados permanentemente a una red de área local.

- **Conectadas ocasionalmente (*occasionally connected*)**

En este caso, la aplicación que reside en el dispositivo móvil no siempre podrá conectarse con el servidor. Como hemos comentado, existen ciertas situaciones en las que habrá ausencia de conectividad, por ejemplo, en aeropuertos, o también cuando el coste de transmisión sea demasiado alto o no se disponga de suficiente ancho de banda. A este tipo de aplicaciones también se les denomina *aplicaciones móviles, aplicaciones offline o disconnected applications*. Este tipo de aplicaciones serán las tratadas en esta tesis.

Dentro de las aplicaciones móviles podemos destacar los siguientes tipos, ordenados de mayor a menor con respecto al modo de trabajo offline:

a) Aplicaciones offline puras

Este tipo de aplicaciones se caracterizan porque para realizar la funcionalidad que aportan no realizan ningún tipo de conexión con el servidor. Las únicas conexiones con el servidor se realizan para enviar la información modificada en el terminal móvil hacia el servidor central y para descargar del servidor central nuevos datos. Este proceso recibe el nombre de *sincronización* (ver apartado 2.5.1.3). En estas aplicaciones, la sincronización se produce habitualmente al comienzo y al final del día en puestos especiales conectados a una red fija. Estas aplicaciones pueden funcionar con terminales móviles que no dispongan de conectividad inalámbrica.

b) Aplicaciones offline con sincronización eventual

Se trata de aplicaciones que, aun siendo offline, ante determinados eventos solicitan una sincronización con el servidor central. Los eventos que desencadenen una sincronización serán aquellos que impliquen operaciones de negocio cuya ejecución o resultado deba ser conocida inmediatamente por el servidor central. También podría darse el caso de aplicaciones que se sincronicen periódicamente o en instantes de tiempo concretos con el servidor central. Dado que la sincronización puede desencadenarse eventualmente y en campo, para ejecutar este tipo de aplicaciones será necesario disponer de un terminal con conectividad inalámbrica.

c) Aplicaciones mixtas offline/online

Son aplicaciones para las cuales determinadas operaciones de negocio se resuelven de manera offline, mientras que otras operaciones de negocio se resuelven online, es decir, con conexión directa vía red móvil con el servidor central. Las operaciones de negocio a resolver online serán aquellas que requieran una respuesta inmediata por parte del sistema central.

Un ejemplo claro es una aplicación para instaladores de servicios de telefonía básica. Una vez realizada la instalación del servicio, el operario tendrá que solicitar al sistema central un test de línea para verificar el correcto funcionamiento y/o el diagnóstico de fallos. La aplicación de gestión de la actividad del operario puede funcionar en modo offline, pero la solicitud de la prueba deberá realizarse online, puesto que no se puede resolver en local y tampoco tiene sentido enviarla en diferido, ya que lo que se requiere es un resultado en el menor tiempo posible.

d) Aplicaciones inteligentes offline/online

En este caso se trata de aplicaciones cuyo modo de funcionamiento normal es online. Sin embargo, están preparadas para conmutar a un modo de funcionamiento offline en caso de que se detecte la pérdida cobertura o cualquier problema de red. Cuando la capacidad de conexión a la red se recupera, la aplicación es capaz de sincronizar la información con el servidor central, de forma transparente al usuario.

e) Aplicaciones online con minimización de interacciones con el servidor

Este tipo de aplicaciones, aun siendo online, están construidas de forma que el número de interacciones con el servidor central se minimice. Son un caso particular de c) donde todas las operaciones de negocio son online, existiendo una sincronización de datos, combinada con un almacén persistente, que permite minimizar el número de conexiones con el servidor central.

Estrictamente hablando, este tipo de aplicaciones no podrían considerarse offline. Sin embargo, las detallamos en este apartado para recalcar el hecho de que es posible diseñar aplicaciones online que minimicen el número de salidas al servidor, mejorando la experiencia del usuario y ahorrando costes de tráfico. Adicionalmente, es posible combinar las estrategias definidas por los casos c), d) y e) para obtener una aplicación inteligente offline/online que además minimiza el número de interacciones con el servidor.

Analizando estos tipos de aplicaciones móviles, debemos destacar que todas ellas siguen el modelo de adaptación libre (ver apartado 2.3), donde es la propia aplicación la responsable de proporcionar soporte a su operación online/offline, realizando este proceso de una manera más o menos óptima pero acorde a sus requisitos. Esto implica que ante cada nueva aplicación móvil, el diseñador deba enfrentarse a esta funcionalidad desde cero. En esta tesis, pretendemos extraer la responsabilidad de realizar esta funcionalidad de las propias aplicaciones, delegándola a un módulo aparte residente en el propio middleware, denominado solución de movilidad, con lo que esta funcionalidad podrá ser utilizada por todas las aplicaciones móviles de manera totalmente transparente y permitiendo llevar a cabo una gestión más inteligente de todas ellas y de los recursos del dispositivo móvil donde residen.

En función de cómo y dónde se realice el soporte a la operación online/offline surgen diversos modelos denominados *modelos de adaptación a la movilidad* que describimos a continuación.

2.3 Modelos de adaptación a la movilidad

La necesidad de adaptación es un elemento intrínseco a la computación móvil [Layne99]. Las aplicaciones movilizadas deberían ser diseñadas para ser capaces de adaptarse de manera dinámica a las condiciones del entorno (cobertura, batería,...). Para cumplir los requisitos de una solución de movilidad es necesario que las aplicaciones se adapten. Las diferentes responsabilidades que conlleva la movilidad pueden ser asignadas a diferentes componentes del sistema, dependiendo de a quien se le asigne esa responsabilidad podemos definir diferentes modelos de adaptación a la movilidad [Kunz+05] [Intel03a] [Edmonds01].

Un modelo de adaptación se describe como “Underlying platform allocate responsibilities for dealing wireless related to mobility” [Intel03a] (pp. 7). Se han definido tres modelos de adaptación que se describen a continuación:

- **Adaptación libre (*laissez-faire*)**

Con este modelo, cada aplicación es totalmente responsable de proporcionar soporte a la movilidad. La adaptación se lleva a cabo sin soporte por parte del sistema. El sistema no realiza un control centralizado de los recursos, por lo que cada aplicación debe competir con otras por el uso de los mismos. Como ejemplo de este tipo de aplicaciones destacamos Rover [Joseph+95].

- **Adaptation transparente a la aplicación (*application-transparent adaptation*)**

Con este modelo la solución de movilidad aísla a la aplicación de los aspectos relativos a la movilidad. La característica más peculiar de este modelo es que la aplicación no es modificada y que no le afectan los cambios externos. Este modelo es deseable pero sólo es válido para aplicaciones sencillas. Se caracteriza porque el sistema realiza un control centralizado de los recursos. El principal problema es que el sistema debe operar sin información específica de las necesidades de las aplicaciones que le ayude a tomar las decisiones más adecuadas sobre el uso de los recursos. Como ejemplo podemos citar las siguientes soluciones de movilidad, que en el capítulo tres describiremos más en detalle: Puppeteer [Lara+05], ECOSystem [Zeng+05], Pagoda [Huang+05], PARM [Mohapatra+03] y PAWP [Rosu+04].

- **Adaptación sensible a la aplicación (*application-aware adaptation*)**

Este modelo recibe también el nombre de *Integrated Adaptation* [Edmonds01]. Con este modelo, la solución proporciona servicios comunes que las aplicaciones pueden usar para dar soporte a la movilidad. El rango de servicios ofrecidos por la solución va desde proporcionar información simple hasta el envío de notificaciones a las aplicaciones. En

versiones más avanzadas, las aplicaciones pueden conocer a otras y cooperar con ellas en el uso de los recursos de movilidad. Este modelo integra los otros dos modelos ya que combina la gestión de los recursos por parte de la solución junto con el conocimiento de las necesidades de las aplicaciones. El principal inconveniente de este modelo es su complejidad ya que requiere de una alta cooperación entre las aplicaciones y la solución. Además, otra característica de este modelo es que las aplicaciones deben ser modificadas para hacer uso de las facilidades de movilidad que les proporciona la solución. Como ejemplo podemos citar las siguientes soluciones de movilidad, que describiremos más en detalle en el capítulo tres: Odyssey/Spectra [Flinn+01a], la definida por Mohapatra et al. [VanAntwerpen+04], Grace [Yuan+06], STPM [Anand+05] y Cooperative I/O [Weissel+02].

En esta tesis hemos definido una subclasificación dentro de este modelo: por un lado, se encuentran las *aplicaciones con soporte a la adaptación* y, por otro, *las aplicaciones sin soporte a la adaptación*. La primera de ellas hace referencia a aquellas aplicaciones que implementan ellas mismas mecanismos de adaptación que pueden ser invocados por parte de la solución de movilidad. Las segundas son aquellas que no implementan mecanismos de adaptación aunque pueden invocar a un módulo externo que implementan estas adaptaciones. Existen diversas ventajas de la segunda aproximación frente a la primera como su flexibilidad, ya que permite añadir o modificar los mecanismos de adaptación después de que la aplicación haya sido implementada. Además, en la primera aproximación, el diseñador debe definir todos los posibles mecanismos de adaptación en tiempo de diseño.

De los tres modelos de adaptación esta tesis se centra en el último, es decir, el sensible a la aplicación. La solución desarrollada en esta tesis proporcionará servicios para la movilidad que podrán usar las aplicaciones, además buscamos que las aplicaciones indiquen sus necesidades a la solución, de esta manera se podrán tomar las decisiones más óptimas en cada momento, con respecto a las propias aplicaciones y a la gestión de los recursos que utilizan. Además, esta solución también dará soporte a las aplicaciones del segundo modelo, es decir, el transparente a la aplicación, para dar soporte a la gestión inteligente de aquellas aplicaciones que no quieran aportar sus necesidades concretas.

Otro aspecto a tener en cuenta es **el nivel donde se realiza la adaptación**. La mayoría de los investigadores definen tres niveles de adaptación [Abukmail+04], [Martin+04], [Sharma+04]:

- Nivel Físico o hardware.
- Nivel del Sistema operativo.
- Nivel de Aplicación.

la **gestión inteligente de las comunicaciones**, al considerarlos los que más repercusión tienen en este tipo de aplicaciones, tal y como justificamos en este capítulo.

A continuación, desarrollaremos el estado del arte de cada uno de estos requisitos.

2.5 Operación online/offline

Como hemos visto en el capítulo anterior, los usuarios pueden perder temporalmente su conectividad inalámbrica. Incluso si los usuarios tienen una buena conectividad de red, las aplicaciones pueden no tener acceso a los recursos de la red cuando ellas desean, por ejemplo, un servicio podría estar ocupado, caído o temporalmente no disponible.

Una aplicación es *occasionally connected* si a veces no puede interactuar con los servicios o los datos que se encuentran en la red. De esta manera, los usuarios no podrían trabajar cuando están en offline. El permitir a los usuarios trabajar cuando se está en offline, aumenta su productividad y reporta varios beneficios tal como se detalló en el capítulo anterior.

Las aplicaciones móviles deben ser capaces de seguir trabajando cuando no hay conexión de red y actualizar los recursos cuando ésta esté disponible. Para permitir que una aplicación pueda seguir trabajando en offline, es necesario proporcionar una infraestructura, denominada genéricamente solución de movilidad. Esta infraestructura podría incluir diversas funcionalidades, por ejemplo, cacheo de datos locales con todos los datos requeridos por el cliente, los cuales deberán ser sincronizados cuando se disponga de red. Las capacidades y características que la infraestructura debe proporcionar a las aplicaciones móviles dependerán de la conectividad, del entorno y de la funcionalidad que el usuario espera cuando se está en online y en offline. En cambio, toda aplicación móvil debería proporcionar una funcionalidad mínima a estos usuarios en offline.

Las soluciones de movilidad deberían ser diseñadas de manera óptima para maximizar el uso de la red cuando está disponible, asegurando que los datos sean actualizados, sin afectar al rendimiento de las aplicaciones.

Esta tesis se centra en garantizar una operación online/offline a dos niveles: a nivel de los datos y a nivel de la invocación de servicios. Al primero le hemos denominado en esta tesis **enfoque orientado a datos** y, al segundo, **enfoque orientado a servicios** (ambos son definidos en el apartado 4.3.1.2). En ambos enfoques deberemos tener en cuenta los recursos disponibles y las necesidades de las aplicaciones.

- **Enfoque orientado a datos.**

La principal característica que tiene este enfoque es que debe disponer de un sistema de almacenamiento tanto en el cliente como en el servidor. En el momento en el que el cliente pase a un estado offline, el usuario puede trabajar con sus datos sin percibir que la comunicación entre ambos se ha perdido. Por lo tanto, en este modelo es necesaria una sincronización entre ambos sistemas de almacenamiento para detectar y modificar los cambios producidos durante la fase de desconexión. En el apartado 2.5.1 detallaremos este enfoque analizando las distintas técnicas usadas.

- **Enfoque orientado a servicios.**

Este enfoque permite que el cliente pueda interactuar con servicios residentes en otras máquinas, sin la necesidad de un sistema de almacenamiento en el dispositivo móvil. Esto se debe a que presenta características diferentes al orientado a datos, porque no se centra en la sincronización de los mismos, sino en la realización de tareas o funciones mediante invocaciones a servicios. En el momento en el que el cliente cambia al estado offline, las sucesivas invocaciones son encoladas y una vez recobrada la conexión, son enviadas al otro destino. En el apartado 2.5.2 detallaremos este enfoque analizando las distintas técnicas usadas.

2.5.1 Enfoque orientado a datos

En muchas arquitecturas de aplicaciones, los datos se almacenan en sistemas de red compartidos en forma de documentos, base de datos o ficheros, con un acceso por parte del usuario combinado a una única copia de datos. Una aplicación diseñada para la movilidad debería permitir a los usuarios el acceso a los datos tanto en online como en offline. Cuando se opere en offline el usuario debería percibir que los datos compartidos están disponibles para leer y actualizar. Cuando la conectividad de red es restaurada, los cambios en los datos locales deberían ser integrados en la copia de los datos que se encuentran en red y viceversa.

Actualmente, la mayoría de las aplicaciones hace uso de base de datos para almacenar sus datos. Las aplicaciones móviles para poder operar en offline deben disponer de los datos en el dispositivo móvil. Estos datos se almacenan también en una base de datos pequeña que recibe el nombre de **base de datos móvil**. Los datos almacenados en la base de datos móvil serán copias (también denominados réplicas) de los datos almacenados en el servidor empresarial. A la técnica que se encarga de la gestión de estas réplicas o copias se denomina **replicación**. Dentro de ésta, existe una tarea esencial para mantener actualizadas las réplicas, ésta es la **sincronización de datos**. En este apartado, explicaremos los conceptos fundamentales, ya que esta área será una parte esencial de esta tesis.

El concepto de base de datos móvil está íntimamente relacionado con las características y principios de las bases de datos distribuidas como la replicación, distribución o fragmentación de los datos, pero añaden una característica fundamental en estos casos como es la movilidad. Una base de datos móvil se define como “una base de datos portable y físicamente independiente del servidor corporativo de base de datos, pero es capaz de comunicarse con ese servidor desde sitios remotos permitiéndole compartir los datos corporativos” [Conolly+05].

2.5.1.1 Base de datos móviles

Desde el punto de vista de la gestión de datos, la base de datos móvil puede considerarse como una variante de las bases de datos distribuidas⁹. Por lo tanto, los aspectos sobre la gestión bases de datos distribuidas [Date01] pueden aplicarse a las bases de datos móviles teniendo en cuenta una serie de consideraciones y variantes adicionales [EPFL+02].

Las bases de datos móviles pueden trabajar en diferentes modos de conexión (siempre conectado y parcialmente conectado) [Schneider+04]. Esta tesis se centrará en las más complejas, las parcialmente conectadas. Las bases de datos usadas por este tipo de aplicaciones reciben el nombre de base de datos sincronizadas intermitentemente (ISDB [Intermittently Synchronized Database]) [Yee+05], [Jermaine+06].

Las siguientes características de las ISDB las diferencian de las bases de datos móviles que hemos definido hasta ahora en que:

- Un cliente se conecta cuando desea recibir actualizaciones de un servidor, enviar sus actualizaciones o procesar transacciones que precisan datos no locales. Esta comunicación puede ser unidestinataria (una comunicación uno a uno entre el servidor y el cliente) o multidestinataria (un emisor o servidor puede comunicarse periódicamente con un conjunto de ordenadores o actualizar un grupo de clientes).
- Un servidor no puede conectarse con el cliente cuando lo desea.
- Un cliente es libre de gestionar sus propios datos y transacciones mientras esté desconectado. También puede realizar su propia recuperación de datos hasta cierto punto.

⁹ Se puede definir la base de datos distribuida “como una colección de múltiples bases de datos interrelacionadas lógicamente y distribuidas por una red de computadores. Mientras que un sistema de gestión de bases de datos distribuidos (SGBDD) como un sistema software que maneja una base de datos distribuida haciendo la distribución transparente para el usuario.” [Elmasri+02] (pp.728).

- Un cliente tiene múltiples formas de conectarse a un servidor y, en el caso de muchos servidores, éste puede elegir un servidor concreto al que conectarse de acuerdo a su proximidad, nodos de comunicación disponibles, etc.

Debido a estas diferencias, hay una necesidad de abordar una serie de problemas relacionados con las ISDB que difieren de los que atañen generalmente a los sistemas de bases de datos móviles. Entre éstos se incluyen:

- La gestión de la coherencia entre el procesamiento de las transacciones.
- El diseño de la base de datos del servidor y de los clientes móviles.
- Gestión inteligente de las conexiones.
- Gestión inteligente de los recursos de los dispositivos móviles principalmente, de la batería.
- Gestión inteligente de la sincronización.

Los cuatro últimos serán los problemas abordados en esta tesis, buscando diseñar una solución de movilidad que optimice el uso de los recursos del dispositivo por parte de las aplicaciones móviles basadas en base de datos móviles.

2.5.1.2 Estrategias de diseño de base de datos móviles

Replicación

En los apartados anteriores hemos visto que el uso de las bases de datos móviles proporcionan una solución al trabajo offline al permitir tener acceso a los datos aún no teniendo conexión. En el uso de base de datos móviles, el primer paso a realizar es decidir qué datos residirán en la base de datos del dispositivo móvil y, posteriormente, cómo gestionar las actualizaciones de esos datos. El proceso que se encarga de la gestión de estos datos se denomina “replicación”.

Se define replicación de datos como “el mantenimiento de copias de datos en múltiples computadores” [Coulouris+01]. La replicación es la clave para la eficacia de sistemas distribuidos en la medida en que proporciona un mayor rendimiento, alta disponibilidad y tolerancia a fallos. La replicación se utiliza ampliamente, por ejemplo, el almacenamiento en memorias caché de recursos procedentes de servidores web en los navegadores y en servidores proxy de la Web es una forma de replicación, dado que los datos que se almacenan en las memorias caché y en los servidores son réplicas de otros.

Un sistema puede ser replicado teniendo múltiples copias: si una falla las otras continúan. El principal problema es que se deben mantener todas las copias sincronizadas y ocultar los fallos. Idealmente, los fallos deben ser gestionados de manera transparente a la aplicación [Wiesmann02]. Esta característica es ideal para las aplicaciones móviles [Ratner+01].

La replicación [Wiesmann+00] es una técnica para la mejora de los servicios. Las motivaciones para la replicación son aumentar el rendimiento de un servicio e incrementar su disponibilidad o hacerlo tolerante a fallos.

- **Mejora del rendimiento:** el almacenamiento de datos local en los clientes y servidores es hoy en día un medio conocido para mejorar el rendimiento. La replicación de datos inmutables es trivial: se incrementa el rendimiento con un bajo coste para el sistema. La replicación de datos cambiantes, tales como los de la Web, ocasiona sobrecargas en forma de protocolos diseñados para asegurar que los clientes reciben datos actualizados. Por lo tanto, hay límites en la efectividad en la replicación como técnica para la mejora del rendimiento.
- **Incremento en la disponibilidad:** los usuarios requieren que los servicios tengan una alta disponibilidad. Esto es, la proporción de tiempo en la que un servicio es accesible con tiempos de respuesta razonables debería estar cercana al cien por cien. Dejando a un lado retrasos debidos a los casos pesimistas en los conflictos de control de concurrencia (bloqueo de datos), los factores relevantes para una elevada disponibilidad son:
 - ▷ Fallos en el servidor.
 - ▷ Particiones de la red y operación sin conexión: desconexiones en la comunicación usualmente no planificadas y que son un efecto colateral de la movilidad del usuario.

Empezando por el primero, la replicación es una técnica para el mantenimiento automático de la disponibilidad de los datos a pesar de fallos en el servidor. Si los datos se replican en dos o más servidores independientes ante fallos, entonces el software del cliente podría ser capaz de acceder a los datos de un servidor alternativo cuando el servidor por defecto falle o no esté accesible.

Las particiones de la red y las operaciones sin conexión son el segundo factor en contra de la alta disponibilidad. Los usuarios móviles desconectan sus computadores deliberadamente o pueden inintencionadamente encontrarse sin conexión de una red inalámbrica mientras se mueven. Para poder trabajar en estas circunstancias, el usuario se suele preparar copiando

los datos frecuentemente usados, tales como los contenidos de una agenda compartida, desde su entorno habitual al portátil. Pero a menudo debe pagarse un precio por la disponibilidad durante el período de desconexión; cuando el usuario consulta o actualiza la agenda, se arriesga a leer datos que, entre tanto, otro puede haber alterado. Por ejemplo, pueden concertar una cita en un hueco que ya ha sido ocupado. El funcionamiento sin conexión sólo es factible si el usuario (o la aplicación, en su nombre) puede arreglárselas con datos anticuados y resolver más tarde cualquier conflicto que surja.

- **Tolerancia a fallos:** datos con alta disponibilidad no equivale necesariamente a datos estrictamente correctos. Pueden no estar actualizados, por ejemplo; dos usuarios en lados opuestos de una partición de la red podrían hacer actualizaciones que entrarán en conflicto y que necesiten ser resueltas. Por el contrario, un servicio tolerante a fallos siempre garantiza un comportamiento estrictamente correcto a pesar de un cierto número y tipo de fallos. La corrección alude a la frescura de los datos. Algunas veces, la corrección también se refiere a la oportunidad de las respuestas del servicio, como, por ejemplo, en el caso de un sistema de control de tráfico aéreo, donde se necesitan datos correctos cada poco tiempo.

La replicación se usa tanto en la base de datos como en los sistemas distribuidos. En los sistemas distribuidos se persigue que incremente la disponibilidad. En la base de datos, en cambio, la replicación se usa principalmente con propósitos de rendimiento y tolerancia a fallos [Wiesmann+01].

Fragmentación

La fragmentación es una tecnología clave para grandes tablas ya que permite descomponerlas en piezas más pequeñas y manejables llamadas fragmentos [Oracle05b]. En la computación móvil, la base de datos en el servidor contendrá más información de la requerida por la aplicación móvil. Por esta razón, la base de datos móvil sólo debería contener un subconjunto de los datos de la base de datos empresarial. Esta capacidad permite obtener varios beneficios entre ellos, la base de datos móvil sólo incluye una porción de los datos relevantes para el usuario con lo que se podrá reducir el número de conflictos con otros usuarios. De esta manera se pueden eliminar aquellas tablas que no va a usar, reduciendo la sobrecarga. Otro beneficio es que se minimiza la cantidad de datos a ser sincronizados.

La fragmentación puede ser de tres formas:

- **Fragmentación horizontal:** La fragmentación horizontal divide una relación “horizontalmente” agrupando filas para crear subconjuntos de tuplas, donde cada subconjunto tiene un cierto significado lógico. Estos fragmentos pueden entonces asignarse a diferentes sitios en el sistema distribuido. La fracción horizontal derivada aplica la partición de una relación primaria a otras relaciones secundarias, que referencian la primaria a través de una clave externa. De esta forma, los datos relacionados entre las relaciones primarias y secundarias se fragmentan de la misma forma.
- **Fragmentación vertical:** Ningún sitio tiene por qué necesitar todos los atributos de una relación, lo que indica la necesidad de otro tipo diferente de fragmentación. La fragmentación vertical divide una relación “verticalmente” en columnas. Un fragmento vertical de una relación mantiene sólo ciertos atributos de la relación.
- **Fragmentación mixta (híbrida):** Permite entremezclar los dos tipos de fragmentación para obtener una fragmentación mixta.

Un *esquema de fragmentación* de una base de datos es una definición de un conjunto de fragmentos que incluye todos los atributos y tuplas de la base de datos y satisface la condición de que la base de datos completa se puede reconstruir a partir de los fragmentos mediante alguna secuencia de operaciones de join.

Un *esquema de asignación* describe la asignación de fragmentos entre los sitios del SBDD; por tanto, es una correspondencia que especifica el sitio o sitios donde se almacena cada fragmento. Si un fragmento se almacena en más de un sitio, se dice que está replicado.

A modo de ejemplo, Bolchini [Bolchini+04] [Bolchini+04a] ha desarrollado una metodología para el diseño de bases de datos móviles, en ella tiene en cuenta en primer lugar, las peculiaridades del dispositivo, como la capacidad de almacenamiento, en segundo lugar, la movilidad soportada por dicho dispositivo y, por último, el contexto (*context awareness*). Por otro lado, Chan [Chan+03] ha desarrollado una nueva técnica denominada *Database Summarisation* que permite la reducción del tamaño y la capacidad de información de la base de datos mientras maximiza la usabilidad de los datos.

En esta tesis haremos uso de la fragmentación en aras de la búsqueda de la máxima optimización de los recursos.

2.5.1.3 Sincronización

La sincronización de datos es la tecnología clave de la replicación para solventar algunas de las debilidades impuestas por la computación móvil. La sincronización de datos es la

tecnología usada para mantener todas las copias distribuidas en el almacén de datos de manera consistente comunicando los cambios entre esas copias y resolviendo los conflictos debidos a cambios contradictorios en diferentes copias de los mismos datos.

Configuraciones de sincronización o topologías

Los cambios realizados en diferentes copias de los datos pueden ser propagados a otras copias de diferentes modos. La topología de sincronización o configuración define el flujo lógico de propagación de los cambios a través de las diferentes copias de esos mismos datos. Existen cuatro topologías principales:

- **Uno a uno.**

La topología uno a uno es el caso más simple. El resto de topologías se pueden ver como una extensión de ésta. En este caso, los datos son compartidos entre un servidor y un único cliente. Un posible escenario de esta topología es un proceso de backup de los datos, donde todos los cambios hechos por el cliente son enviados al servidor, asegurando que la copia de los datos dispone de la última versión. En este caso no hay peligro de conflicto y a esta topología se la conoce también como “Dedicated Pair”.

- **Uno a N.**

La topología uno a N (incluso conocida como topología en estrella) es la más utilizada en los sistemas comerciales. En esta topología los datos se propagan desde un maestro central a las diferentes entidades que contienen las copias de los datos, a esta topología también se la conoce con el nombre de *Hub-and-spoke configuration* [Wallick03].

Su principal ventaja es la simplicidad de implementar comparada con la topología N a N. En esta topología, todos los clientes intercambian datos con el servidor con lo que los conflictos sólo pueden producirse en este servidor, que necesitará detectarlos y resolverlos. Los clientes no deberán preocuparse por los conflictos. Sólo deberán informar al servidor central de las modificaciones locales. El principal inconveniente de esta topología es que el servidor central puede llegar a convertirse en un cuello de botella y en un único punto de fallo que podría inmovilizar todo el sistema. Por ello, este servidor central no debería ser un único servidor, sino un cluster de servidores que solventen este inconveniente.

- **Topologías híbridas de Uno a N y de N a N.**

Como se muestra en la Figura 2.2, se pueden obtener topologías híbridas combinando las topologías 1-N y N-N, la primera de las topologías híbridas se la denomina *cluster*. El

- “Two-way sync”. Es el modo habitual de sincronización, el cliente y el servidor intercambian información sobre los datos modificados en sus dispositivos. Es una comunicación bidireccional.
- “Slow sync”. El “Slow sync” es una forma de sincronización Two-way en la cual todos los datos en la base de datos cliente son comparados con los del servidor, campo por campo. Es una comunicación bidireccional.
- “One-way sync from Client only”. El cliente envía sus modificaciones al servidor pero el servidor no envía sus modificaciones de vuelta al cliente. Es una comunicación unidireccional.
- “Refresh sync from Client only”. El Cliente exporta todos sus datos desde una base de datos al Servidor. Es una comunicación unidireccional.
- “One-way sync from Server only”. El cliente recoge todas las modificaciones del servidor pero el cliente no manda sus modificaciones al servidor. Es una comunicación unidireccional.
- “Refresh sync from Server only”. El servidor exporta todos sus datos desde una base de datos al cliente. Es una comunicación unidireccional.
- “Server alerted sync”. El servidor informa al cliente de la necesidad de iniciar un tipo específico de sincronización con el servidor.

En esta tesis haremos uso de la técnica “two-way sync”.

Estrategias de sincronización

La estrategia o técnica de sincronización es la definición de las condiciones y mecanismos por los cuales se va a llevar a cabo la sincronización de los datos entre el servidor y la base de datos remota. Para determinar cuál es la más apropiada debemos considerar dos aspectos: la frecuencia en la que los datos son sincronizados y cuántos cambios se producen en los datos entre sincronizaciones. Cuando se usan comunicaciones inalámbricas, es aconsejable reducir la transferencia de datos al mínimo para incrementar la eficiencia y reducir el consumo de energía y los costes (aspectos principales que abordaremos en esta tesis).

De manera natural surgen dos técnicas, la denominada *sincronización basada en timestamp*¹⁰ [Burris04], que podemos resumir en “actualiza lo que tenga una fecha más reciente a los datos que hay en remoto” y otra denominada *sincronización basada en snapshot*, que resumimos en “vuelve a cargar todos los datos”. Sobre estas técnicas básicas de sincronización podemos aplicar a su vez la técnica de *sincronización selectiva*, que apoyándose en alguna de las anteriores, trata de optimizar el proceso de sincronización evitando tener que sincronizar todas las tablas.

La elección de la técnica de sincronización está condicionada por el sistema de sincronización elegido, ya que no ofrecen todas las técnicas. La solución de movilidad propuesta en esta tesis dará soporte a todas ellas. A continuación explicamos más en detalle en que consiste cada una de ellas.

Sincronización basada en timestamp

El método basado en timestamp (también denominado *Net changes* [Mallick03]) es la técnica más empleada en general para llevar a cabo una sincronización de datos eficiente. Esta técnica implica controlar el último instante en el que el usuario sincronizó sus datos. Mediante esta información es posible controlar las filas descargadas en cada base de datos remota para transferir solamente aquellas filas que hayan cambiado desde la última sincronización.

Esta forma de sincronización es adecuada para aplicaciones que se sincronizan varias veces al día, enviando poca cantidad de datos cada vez. Es también una buena técnica para situaciones en las que grandes cantidades de datos son sincronizados con menos frecuencia.

Pero esta técnica tiene una debilidad: si se quiere almacenar información sobre cada transacción individual (inserciones, modificaciones y borrado) realizadas a los datos, esta técnica no es aconsejable ya que sólo guarda el cambio más reciente. En la solución desarrollada en esta tesis deberemos guardar esta información para operar de manera eficiente.

Esta técnica de sincronización bien aplicada resulta una de las más eficientes, ya que se limita el tráfico y el proceso únicamente a los registros que han cambiado [Burris04].

Sincronización basada en snapshot

Como alternativa a la sincronización descrita en el apartado anterior, existe la sincronización basada en snapshot [Mallick03] [Burris04]. Esta técnica implica descargar completamente

¹⁰ Marca de tiempo.

Por ejemplo, dado un sistema que posee tres subscriptores en el que se han definido cinco publicaciones (1, 2, 3, 4 y 5). Los clientes necesitan [1, 2], [1, 2, 3, 5] y [4, 5], respectivamente. Se pueden definir los *datagroups* siguiendo estos modelos como se muestra en la Figura 2.6:

Se ha demostrado en sus experimentos que la opción *Op* trabaja mejor cuando el número de clientes se incrementa. De las tres opciones la que mejor resultado obtiene es la *Op* y la que peor resultado obtiene es la *Cc*. El diseñador deberá definir las publicaciones más adecuadas y optar por una de estas tres opciones.

Gestión de los conflictos

La sincronización de datos es una tecnología clave para la operación offline, pero presenta varios problemas, entre ellos, se encuentran los conflictos. El principal reto que deben soportar los sistemas de sincronización es cómo asegurar que los cambios de los datos realizados por múltiples usuarios sobre el mismo conjunto de datos son propagados correctamente a otros usuarios. Este problema se denomina **colisión de datos** [Hemant+99].

Resolver este problema en tiempo real puede ser relativamente simple o muy complejo dependiendo de dos factores importantes:

- El diseño del esquema de la base de datos para soportar la naturaleza distribuida de la aplicación junto a la aplicación de técnicas como la fragmentación.
- La existencia de software de sincronización que detecte y resuelva cualquier colisión que se produzca.

El primero de los puntos es el correcto diseño del sistema de la base de datos. Este esquema de base de datos debe cumplir las reglas de normalización [Elmasri+02] definidas para las bases de datos. El realizar un buen diseño del esquema de base de datos puede permitir reducir las potenciales colisiones de datos y otros problemas de consistencia, por ejemplo, las relaciones N a N o circulares, por su naturaleza, pueden causar colisiones de datos. Para reducir el número de colisiones e incrementar la efectividad de la tecnología de replicación, lo aconsejable es eliminar cualquiera de estos aspectos del diseño del esquema antes de desarrollar una solución de replicación.

Incluso con un diseño de esquema correcto no se pueden eliminar todas las colisiones de datos. Los siguientes tipos de colisiones de datos, causados por las limitaciones en las comunicaciones debido a la movilidad, deben ser resueltos para mantener la consistencia global de los datos:

- **Colisiones de actualización**

Los registros que son actualizados por dos o más aplicaciones pueden generar conflictos causando que uno sobrescriba a otro en tiempo de sincronización. Las colisiones de actualización pueden ser evitadas utilizando la fragmentación. La fragmentación se puede usar en aquellas situaciones donde diferentes individuos tienen control sobre secciones de datos separadas. En este caso, al fragmentar los datos de este modo, los usuarios pueden realizar cambios de manera arbitraria de los datos sin generar conflictos. Holliday [Holliday+02] plantea varios modos de desconexión para bases de datos móviles fragmentadas, buscando reducir el número de conflictos.

- **Colisiones de inserción**

Los registros generados por aplicaciones locales generan claves públicas [Elmasri+02], que deberán ser únicas cuando lleguen a la base de datos central. Las colisiones por inserción son debidas a la duplicación de la clave primaria, que la base de datos usa para garantizar la unicidad de los registros de base de datos. El principal problema es cómo generar identificadores únicos y asegurar su unicidad cuando se encuentra desconectado.

Hay varios modos de solventar este problema por parte de la propia aplicación y en esta tesis se tienen en cuenta cada uno de ellos. Pero existen otras aproximaciones para solventar este problema dejando la responsabilidad de generar identificadores únicos fuera de la aplicación.

A continuación detallamos las tres soluciones más utilizadas actualmente [Burris04]:

- **Universally Unique Identifiers (UUID)**

Algunas bases de datos soportan un identificador único denominado UUID. Consiste en combinar alguna información que es única en el ordenador en el que opera (como el número de serie de la CPU) junto con una marca de tiempo (*timestamp*). De esa manera, estos valores garantizan que no serán asignados más que una vez.

La ventaja de los UUID es su simplicidad. Si se está diseñando una base de datos para un dispositivo móvil, la desventaja es que su tamaño es muy grande, típicamente 16 bytes.

- Global auto-increment values (GUID)

La segunda aproximación consiste en la partición de los valores de la clave primaria entre las bases de datos móviles. A cada base de datos móvil se le asigna un valor ID único con el comienzo de su partición. Este ID se auto-incrementa ante la inserción de un nuevo elemento de la base de datos. Esta aproximación tiene un mínimo trabajo de configuración y además ocupa menos que una clave primaria UUID.

- Pools de claves primarias

La tercera aproximación permite más flexibilidad que las anteriores. Esta aproximación es adecuada cuando la clave primaria en las bases de datos central tiene un formato específico y se quiere mantenerlo en las bases de datos móviles.

Un pool de claves primarias es un modo de distribuir rangos de claves primarias, de cualquier formato, a bases de datos móviles, donde usarán un valor dentro del rango asignado. Cuando el rango de claves finaliza, el servidor le asigna un nuevo rango.

Una posible implementación de este pool es mediante el uso de una tabla en la base de datos central que contiene la lista de usuarios y el conjunto de valores de clave primaria válidos para dicho usuario.

Los beneficios de este particular sistema son claros: es fácil de usar y la aplicación garantiza la unicidad. En cambio este sistema puede no ser óptimo para aquellas aplicaciones cuyas claves primarias tengan ciertas restricciones de esquema o para aplicaciones que necesiten valores generados en ciertos rangos. Una aproximación podría ser implementar su propio sistema GUID usando *timestamp* u otros valores únicos.

Incluso con el mejor esquema de base de datos y con estas técnicas para evitar colisiones de datos, hay casos en que las colisiones no se pueden evitar con lo que se producirán conflictos de datos: una situación que debe ser resuelta durante la sincronización para evitar la corrupción de los datos. Existen diferentes tipos de conflictos de datos:

- **Conflicto de inserción.**

Una clave generada manualmente, por ejemplo, un email puede dar lugar a que ya exista. Se puede afrontar este problema de dos maneras: permitir el fallo de la inserción o aplicar la lógica de negocio para resolver el conflicto. Esta lógica puede cambiar la inserción o denegar la inserción.

- **Conflicto de actualización.**

También se denomina *Write-Write Conflict* [Hassmann+02]. Al igual que antes, dos actualizaciones del mismo registro desde dos aplicaciones móviles diferentes pueden generar un conflicto. Existen dos opciones para resolver este problema: permitir que la actualización falle o aplicar la lógica de negocio para resolver el conflicto, esta lógica deberá decidir cuál es la actualización válida.

- **Conflictos de actualización frente al borrado.**

El resolver la actualización de un registro que ha sido borrado por otro usuario es un problema clásico de base de datos que es crítico en aplicaciones móviles. Existen varias técnicas para resolver este problema, por ejemplo, permitiendo que el cambio más reciente prevalezca o establecer una regla de “actualización sobre borrado” o “borrado sobre actualización”.

Todos estos conflictos entran dentro de la categoría de conflictos llamados *conflictos mecánicos* [Hassmann+02] ya que suceden cuando hay modificaciones concurrentes sobre un mismo registro, pero existen otro tipo de conflictos que desde el punto de vista de la tecnología de sincronización no son conflictos, pero para un usuario sí que lo son. A estos conflictos se les denomina *conflictos semánticos* o *conflictos de reglas de negocios*. Por ejemplo, que un director introduzca una reunión en su agenda con un cliente de ocho a diez de la mañana mientras que su secretaria inserta la entrada en la agenda con otra reunión de nueve a diez y media. Desde el punto de vista de la sincronización, no hay conflicto pero desde el punto de las reglas de negocio sí lo hay. A la hora de detectar estos conflictos habrá que implementar una cierta lógica en el servidor para que sea capaz de detectarlos y resolverlos.

Detección de conflictos.

La mejor manera de gestionar los conflictos es diseñar un buen esquema de base de datos y utilizar técnicas de fragmentación evitando la aparición de conflictos. Si esto no es posible, el software de sincronización tendrá que gestionarlos de una manera aceptable. El primer paso para corregir un conflicto es detectarlo.

Para ello, se deberán almacenar tanto los datos originales como los nuevos valores de cada fila modificada. En este caso, cuando los datos son modificados, la lógica de sincronización podrá comprobar si el valor original del cliente coincide con el valor actual en la base de datos empresarial. Si coincide, no hay conflicto. En cambio si no coinciden los valores, esto supone que otro usuario ha modificado los datos desde la última sincronización y que ha sucedido un conflicto. Para poder detectarlo un requisito es la existencia de identificadores únicos para cada fila de la base de datos.

Resolución de conflictos.

La resolución de conflictos puede ser complicada y dependiente de la empresa. Existen tres opciones generales para la resolución de conflictos:

- **Resolución de conflictos automática en el servidor.**

Existen una variedad de métodos que se pueden usar para resolver conflictos [Shapiro+04], [Hemant+99]. Muchas veces el servidor puede usar reglas de negocio y procesos automáticos para gestionar los conflictos, sin afectar al cliente. Una forma sencilla de solucionar los problemas es guardar los conflictos detectados en un registro y que el administrador los solucione manualmente.

Otra solución es que de manera automática el servidor tome la decisión basándose, por ejemplo, en la preferencia del usuario:

- El cliente siempre gana.
- El servidor siempre gana.
- El último cambio gana.

Esta última opción requiere almacenar el instante de tiempo en el que se hizo el cambio y sincronizarlo entre el cliente y el servidor.

- **Resolución personalizada de recursos de conflictos en el cliente.**

En muchos casos, el cliente es el mejor lugar donde se puede realizar la resolución de conflictos porque conoce el contexto en el que ha sido realizado. En este caso la aplicación podría resolver automáticamente los conflictos. Su principal problema es que el servidor debería enviar al cliente suficiente información para permitir al cliente tomar la decisión correcta.

- **Resolución por terceras partes.**

En algunos casos se requiere la participación de una tercera parte, normalmente, un administrador. A esta resolución se le denomina también resolución manual [Hassmann+02]. El administrador determinará cuál es la opción correcta a realizar.

Proceso de sincronización de datos

El proceso de sincronización consiste en una serie de fases:

- **Fase de subida (Upload).**

Objetivo: el cliente envía sus modificaciones de la base de datos al servidor para que las procese. El servidor recibe esas actualizaciones y gestiona los conflictos que se puedan producir. Como resultado de esta fase, se obtienen las nuevas actualizaciones a enviar a los clientes.

- **Fase de bajada (Download).**

Objetivo: el servidor envía los cambios al cliente para que actualice la base de datos.

- **Fase de confirmación.**

Objetivo: el cliente envía una confirmación de que todo ha salido correctamente al servidor antes de cerrar o dar por terminada la sincronización.

Modo de propagación de datos

Existen dos opciones de propagar los datos [Barbará99]:

- **Modelo Push.**

En este modelo, el servidor notifica al cliente y trata de enviarle datos. Aplicándolo a la sincronización, es el servidor el que envía los datos actualizados al cliente.

- **Modelo Pull.**

En este modelo es el cliente quien contacta con el servidor para realizar la sincronización. El cliente de manera periódica realizará la sincronización.

El modelo Pull es el más utilizado en los dispositivos móviles porque no siempre el servidor podrá contactar con el cliente, sobre todo si el cliente usa direccionamiento IP privado [Wallick03].

Eficiencia y rendimiento

Las soluciones de sincronización deben ser eficientes en la transferencia de datos entre los clientes y los servidores y en la cantidad de datos que son transferidos entre ellos. Existen diferentes técnicas para mejorar el rendimiento y que tendremos en cuenta a la hora de diseñar la solución propuesta en esta tesis, entre ellas:

- *Fragmentación*: La fragmentación permite que no se envíe toda la base de datos a los clientes sino que se envíen aquellos datos que verdaderamente necesitan los clientes. De esta manera, se consigue reducir el tiempo y la cantidad de datos necesarios a sincronizar.
- *Envío de los cambios*: Otra técnica muy utilizada es enviar exclusivamente aquellos datos que han sido cambiados desde la última sincronización (sincronización basada en timestamp y selectiva). De esta manera, igual que antes, se reduce el tiempo y la cantidad de datos a enviar.
- *Compresión de datos*: Otro modo de reducir la cantidad de datos transmitidos es implementar un algoritmo de compresión. De esta manera se reducirán los tiempos de sincronización y los costes de las comunicaciones. Su principal inconveniente es que la compresión y la descompresión de datos añade una sobrecarga extra en el proceso de sincronización tanto en la parte cliente como en la servidora. Otro aspecto a tener en cuenta es que los dispositivos móviles disponen de recursos mínimos por lo que no se podrán utilizar algoritmos de compresión complejos.

2.5.2 Enfoque orientado a servicios

Mientras que en el apartado anterior nos centramos en la gestión de los datos, en éste nos centraremos en la gestión de las invocaciones realizadas a diferentes servicios remotos desde un dispositivo móvil. La necesidad de acceder o invocar a servicios que se encuentran en Internet desde un dispositivo móvil se ha convertido en un aspecto esencial. Estos servicios son muy variados, van desde la simple descarga de una página web, pasando por la descarga de un vídeo, el envío de un SMS, hasta servicios más complejos como los servicios empresariales que la empresa pone a disposición de sus trabajadores. A estos últimos servicios son los que esta tesis pretende darles soporte buscando desarrollar una solución de movilidad que permita a los trabajadores móviles el acceso a los diferentes sistemas que cuenta la empresa (Base de datos, ERP¹¹, PRM¹², CRM¹³,...). De esta manera, se obtienen beneficios tanto para la empresa como para los trabajadores, tal y como detallamos en el capítulo uno.

Pero lograr este objetivo es un reto, ya que los sistemas con los que cuentan las empresas son sistemas que no fueron diseñados para facilitar la integración entre sistemas y, menos aún, para permitir la comunicación con terceros [Alonso+04]. Como resultado, una empresa que

¹¹ Enterprise Resource Planning.

¹² Partner Relationship Management.

¹³ Customer Relationship Management.

decide integrar sus aplicaciones internas con sus socios comerciales y de negocio, se encuentra con una tarea de un elevado costo, gran nivel de complejidad y una alta demanda de tiempo. Por lo que para realizar transacciones con sus proveedores, clientes y/o distribuidores, normalmente, se realiza de forma manual o parcialmente electrónica con todos los riesgos, desventajas y costos que esto implica.

Por todo ello, es necesario, por un lado, integrar las diferentes aplicaciones dentro de la propia empresa y, por otro lado, lograr la cohesión de estos procesos con los análogos de las empresas y clientes con las que se relaciona.

La solución más ampliamente adoptada para permitir la integración de estos sistemas es el uso de una arquitectura orientada al servicio (*Service-Oriented Architecture* [SOA]) y de una computación orientada al servicio (*Service-oriented computing*) [Carbonell+03]. En esta tesis hemos optado por una implementación de SOA basada en servicios web.

En este apartado detallaremos la necesidad de integración por parte de las empresas y cómo el uso de SOA basada en servicios web es una solución muy eficiente. Por último, detallaremos las técnicas utilizadas para la invocación de servicios en dispositivos móviles.

2.5.2.1 Soluciones de integración de procesos de negocio

La mayoría de las compañías poseen un entorno variado de sistemas antiguos, aplicaciones, procesos y fuentes de datos, que típicamente interactúan entre ellos a través de interconexiones que están pobremente documentadas y difíciles de mantener. La segmentación de los sistemas de información fue incrementada con la introducción de aplicaciones comerciales a medida como ERP, CRM, portales, etc. Estos sistemas fueron diseñados como “cajas negras” no permitiendo el acceso a los datos o procesos internos de los mismos. A pesar de que muchas de estas aplicaciones actualmente proporcionan un mejor acceso a los datos y a la lógica de negocio, la integración con otros sistemas de la empresa es todavía un reto.

La integración de los sistemas se puede clasificar en dos tipos [Apshankar+02]: integración de aplicaciones empresariales (*Enterprise Application Integration* [EAI]) e integración entre empresas (*Business-to-Business Integration* [B2BI]), los cuales detallaremos a continuación.

Integración de aplicaciones empresariales

La necesidad de comunicar sistemas dentro de la organización conduce a la evolución de EAI. EAI es una arquitectura tecnológica por medio de la cual se puede compartir información de fuentes y aplicaciones corporativas heterogéneas dentro de una empresa. Mediante la utilización de estos sistemas EAI, la integración y convergencia de las

aplicaciones de gestión empresarial internas de una empresa se realiza de forma eficaz, empleando mecanismos que facilitan la comunicación sistema a sistema y soluciones “end-to-end”. De este modo, se automatiza el flujo de información entre empleados, sistemas, aplicaciones corporativas y entornos de negocio B2B.

El objetivo es mejorar y optimizar los diferentes procesos de negocio internos de la compañía, habilitando mecanismos que permitan el flujo de información entre las aplicaciones de negocio.

Las soluciones EAI proporcionan un intercambio de datos entre ERP, CRM, SCM, bases de datos, data warehouses y otros sistemas dentro de la empresa desarrollados internamente como aplicaciones internas desarrolladas con diferentes lenguajes de programación (COBOL, C++, Java, .NET,...), middleware distribuidos como TP monitors, objetos distribuidos, CORBA¹⁴, RMI¹⁵, MOM¹⁶, DCOM¹⁷ y servicios web.

Integración entre empresas

B2BI es una arquitectura tecnológica por medio de la cual se puede compartir información de fuentes y aplicaciones heterogéneas entre las aplicaciones de negocio de diferentes empresas. B2B proporciona a las empresas una gran ventaja competitiva a través de la reducción de los tiempos de ciclos de venta, incremento del servicio del cliente,...

Entre los procesos de negocio compartidos se encuentran el comercio electrónico, redes colaborativas, etc. El mecanismo más comúnmente utilizado para solventar la necesidad de comunicación entre empresas que establecen una relación cliente-proveedor ha sido la implementación del intercambio de mensajes de negocio utilizando EDI (*Electronic Data Interchange*), enviando y recibiendo los mensajes por redes privadas de valor agregado conocidas como VAN (*Value Added Networks*).

2.5.2.2 Arquitectura orientada a servicio

Para una verdadera integración dinámica, los recursos software como aplicaciones, objetos y programas deberían estar débilmente acoplados. Estos recursos deberían proporcionar interfaces públicas que describen sus acciones y, además, la comunicación entre estos recursos debería estar basada en estándares abiertos.

¹⁴ Common Object Request Broker Architecture.

¹⁵ Java Remote Method Invocation.

¹⁶ Message-oriented Middleware.

¹⁷ Distributed Component Object Model.

Aquí es donde la arquitectura orientada a servicio (*Service-Oriented Architecture* [SOA]) entra en juego. SOA proporciona un framework que interconecta aplicaciones y componentes software, permitiendo invocar servicios de negocio remotos sin necesidad de escribir código. Una definición formal de SOA es la siguiente: “A service-oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. Some means of connecting services to each other is needed.” [ServiceArchi]

Otra definición proporcionada por Sun: “A service-oriented architecture is an information technology approach or strategy in which applications make use of (perhaps more accurately, rely on) services available in a network such as the World Wide Web. Implementing a service-oriented architecture can involve developing applications that use services, making applications available as services so that other applications can use those services, or both” [Ort05].

Últimamente, la computación orientada al servicio [Papazoglou+03] y las arquitecturas orientadas a servicios han ganado adeptos en la Ingeniería del Software. Ambas no son conceptos completamente nuevos; otras tecnologías de computación distribuidas como CORBA [OMG04] y DCOM [Horstmann+97] se basan en principios similares. SOA son extensiones de conceptos existentes y de nuevas tecnologías como servicios web que son usados para implementar una plataforma independiente de sistemas distribuidos. Lo que distingue a SOA de otras arquitecturas es que es débilmente acoplada. Esto significa que el cliente de un servicio es independiente del servicio. El modo en el que un cliente se comunica con otro servicio no depende de la implementación del servicio. El cliente se comunica con el servicio gracias a una interfaz bien definida, totalmente independiente de la implementación del mismo. Otras ventajas de SOA que se pueden destacar [Ort05] son la reusabilidad, la interoperabilidad, la escalabilidad y la eficiencia de coste.

En cambio, lo que es relativamente nuevo es la emergencia de SOA basada en servicios web. Un servicio web es un servicio que se comunica con los clientes a través de un conjunto de protocolos estándares. Estos estándares se utilizan en la mayoría de las plataformas y productos de los fabricantes de software, haciendo posible la comunicación entre clientes y servicios de manera consistente a través de un amplio espectro de plataformas y sistemas operativos. Esta universalidad hace que los servicios web sean la solución más utilizada para implementar SOA.

SOA es un área de actualidad en las empresas ya que ven el potencial de una arquitectura SOA en el desarrollo de aplicaciones. Gartner informa que “By 2008, SOA will be a prevailing software engineering practice, ending the 40-year domination of monolithic

- Para la segunda fase:
 - ▷ No se puede garantizar que el dispositivo móvil esté en online para recibir las respuestas de los servicios web por lo que será necesario la utilización de un intermediario (*Proxy*) que almacene las respuestas para que le sean enviadas al dispositivo cuando se encuentre en online, aquí se podría aplicar aproximaciones como WSMQ.
 - ▷ El mecanismo de planificación deberá tener en cuenta el tiempo máximo de espera de la respuesta por parte de la aplicación para recoger las diferentes respuestas que le esperan en el intermediario.

En esta tesis nos centramos principalmente en el diseño de una solución que permita llevar a cabo la planificación del uso de la interfaz inalámbrica teniendo en cuenta los dos enfoques: el orientado a datos y el orientado a servicios, basándose en una serie de políticas que tenga en cuenta aspectos como las necesidades de las propias aplicaciones (*application-aware*) como de los recursos del dispositivo móvil (*context-aware*), como, por ejemplo, son el tiempo de vida de la batería, el estado de los recursos, etc.

Concretamente, nos centramos en planificar cuándo es el momento más adecuado para llevar a cabo las sincronizaciones y cuándo es el mejor momento para realizar las invocaciones y la recepción de las respuestas de los servicios web, buscando realizar una gestión óptima de los recursos teniendo en cuenta las necesidades de las múltiples aplicaciones que se están ejecutando a la vez.

2.6 Gestión de la energía en la computación móvil

Los dispositivos móviles se han convertido en parte de nuestras vidas diarias. Estos dispositivos requieren de unas características para que sean útiles: deben ser ligeros, pequeños, portables y tener la habilidad de realizar una variedad de tareas incluyendo acceso a Internet móvil. Pero su principal debilidad es su dependencia a las baterías [Pouwelse03]. Éstas, al ser limitadas, hacen que se conviertan en un recurso esencial a tener en cuenta en el diseño de sistemas móviles.

El problema del consumo de energía¹⁸ está causado por numerosos factores. Por un lado, se encuentran los usuarios que demandan más funcionalidad, más procesamiento, tiempo de vida de la batería mayor y tamaño y peso menor. Por otro lado, la tecnología de las baterías

¹⁸ Se puede definir power consumption como “the rate at which energy is consumed” [Pouwelse03].

está progresando lentamente; el rendimiento mejora sólo un porcentaje pequeño cada año. Los dispositivos portátiles son cada vez más pequeños, lo que implica que el espacio para las baterías sea menor. La energía consumida por el sistema va a determinar el tamaño, peso y volumen de la batería que debe alimentar a dicho sistema, con lo que la batería se convierte en una parte significativa del total del peso y tamaño del sistema final. Por ello, la clave está en conseguir reducir el consumo de energía y a la vez conseguir un sistema ligero y pequeño. Por todo ello, es importante utilizar la energía disponible dentro de las baterías tan eficientemente como sea posible para cumplir las demandas de los usuarios.

En este apartado describimos las consideraciones más importantes sobre la gestión de energía que hay que tener en cuenta en los sistemas de computación móvil. Actualmente, no existe una solución óptima para alcanzar la eficiencia energética¹⁹. Se debe tener en cuenta todo el sistema incluyendo tanto las fuentes de energía (baterías) como los consumidores de energía (hardware y software). Cada aspecto del sistema tiene un impacto en la eficiencia de la energía y el impacto de un determinado subsistema es dependiente normalmente de sus interacciones con otros subsistemas. La computación sensible a la energía (*energy-aware computing*) persigue mantener el equilibrio entre dos objetivos: por un lado, las expectativas de rendimiento del usuario y, por otro, maximizar la cantidad de trabajo que el usuario puede hacer antes de que la batería se descargue [Martin03]. Schurgers define *energy or power awareness* como “the right power/energy at the right time and the right place” [Schurgers03].

Para ello, la *energy-aware computing* aplica una serie de técnicas que persiguen incrementar la energía suministrada o disminuir la demanda de la misma. Si se busca incrementar la energía suministrada, la solución es difícil. Históricamente, el rendimiento de la batería ha mejorado muy lentamente en el tiempo (sobre el 5-10% en las baterías Li-ion [Zhong05], en términos de energía disponible por unidad de tamaño o peso) [Flinn01] [Havinga00]. Es más, como se ha indicado, la necesidad de movilidad requiere que los sistemas sean más pequeños y ligeros como sea posible. Por lo que no podemos incrementar el tamaño de la batería para conseguir más energía sin atentar contra estas dos características. Ante esto, es de esperar que la tecnología de la batería sólo no resuelva los problemas de consumo de la energía en dispositivos móviles, por lo que vemos necesario desarrollar estrategias alternativas para la búsqueda del ahorro de energía y la gestión de la energía.

La búsqueda del ahorro de energía ha sido más fructífera históricamente. En primer lugar, los avances en el diseño de circuitos de bajo consumo han conducido al desarrollo de componentes hardware más eficientes en cuanto al consumo de energía. En segundo lugar, la gestión de energía del hardware (*hardware power management*) ha logrado reducciones de

¹⁹ Se puede definir energy efficiency como “doing more work with the same amount energy” [Havinga00].

- Sujeto a regulaciones de traslado: el traslado de cantidades importantes de baterías de Li-Ion puede ser susceptible de controles regulatorios. Esta restricción no se aplica a movimientos personales.
- Es cara: su precio supone cerca de un cuarenta por ciento más que la de NiCd.
- No totalmente maduro: cambios en metal y en las combinaciones químicas afectan los resultados de las pruebas de la batería.

Han surgido varios tipos de baterías de Li-Ion. La versión original de Sony usaba carbón como electrodo negativo (ánodo). Desde 1997, la mayoría de las baterías de Li-Ion, incluso la de Sony, han cambiado a grafito. Este electrodo proporciona una curva de tensión de descarga más plana que la del carbón y ofrece una curvatura cerrada, seguida por una caída de tensión rápida antes del corte de la descarga²¹ (ver la Figura 2.11) [Buchmann01].

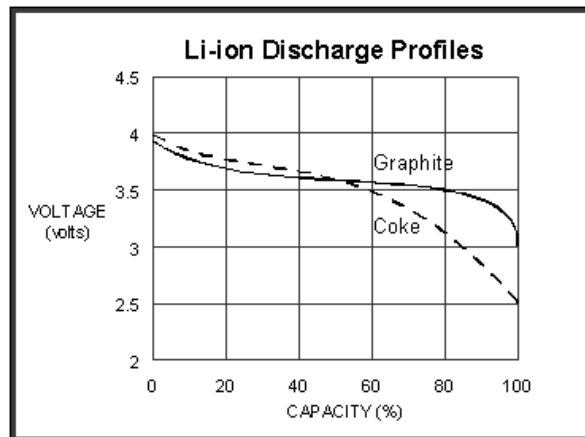


Figura 2.11 Características de descarga de Li-Ion con cobre y con grafito.

Consecuentemente, la energía útil del sistema de grafito puede ser recuperada descargando sólo a 3.0 voltios la celda, en tanto que la versión de carbón de Sony debe descargarse a 2.5 voltios para conseguir el mismo rendimiento.

Resumiendo, la batería Li-Ion recibe buenas calificaciones en cuanto a rendimiento y confiabilidad. La escasez de modelos se ha reducido y los precios son accesibles, consecuentemente, se equipan más equipos portátiles con baterías Li-Ion.

Las baterías Li-Ion han encontrado su lugar en el mercado con dispositivos portátiles que exigen una forma física pequeña. Los usos más populares son los teléfonos celulares y los ordenadores portátiles. Debido al problema del envejecimiento, las baterías Li-Ion son muy

²¹ El corte de descarga o Cutoff voltage se puede definir como "voltage at which a battery is considered fully discharged"[Martin99]

convenientes para aplicaciones de usuarios activos. Las baterías Li-Ion no satisfacen plenamente en las aplicaciones de elevada corriente, tales como herramientas de potencia, desfibriladores de corazón y radios móviles para seguridad pública.

Otro campo donde las baterías Li-Ion han mostrado ser menos favorables, es en aquellas aplicaciones que requieren solamente el uso ocasional de batería. En un portátil, por ejemplo, la batería de Li-Ion envejece con el tiempo y no se puede aprovechar todo el beneficio de la misma. El elevado nivel de temperatura en la mayoría de los portátiles hace también que las baterías Li-Ion envejezcan prematuramente. Las pruebas de campo revelaron, sin embargo, que las baterías Li-Ion se ven menos afectadas por el calor que las de NiMH.

2.6.3 Consumidores

El consumo de energía en los dispositivos móviles no depende solamente de un único componente. Diversos estudios [Lorch+99] [Truman+98] [Stemm+96] se han centrado en la investigación sobre el consumo de energía de dispositivos portátiles y la conclusión a la que han llegado es que no hay un único componente o una única actividad que domine el consumo de energía en un dispositivo portátil. Y aún es más, si se persigue reducir el consumo total de energía es necesario reducir el consumo de energía de todos los componentes.

Dependiendo del tipo de dispositivo, deberemos considerar diferentes tipos de componentes periféricos del mismo. Desde la perspectiva del consumo de energía, algunos de los componentes más relevantes son: el procesador, el disco duro, las pantallas, las interfaces de comunicación, etc. A continuación, detallaremos cómo cada uno de estos componentes afecta al consumo de energía total del dispositivo.

2.6.3.1 Procesadores

El consumo de energía de los microprocesadores de los dispositivos se ha incrementado significativamente [Olsen+06].

En la Figura 2.12 se muestra el incremento del consumo de energía de los microprocesadores típicos de Intel. Examinando el consumo actual de dos microprocesadores, el AMD Athlon™ 64 X2 Dual-Core Processor y el Intel Pentium D Dual-Core Processor, el primero ya ha alcanzado el consumo de 110 W y el segundo 130W [AMD07]. Combinando este dato con la figura anterior, podemos observar que ha habido un crecimiento exponencial en el consumo de energía por parte de los microprocesadores. Los microprocesadores para dispositivos portátiles no pueden seguir esta tendencia ya que las baterías no podrían soportarlos pero,

A la hora de abordar el diseño de la gestión de la energía, se lo podemos considerar como una adaptación a realizar en el sistema para que realice esta función, tal y como detallamos en el apartado 2.3. Desde este punto de vista, el diseño se puede clasificar en función del nivel donde se lleve a cabo esta adaptación (la gestión de energía de las aplicaciones). Por un lado, ciertos investigadores lo dividen en dos subniveles: nivel middleware y el nivel de aplicación [Zhong05], donde el middleware será el encargado de realizar la tarea de gestión de energía de todas las aplicaciones. Otros investigadores optan por integrar el módulo de gestión de energía con el nivel del sistema operativo [Havinga00] [Jones+01], mientras que otros optan por integrarlo junto con la propia aplicación [Abukmail+04].

Todos concluyen que existen tres o cuatro niveles de oportunidad para reducir el consumo de energía en un sistema:

1. A nivel hardware.
2. A nivel del sistema operativo, que incluye las comunicaciones.
3. A nivel de middleware y/o a nivel de aplicación (algunos incluyen el nivel middleware dentro del nivel de aplicación).

A continuación vamos a comentar cada uno de estos niveles empezando por el más bajo.

2.6.4.1 Hardware

La optimización de la energía a este nivel se centra en el óptimo diseño del componente hardware a través del uso de un mejor material, mejoras en la mecánica, el circuito y el diseño de la arquitectura [Karl03], aunque esta área queda fuera del alcance de esta tesis. Los fabricantes de componentes hardware pueden proporcionar mecanismos de ahorro de energía de sus componentes al Sistema Operativo a través de interfaces estándar. La interfaz más ampliamente usada es *Advanced Configuration and Power Interface* (ACPI) [Zhong05]. En el siguiente apartado abordaremos una descripción más detallada de ACPI.

Reducción de energía a través del óptimo diseño de la arquitectura

Realizar un diseño óptimo de la arquitectura puede ser una solución atractiva al problema de la energía seleccionando los componentes que supongan un menor consumo de energía. A modo de ejemplo, Patel et al. [Patel+06] se centran en añadir una nueva caché a sus diseños obteniendo un considerable ahorro de energía y Dutt et al. [Dutt+07] recogen diversas técnicas para reducir el consumo de energía.

Ajuste del rendimiento

Algunos dispositivos tienen la habilidad de permanecer activos pero reduciendo su consumo y rendimiento, equilibrando rendimiento frente al consumo de energía. El objetivo de este equilibrio debería ser cumplir con las expectativas de rendimiento del usuario mientras se incrementa la cantidad de trabajo que puede ser completada antes de que la batería se agote.

Las principales investigaciones que se han realizado en esta área se han centrado en establecer dinámicamente la velocidad de la CPU, ajustando la frecuencia del reloj de la CPU para poder realizar el trabajo solicitado mientras se consigue ahorrar energía [Yuan+06]. Weisser [Weisser+94] fue el primer investigador que construyó un sistema que podía modificar la velocidad de la CPU para reducir el consumo de energía. Estableció que la velocidad debería ser establecida por el sistema operativo buscando reducir el consumo de energía de la CPU mientras se cumplían los requisitos de rendimiento del usuario.

Esta técnica recibe el nombre de *Performance scaling*. Esta técnica no afecta a la funcionalidad del hardware pero cambia su rendimiento. A pesar de que normalmente ahorra menos energía que la gestión de energía, puede ser aplicada a más escenarios porque no modifica la funcionalidad.

El término *rendimiento* dentro de esta área tiene diferentes significados dependiendo del componente hardware donde se aplique. Por ejemplo, para los procesadores el rendimiento significa modificar la velocidad o frecuencia del reloj. Otro ejemplo, reducir el voltaje suministrado es una técnica muy efectiva para ahorrar energía [Yuan+06]. Por otro lado, para los discos duros, el rendimiento se aplica al ratio de datos. La técnica consiste en que el motor de los discos duros puede rotar a diferentes velocidades. A menor velocidad, menor es el ratio de los datos y el consumo de energía [Gurumurthi+03]. Para las pantallas, el rendimiento implica modificar el color, contraste o brillo. Con una pequeña reducción de una de estas tres características se consigue un ahorro significativo sin sacrificar su usabilidad [Cheng+04].

De todos ellos, a continuación vamos a proporcionar una descripción más detallada de la aplicación de esta técnica en los procesadores al ser la más ampliamente utilizada.

Ajuste del rendimiento en procesadores

En el caso de los procesadores, esta técnica se centra en utilizar los períodos inactivos que son demasiados cortos para que un procesador pueda entrar en modo de ahorro de energía. Reduciendo el rendimiento del procesador para eliminar estos períodos de inactividad, se puede conseguir ahorrar energía sin sacrificar el rendimiento del sistema. Por ejemplo, si el

Dentro de los estados C0/D0, los dispositivos y procesadores pueden disponer de diversos estados de rendimiento (P0 a Pn, donde $n < 16$), donde se encuentran completamente encendido pero se ejecutan con una potencia y un rendimiento reducido.

ACPI permite al sistema operativo interactuar con los componentes del sistema para obtener información sobre sus estados de energía y parámetros. Haciendo uso de esta información se pueden realizar transiciones entre estados de energía del sistema y los dispositivos buscando el máximo ahorro de energía. Decidir qué transiciones deben ser realizadas y cuándo realizarlas es un área de mucha investigación que recibe el nombre de *gestión dinámica de la energía* (*Dynamic Power Management* [DPM]) [Pettis+06] [Irani+05], que abordaremos en el siguiente apartado.

Gestión dinámica de la energía

Benini define DPM como: “is a design methodology that dynamically reconfigures an electronic system to provide the requested services and performance levels with a minimum number of active components or a minimum load on such components. DPM encompasses a set of techniques that achieve energy-efficient computation by selectively turning off (or reducing the performance of) system components when they are idle (or partially unexploited)” [Benini+00] o usando una definición más breve: “Dynamic power management decreases the energy consumption by selectively placing idle components into lower power states” [Simunic01] (pp. 6).

DPM se basa en dos supuestos [Benini+00]:

- La idea de que los componentes del sistema no siempre poseen carga de trabajo y aún así están consumiendo energía, por lo que no es necesario que estén activos siempre, consiguiendo un ahorro de energía.
- Es posible predecir con cierto grado de fiabilidad, las fluctuaciones de la carga de trabajo.

En los siguientes apartados detallaremos la implicación de cada uno de estos supuestos.

Estados de bajo consumo de los dispositivos

Algunos dispositivos disponen de múltiples estados de bajo consumo. Por ejemplo, los discos duros poseen de dos estados: *standby* y *sleep*. Estos discos consumen menos energía en el estado *sleep* comparado con el estado *standby*. En contrapartida, la transición entre estados conlleva un consumo de energía y un tiempo para que el disco duro esté operativo. La mayoría de las políticas de gestión de energía asumen la existencia de un único estado

working y un único estado *sleep*. Teniendo en cuenta que el dispositivo sólo puede servir peticiones en el estado *working*.

Como ejemplo de estos dispositivos se encuentran los controladores de entrada/salida, los discos duros, las interfaces de red y las pantallas. La técnica más ampliamente utilizada de ahorro de energía por parte del sistema en ordenadores convencionales es apagar los discos duros y la pantalla cuando se detecta un período de inactividad por parte del usuario.

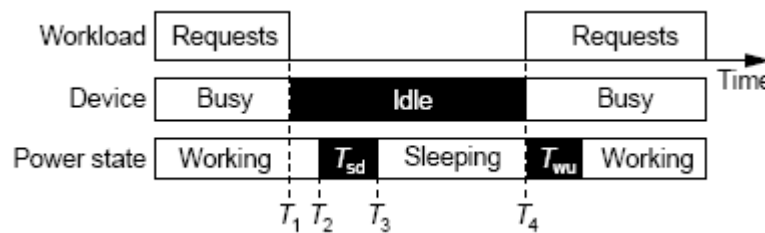


Figura 2.16 Gestión de la energía en desde el punto de vista de la carga de trabajo, el dispositivo y los estados de energía [Lu+01].

En la Figura 2.16 se muestra concepto de gestión de energía. La carga de trabajo consiste en múltiples peticiones. Para los discos duros, las peticiones son comandos de lectura y escritura; para las interfaces de red, las peticiones son paquetes a ser enviados o a ser recibidos. Cuando hay peticiones, el dispositivo está ocupado; en cualquier otra circunstancia, el dispositivo está inactivo (*idle*).

En el caso de esta figura, el dispositivo está inactivo entre T_1 y T_4 . Cuando el dispositivo está ocioso, se podría cambiar al estado *sleep*, logrando así un mayor ahorro de energía. En este caso, el dispositivo pasa al estado *sleep* en T_2 y despierta en T_4 cuando llegan nuevas peticiones. Hay que tener en cuenta dos efectos colaterales que posee el cambio de estados: en primer lugar, el cambio de estado conlleva un cierto tiempo, en este caso estos tiempos son T_{sd} y T_{wu} , el primero corresponde al tiempo necesario para pasar al estado *sleep* y el segundo corresponde al tiempo necesario para despertar al dispositivo y que se encuentre totalmente operativo. En segundo lugar, el cambio de estado también conlleva un cierto consumo de energía. En otras palabras, el cambio de estados de energía supone una sobrecarga.

A modo de ejemplo, en la Tabla 2.4 se muestran los diferentes estados de energía que posee una interfaz Bluetooth y una interfaz 802.11b, junto con el tiempo necesario para realizar la transición entre estados y el consumo de energía que supone esta transición.

- **Política codiciosa (*Greedy approach*)**

Esta aproximación es la más sencilla ya que no realizan ningún tipo de predicción. Consiste en que el componente pasa al estado de bajo consumo inmediatamente después de estar inactivo y se despierta cuando recibe una petición. Esta aproximación sólo es adecuada cuando la latencia de la transición de estado es muy corta, para así evitar la degradación del rendimiento. Por tanto, la capacidad de ahorro de energía se encuentra limitada. Debido a su simplicidad, esta aproximación ha sido adoptada ampliamente por los fabricantes y se ha implementado en muchos sistemas operativos. Por ejemplo, la tecnología *QuickStar* de Intel [Intel07] pone el procesador en modo de bajo consumo incluso entre dos períodos de tecleo. El kernel de Linux pone el procesador inmediatamente en modo de bajo consumo cuando no se halle un proceso ejecutándose.

- **Política de time-out**

En esta aproximación el componente hardware entra en el estado de bajo consumo cuando ha estado un determinado periodo de tiempo inactivo (denominado time-out). Al igual que la anterior aproximación, el componente despierta al recibir una petición. A pesar de su sencillez y amplio uso, no es capaz de explotar períodos de inactividad cortos como los que ocurren durante la interacción del usuario con el sistema.

Siguiendo con esta política, existen diversas variaciones dentro de la misma. Por ejemplo, algunas de estas políticas se basan en un time-out fijo. Esta aproximación está ampliamente adoptada en las soluciones comerciales. Por ejemplo, en Microsoft Windows, los usuarios pueden establecer valores de time-out para los discos duros y los monitores desde el panel de control. Otras políticas ajustan el valor de time-out durante el tiempo de ejecución. Por ejemplo, una política de time-out adaptativa ajusta este valor en base a los anteriores periodos de inactividad [Douglass+95].

- **Política predictiva**

Esta política se basa en calcular la duración del siguiente período de inactividad del componente basándose en su historia, es decir, su comportamiento pasado. En el momento que comience ese período que ha sido predecido, el componente entra en modo de bajo consumo inmediatamente sin esperar a un time-out. Por otro lado, el componente será despertado bien antes de que el periodo de inactividad predecido finalice o bien por la llegada de la petición. Como ejemplo esta política se encuentra LT (*adaptive Learning Tree*) que codifica la secuencia de períodos de inactividad pasados en nodos de árbol [Chung+99]. Esta política predice la longitud del período de inactividad mediante el uso de máquinas de estados finitos. Si el período de inactividad es más grande, el nivel de confianza aumenta. En caso contrario, el nivel de confianza decrece.

Mediante el uso del árbol y los niveles de confianza, se determinan los periodos de inactividad futuros. Otra política predictiva usa la longitud de los periodos de inactividad anteriores y el actual [Lu+01]. En [Yuan+03], hacen uso de la información proporcionada por la interfaz de usuario para predecir los periodos de inactividad durante la interacción del usuario. Despiertan al sistema antes de la siguiente entrada del usuario realizando un equilibrio entre el ahorro de energía y la degradación en la productividad del usuario.

- **Políticas estocásticas (*Stochastic approach*)**

La mayoría de las técnicas que siguen esta aproximación se basan en el siguiente modelo [Simunic+00], [Simunic02]: el componente a ser gestionado se representa como un proveedor de servicio (*service provider*), el cual usa una colección de estados que corresponden a sus modos de ahorro de energía. Se asume que las peticiones son generadas por un servicio denominado *service requester*. Las peticiones se almacenan en la cola de servicio (*service queue*) cuando el componente no está operativo [Simunic01], [Simunic+00a].

Benini et al. [Benini+00] [Benini+99] usan cadenas de Markov para modelar el servicio peticionario y la cola de servicio. Basándose en estos modelos, el problema de la gestión de energía se reduce a una estimación de diferentes parámetros de los modelos y a un proceso de decisión estocástico. En [Chung+02], extienden el modelo de servicio del peticionario y realizan una adaptación online para manejar carga de trabajo desconocida o no estacionaria.

Resumiendo, con esta técnica normalmente se obtiene una solución óptima cuando las suposiciones sobre las estadísticas del proveedor de servicio, peticionario y la cola sean verdaderas. Desafortunadamente, en la realidad, no todas estas suposiciones son siempre ciertas, haciendo que esta política de gestión de energía tenga resultados peores. Más aún, esta aproximación es mucho más complicada que las otras tres y conlleva un mayor consumo de los recursos [Lu+01]. Por lo tanto, no ha tenido mucha aceptación en la industria.

Como conclusión, en este apartado hemos presentado diversas técnicas predictivas que consiguen un ahorro de energía a nivel del sistema operativo. Analizando el estudio realizado por Lu [Lu+01], las políticas más óptimas son las estocásticas, pero como hemos detallado presentan problemas. Por ejemplo, se basan en suposiciones que no siempre son ciertas, son más complejas y requieren de un mayor consumo de recursos, tanto de memoria como de computación y de generación y mantenimiento de temporizadores. Siguiendo con el análisis de cuáles son las políticas más óptimas, las siguientes serían las predictivas que ofrecen un mayor ahorro de energía que las dos primeras, pero requiere de un gran

Dentro de las redes de área local inalámbricas, el estándar más utilizado en este nivel es el IEEE 802.11 [IEEE80211]. Este estándar proporciona una técnica de gestión de la energía que consiste básicamente en que todas las estaciones que operan en modo *Power-Save* (PS), se sincronicen y despierten al mismo tiempo. Si la estación recibe un indicador por parte de la estación base, de que tiene datos en espera de ser entregados, permanece despierta hasta que recibe los datos, sino es así vuelve al estado *sleep*. Esto se hará de manera periódica enviando un tipo de trama especial denominada *beacon*, que contiene tanto una marca de tiempo como un indicador de tráfico (TIM), el cual indica si hay tráfico esperando para la estación.

Existen diversos trabajos de investigación que exploran los diferentes modos de mejorar el mecanismo de conservación de la energía propuesto por IEEE 802.11 [Sailhan+04] [Karl03]. A modo de ejemplo, el protocolo EC-MAC (*Energy Conserving MAC*) [Sivalingam+00] que realiza una planificación del orden exacto de transferencia de los datos por parte de los nodos, con lo que las colisiones son evitadas y el número de retransmisiones se reduce.

Nivel LLC

Este nivel construye los paquetes y selecciona las capacidades de manejo de errores. Las técnicas de ahorro de energía en este nivel se centran en estas dos características.

Centrándose en los paquetes, el tamaño del paquete puede ser adaptado de acuerdo a las características del canal como propone Modiano [Modiano99]. Éste ha desarrollado un protocolo ARQ (*Automatic Repeat Request*) que adapta automáticamente el tamaño del paquete a las condiciones cambiantes del canal (tasas de error) con el objetivo de mejorar el rendimiento. Como efecto colateral, el consumo de energía se reduce. El reto de este protocolo es obtener una buena estimación de las condiciones del canal.

Centrándose en el control de errores, aparte de las interferencias, pérdidas de rutas y otras características típicas de los enlaces inalámbricos, la movilidad de red introduce otro factor que afecta a la calidad de la señal con lo que el ratio de error en el receptor puede variar drásticamente en un breve intervalo de tiempo.

Para mitigar estos inconvenientes, las redes inalámbricas desarrollan varios mecanismos de corrección de error denominados FEC (*Forward Error Correction*) y ARQ (*Automatic Repeat Request*) para proporcionar un cierto nivel de calidad de servicio. Se pueden conseguir ahorros de energía significativos implementando una estrategia de control de errores adaptativo que aplica los parámetros del nivel de enlace de datos dependiendo de la calidad del canal para minimizar el consumo de energía, modificando el grado de calidad de servicio ofrecido a los niveles superiores.

Otra técnica se centra en la predicción de las condiciones futuras del canal. Esta técnica conllevaría ahorros considerables de energía pero posee dos debilidades: su coste y la calidad de la predicción [Simunic05] [Karl03].

Nivel de red

La mayoría de los trabajos realizados en este nivel se centran en redes ad-hoc, concretamente en protocolos de encaminamiento para este tipo de redes. Éstos calculan el camino más corto y usan una métrica de coste basada en el número de saltos, asumiendo que el consumo de energía se incrementa proporcionalmente con el número de saltos. Las métricas de encaminamiento utilizadas en redes tradicionales no son adecuadas en este tipo de redes, ya que pueden sobrecargar unos nodos a favor de otros.

Existen dos tipos básicos de protocolos encaminamiento en este tipo de redes: los proactivos y los reactivos (también denominados *on-demand*). Los protocolos proactivos (por ejemplo, Optimized Link State Routing [OLSR] [Causen+03]) modifican sus tablas de encaminamiento periódicamente. Según Cano et al. [Cano+00], la energía consumida por los protocolos proactivos es estable con respecto a la carga de tráfico. Desafortunadamente, si la red está inactiva en términos de tráfico de datos, estos protocolos inducen a un consumo de energía continuo. Comparado con estos protocolos, los reactivos (por ejemplo, Ad-Hoc On-Demand Distance Vector [AODV] [Perkins+03]) calculan el camino de comunicación entre cualquier par de nodos sólo cuando se pide una comunicación entre los dos, consiguiendo así el ahorro de energía.

Martín [Martin+03a] propone un método de encaminamiento basado en la energía consumida por el subsistema de red, consiguiendo reducir el consumo de energía el 15% de media y la latencia en 75% frente a otros métodos que consideran sólo la potencia transmitida. Existen numerosos trabajos dentro de esta área [Zheng+02], [Harris+04] [Sankar04].

Nivel de transporte

Una vez que el camino es descubierto por un protocolo de encaminamiento, la transferencia de datos se realiza a través del protocolo de transporte como TCP o UDP. TCP es el más utilizado y es el responsable de verificar la entrega correcta de los datos desde el origen al destino. TCP fue diseñado para redes cableadas y presenta un buen rendimiento. Para redes inalámbricas, TCP exhibe un rendimiento peor, principalmente porque este tipo de redes sufre pérdidas de paquetes debido a la alta tasa de errores y a la movilidad.

TCP reacciona usando mecanismos de control de congestión mientras que los errores no son debidos a la congestión con lo que el rendimiento disminuye. Se han llevado a cabo varias investigaciones en este nivel a través de la simulación [Bansal+06] o de la experimentación [Agrawal+01].

En particular, los resultados presentados en [Agrawal+01], muestran que la frecuencia ante fallos en la ruta y el retardo en el establecimiento de la ruta son parámetros clave que afectan al rendimiento de TCP, con lo que la estabilidad de la ruta es crucial para alcanzar un buen nivel de rendimiento en TCP. Es más, el rendimiento cambia drásticamente, dependiendo del protocolo de encaminamiento utilizado [Ahuja+00].

Se pueden afrontar de diversas maneras como la creación de nuevos protocolos de enlaces para las redes inalámbricas y nuevos protocolos *end-to-end* [Karl03].

Nivel de aplicación y nivel middleware

Algunos autores, consideran la existencia de un único módulo a este nivel, mientras que otros consideran dos niveles independientes: el nivel middleware y el nivel de aplicación. Algunas de las técnicas que explicamos a continuación pueden ser aplicadas indistintamente en ambos niveles. La diferencia radica en quién es el responsable de implementar las técnicas bien la aplicación o bien el middleware. Las técnicas aplicadas en el nivel de aplicación y de middleware a un sistema inalámbrico son muy variadas, entre ellas, están la partición de tareas entre nodos móviles y fijos, el tratamiento de audio y video y la adaptación al contexto [Simunic05]. A continuación vamos a detallar algunas de las técnicas utilizadas para ahorrar energía en este nivel.

Reducción de los datos enviados

Desde que la transmisión de datos es cara, se persigue reducir el tiempo de transmisión por medio de técnicas de reducción de los datos y protocolos de transferencia de datos inteligentes. La técnica más usada para la reducción de datos es la compresión que reduce el tamaño de los datos a transmitir, pero requiere un consumo mayor de recursos para realizar la compresión. Aún así se consigue un mayor ahorro de energía comprimiendo los datos a pesar del consumo mayor por parte de la CPU [Kravets+00]. Por otro lado, el uso de protocolos de transferencia de datos conscientes del consumo de energía es otra de las técnicas utilizadas, por ejemplo, los protocolos podrían reducir el tamaño de sus cabeceras y el número de mensajes de control con lo que se ahorra energía, tal y como detallamos en el apartado 2.5.2.5.

Uso de proxies

Otra manera de gestionar el ancho de banda y la energía de las aplicaciones en terminales móviles es usar proxies. Éstos son middleware que adapta automáticamente las aplicaciones a los cambios en el ancho de banda y la energía. Pueden residir tanto en el dispositivo como en la estación base. Un ejemplo podría ser el uso de un proxy durante las transmisiones multimedia que suprima el vídeo en aquellos entornos donde haya poco ancho de banda o poca energía.

Como ejemplo de uso de proxies, se puede nombrar PAWP (*Power Aware Web Proxy*) [Rosu+04], una arquitectura diseñada para planificar el tráfico web entrante en intervalos de alta comunicación y ausencia de comunicación, de esta manera, permite a las interfaces inalámbricas pasar al estado *sleep* logrando un ahorro de energía. PAWP usa una colección de técnicas a nivel HTTP en un Proxy (como *prefetching & caching*) para compensar cualquier impacto negativo que la planificación del tráfico pueda producir.

Procesamiento de vídeo

El procesamiento y la transmisión de contenido multimedia requieren de un considerable consumo de energía como también de ancho de banda de red. De ahí que una gran parte de los esfuerzos en este nivel se centre en este tipo de aplicaciones. Existen diversas técnicas como la disminución de la calidad del vídeo para obtener ahorro de energía o utilizar técnicas de codificación y decodificación más ligeras. Ante situaciones de bajo ancho de banda y baja energía, podrían eliminarse ciertos frames antes de ser transmitidos buscando mantener una calidad de vídeo tolerable.

Chandra [Chandra02] investiga un protocolo específico de aplicación para reducir el consumo de la interfaz de red para aplicaciones multimedia de *streaming*. Su propuesta usa una estrategia basada en los hechos pasados para establecer el intervalo de *sleep*.

Weissel [Weissel+04] argumenta que una reducción tolerable del rendimiento o de la calidad depende de la aplicación y del usuario. Propone clasificar las aplicaciones por las características del tráfico de red definiendo un conjunto de perfiles de aplicación. Posteriormente, a cada aplicación se le asigna una política de ahorro de energía que logre el equilibrio entre el rendimiento y el consumo de energía.

Mohapatra et al. [VanAntwerpen+04] [Mohapatra+03a] han definido un framework específico para el streaming de vídeo. Proponen un framework que realiza una gestión de energía integrada que unifica optimizaciones a nivel de hardware (CPU, memoria y registros), a nivel del sistema operativo (DVS) y a nivel de middleware (control de admisión, transcoding óptimo, regulación del tráfico de red). Con el conocimiento de la configuración

de los dispositivos y las condiciones actuales, el middleware selecciona una calidad de vídeo determinada. Con respecto a la interfaz de red, proponen reducir el consumo de energía de la misma, manteniéndola en modo *sleep* el mayor tiempo posible. Para ello, el middleware planifica el tráfico para que sea enviado a ráfagas y así poder dormir el mayor tiempo posible.

2.6.4.4 Nivel de aplicación y nivel de middleware

Hasta ahora, hemos detallado las técnicas a bajo nivel que se centran principalmente en los componentes hardware del sistema. Hemos demostrado que aplicando sólo técnicas a nivel de hardware puede tener un impacto negativo en el consumo de energía. En la mayoría de los diseños, existe falta de comunicación entre los diseñadores de estos componentes y los diseñadores de los programas a ejecutar sobre estos componentes. Desafortunadamente, los ingenieros software, a menudo no son conscientes del consumo de energía de sus diseños. Y no existen herramientas adecuadas para estimar el consumo de energía de estos programas. Como consecuencia, los componentes se suelen usar de un modo que no permiten sacar el máximo beneficio a sus posibilidades de ahorro de energía [Zhong05].

Las soluciones software, como técnicas de gestión de energía de alto nivel se están convirtiendo en una opción a considerar para la gestión de energía. Ellis [Ellis+99a] propone un ejemplo en el que muestra la necesidad de la gestión de energía a alto nivel. En este nivel se pueden conocer los intereses de los usuarios y los desarrolladores de aplicaciones pueden tener en cuenta estos intereses y complementar este conocimiento con soluciones hardware para obtener mejores resultados.

La mayoría de las soluciones propuestas a este nivel se centran en un único tipo de aplicaciones y la mayoría no pueden ser generalizadas para permitir su uso en otras aplicaciones. La mayoría del trabajo se ha realizado en aplicaciones de procesamiento de señales e imagen debido a la cantidad de energía que consumen y al intensivo uso de la computación y/o de la pantalla.

Loy introduce el concepto de AM/PM (*active mode power management*) [Loy01]. Esta técnica consigue un ahorro 62 veces mayor comparado con la ausencia de gestión de energía y examina la cantidad de energía usada por una interfaz inalámbrica cuando no se necesita la conexión. En este punto, intentan ahorrar energía lo más posible. Las aplicaciones utilizadas en este trabajo fueron un cliente de correo y un navegador. La principal conclusión a la que llegaron en este trabajo es que cada aplicación es la que mejor conoce cuáles son sus necesidades y puede adaptarse a las necesidades de usuarios específicos.

Flinn [Flinn+04] lleva a cabo una relación colaborativa entre el sistema operativo y las aplicaciones para conseguir alcanzar los objetivos especificados por el usuario sobre el

tiempo de vida de la batería. En este trabajo, las aplicaciones se modifican dinámicamente para conservar energía. Ésto se realiza gracias a que el sistema operativo monitoriza la energía suministrada y la demandada. La idea es calcular la energía restante para comprobar si es suficiente para que la aplicación se pueda ejecutar manteniendo el tiempo vida de la batería especificada por el usuario. En caso afirmativo, la aplicación se ejecutará a alto rendimiento y si no, tendrá que modificar su funcionamiento ofreciendo un rendimiento menor pero con un consumo menor.

En [Haid+03], se toma como base un reproductor de MP3 (*Motion Picture Experts Group Audio Layer-3*). Este trabajo presenta el diseño de un reproductor de MP3 sensible a la energía. No sólo afrontan el diseño hardware sino también el diseño software. Se centra en dos tecnologías que influyen en el diseño de dispositivos portátiles: *System-on-chip* (SOC) y *System-in-package* (SIP). Analiza un sistema multimedia de un único chip a incluir en los dispositivos portátiles y realiza un análisis previo y detallado del consumo de energía.

Otros trabajos [Yuan+06] investigan también aplicaciones multimedia sensibles a la energía y presentan un middleware para coordinar la adaptación de las aplicaciones multimedia desde el nivel de aplicación al nivel hardware. El sistema diseñado persigue tres objetivos:

- Garantizar los requisitos de tiempo de la aplicación multimedia.
- Disponer de suficiente energía para que las aplicaciones terminen su tareas.
- Consumir la menor energía posible.

Para alcanzar estos objetivos, el framework hace tres contribuciones:

1. Hacer que la aplicación multimedia haga una reserva del procesador sensible a la energía.
2. Modelar formalmente la adaptación del hardware, software y las preferencias del usuario.
3. Proporcionar suficientes recursos de procesador y de energía a la aplicación, persiguiendo que el procesador vaya lo más lento posible y así ahorrar energía.

Weinlein y Faerber [Weinlein05] [Faerber04] presentan una aproximación que permite al usuario especificar sus demandas de rendimiento para diferentes perfiles de uso de cada aplicación. Estos perfiles se basan en las características de uso de la CPU, la interfaz inalámbrica y el disco duro de cada aplicación durante su ejecución. De esta manera, se puede predecir el consumo que tendrá un tipo determinado de aplicación en función del perfil al que pertenece y así poder garantizar el rendimiento especificado por el usuario.

Dentro del nivel middleware o de aplicación, las técnicas más comunes son:

Prefetching y caching

Esta técnica ha sido usada para mejorar el rendimiento en muchas aplicaciones y sistemas de ficheros. El cacheo de los datos solicitados frecuentemente en los dispositivos portátiles es un mecanismo prometedor para reducir los requisitos de ancho de banda de las redes inalámbricas como del consumo de energía. El reto es mantener el contenido de la caché consistente con respecto al almacenado en el servidor. Esto es, desafortunadamente, difícil de asegurar debido a las frecuentes desconexiones derivadas de la movilidad de este tipo de clientes. En entornos móviles, estas técnicas son usadas por estos sistemas para limitar el consumo de energía y comunicación causada por la movilidad y mejorar el rendimiento y la disponibilidad de los servicios [Tan04].

En [Cao04], Cao realiza un estudio de la gestión de una caché sensible a la energía en entornos móviles, proponiendo el esquema de adaptación básico para ahorrar energía durante el *prefetch*. La conclusión que saca es que dentro de unos márgenes, la técnica de *prefetching* reduce el consumo de energía, fuera de ellos, el consumo aumenta.

Irmak [Irmak+05] estudia diversas técnicas (optimizaciones de protocolo, compresión,...) para reducir la sobrecarga introducida por el cacheo en entornos con pequeño ancho de banda.

Técnicas de compilación

La investigación realizada hasta ahora ha demostrado que la compilación no tiene en cuenta el consumo de energía [Valluri+01]. Cada vez más se están realizando esfuerzos en esta área para contribuir al ahorro de energía. Es más, esta solución es atractiva para los programadores de aplicaciones que no tendrían que preocuparse sobre el consumo de energía puesto que lo realizaría el compilador de manera transparente.

Los compiladores optimizados para lograr la eficiencia energética utilizan diferentes técnicas como la reorganización de las instrucciones y la reducción de los operandos de memoria [Heath+02] [Havinga00].

- **Reorganización de las instrucciones**

Esta técnica consiste en seleccionar y ordenar las instrucciones de un programa para reducir la actividad del bus de instrucciones buscando el ahorro de energía. Los compiladores tienen en cuenta el coste de energía de cada instrucción (que se determina a priori). Un compilador sensible a la energía busca equilibrar el tamaño y la velocidad

frente a la reducción de energía. La energía consumida por un procesador depende del estado previo del sistema y de las actuales entradas. Es decir, le afecta el orden de las instrucciones.

- **Reducción de los operandos de memoria**

Se ha demostrado que las instrucciones con operandos de memoria tienen un consumo más alto frente a las instrucciones con operandos de registro [Abukmail+04]. Las optimizaciones de los compiladores intentan reducir el número de operandos de memoria.

Remote Task Execution

La ejecución remota es una técnica prometedora para extender el tiempo de vida de la batería en entornos limitados. Usando esta técnica, se permite ejecutar ciertas las tareas en otros dispositivos como proxies, logrando así considerables ahorros de energía. La clave del éxito de esta técnica es la correcta predicción. Es decir, se debe predecir cuándo el coste de la ejecución remota supera los beneficios. Por ejemplo, si una tarea se realizaría en menor tiempo si se ejecuta de forma remota, entonces ésta debería ser la solución. Por el contrario, si el coste de la ejecución remota excedería a la local, por ejemplo, implica la transferencia de grandes cantidades de datos, la ejecución debería ser local [Balan+03].

Existen diversos trabajos que se centran en esta temática. Por ejemplo, Rudenko [Rudenko+98] introduce mecanismos de ejecución de tareas remotas migrando procesos enteros a servidores remotos y entonces el cliente, que permanece en espera, recibe los resultados del servidor. Przybilski [Przybilski05] realiza un estudio donde recoge los beneficios de esta técnica. En el trabajo [Flinn+99], se utilizan como base aplicaciones desarrolladas en Java, migrando la compilación de un programa Java a un servidor debido al alto consumo de energía de la misma.

Otro sistema de ejecución remota es el sistema Spectra, también desarrollado por Flinn. Spectra decide dónde y cómo ejecutar las aplicaciones. Para tomar esta decisión, Spectra recolecta información de una variedad de monitores de recursos tanto en el cliente como en el servidor. En particular, Spectra monitoriza la carga de CPU del cliente y del servidor, el ancho de banda, la batería y el estado de la caché de ficheros. Las decisiones de Spectra se basan en tres métricas: latencia en la ejecución, tiempo de vida de la batería y la calidad de la aplicación.

Gurun et al. [Gurun+03] ha desarrollado una plataforma de procesamiento remoto denominada RPP (*Remote Processing Platform*) basada en una arquitectura cliente/middleware/servidor. El cliente y el middleware residen en el dispositivo móvil.

Existe un módulo que analiza si la tarea es aconsejable su ejecución en modo remoto, si es así se procede a su realización en el servidor. El middleware gestiona la comunicación con el servidor y los componentes del dispositivo móvil y monitoriza diversos parámetros. El servidor realiza la tarea y devuelve los resultados al middleware. Gurun et al. [Gurun+04] define una extensión denominada NWSLite que predice el coste de la ejecución remota de una tarea, con lo que permitirá evaluar la viabilidad de llevar a cabo una ejecución remota frente a una local.

2.7 Gestión inteligente de las comunicaciones.

La gestión de las comunicaciones es un área que a lo largo de los años ha ido adquiriendo mayor protagonismo, sobre todo por parte de los operadores de telecomunicaciones, que buscan desarrollar nuevos servicios para los usuarios, gestionando de forma más eficiente las comunicaciones, facilitando entre otras funciones, el encaminamiento eficiente de paquetes, balanceo de carga, movilidad [Johnston+04], servicios de valor añadido como servicios de localización [Vazquez+05], etc. La gestión de las comunicaciones también se puede ver no sólo desde la perspectiva del operador sino también desde la perspectiva del usuario. Desde esta perspectiva, se persigue realizar una gestión óptima de las comunicaciones desde el dispositivo. Esta tesis se centra en esta última perspectiva.

Dentro de esta perspectiva, entre las funciones que se suelen afrontar se encuentran:

- *Detección y monitorización de la conectividad:* El propio sistema operativo debe proporcionar APIs que permitan consultar a los drivers de la interfaz de red sobre el estado de la misma. De esta manera, las aplicaciones se podrán adaptar a los cambios de estado de la red [VanAntwerpen+04] [Ochi+04].
- *Selección del tipo de red más adecuado:* Ésta es un área muy activa, donde actualmente existen diversos proyectos que abordan esta problemática. Entre ellos, destacar Multinet [Multinet07], proyecto europeo financiado por el 6º Programa Marco, en el que vamos a desarrollar una plataforma que persigue tres objetivos: proporcionar un mecanismo eficiente para terminales *multihoming* móviles, desarrollar algoritmos de selección de redes centrado en los usuarios y adaptar dinámicamente los servicios ante cambios en la red.
- *Gestión de la conexión manual:* Se debe poder dar la posibilidad a un usuario de trabajar en offline de manera voluntaria aunque exista conectividad persiguiendo el ahorro en el coste, energía, etc. [Anand+05]. En esta tesis gestionaremos la conexión a través de un algoritmo que busca realizar una gestión inteligente de las comunicaciones.

Entre las técnicas utilizadas para la mejora de la eficiencia y el rendimiento se encuentran:

- *Envío de los datos necesarios*: Por ejemplo, en una base de datos móvil sólo se almacenarían los datos que verdaderamente se necesitan utilizando fragmentación. Otro ejemplo, sería enviar sólo los cambios y no toda la base de datos durante la sincronización (sincronización selectiva y basada en timestamp).
- *Compresión de datos*: El uso de algoritmos de compresión que reduzcan la cantidad de datos a transmitir es una técnica que mejora la velocidad y el coste así como la eficiencia. Aunque debemos tener en cuenta que su uso puede conllevar un aumento del consumo de energía por parte de la CPU (ver apartado 2.6.4.3).
- *Transcoding de vídeo*: Permite modificar la calidad del vídeo para lograr entre otros parámetros, reducir el tamaño, reducir el consumo de energía,... (ver apartado 2.6.4.3).
- *Planificación del tráfico*: Permite organizar el tráfico teniendo en cuenta los objetivos perseguidos de una manera eficiente.

Entre otros aspectos a tener en cuenta se encuentra la gestión, tanto del tráfico de entrada como el de salida, y el tipo de tráfico gestionado.

En esta tesis, pretendemos diseñar un módulo que realice una gestión inteligente de estos recursos teniendo en cuenta las necesidades de las aplicaciones en cuanto a la actualización de los datos y a la invocación de servicios para que se realice de manera óptima y eficiente.

A continuación realizamos una comparativa de diferentes soluciones propuestas que se centran en esta área:

- **Mohapatra**

La solución propuesta por Mohapatra et al. [Mohapatra+03a] aplica por un lado, la técnica de DPM sobre la interfaz de red persiguiendo disminuir el consumo de la misma. Por otro lado, aplica la técnica de transcoding de vídeo que adapta la calidad de vídeo al nivel de energía disponible en el dispositivo móvil y, por último, se realiza una planificación del tráfico por parte del middleware del proxy para lograr disminuir el consumo de energía, manteniendo la interfaz inalámbrica en modo *sleep* el máximo tiempo posible, mediante el envío del tráfico a ráfagas.

Las principales características a destacar, que se recogen en la Tabla 2.6, son que toda la gestión de las comunicaciones se realiza en el propio proxy, que sólo se gestiona el tráfico entrante del dispositivo y que únicamente está enfocada a streaming de vídeo.

Capítulo 3

Estudio de las Soluciones de movilidad y Gestión de recursos previos

Un conjunto de estudios y trabajos precedentes nos han servido como inspiración para el desarrollo de la solución de movilidad objeto de esta tesis, bien porque las conclusiones obtenidas por éstos o las soluciones adoptadas nos han servido como base para la construcción de ciertas partes de la solución propuesta, bien porque presentan limitaciones o exponen necesidades que han motivado y justificado el desarrollo de esta tesis y su interés. En este capítulo describimos estos trabajos y mostramos sus limitaciones.

Cabe decir que se trata de un conjunto de estudios inconexo y que no hemos perseguido recoger todos los trabajos o estudios relevantes en las áreas incluidas en esta tesis, sino aquellos que se encuentran más íntimamente relacionados con este trabajo de tesis en alguno o en ambos de los términos anteriormente indicados.

Al final de la descripción de cada uno de los trabajos mostraremos una tabla resumen con los aspectos más destacados, focalizándose, como ya se ha dicho, en resaltar las necesidades, limitaciones, problemas o áreas de trabajo futuras propuestas, es decir, en las oportunidades de mejora, tanto documentadas por los propios autores del trabajo, como extraídas de un análisis personal de los mismos; y en destacar aquellas conclusiones extraídas e ideas o soluciones adoptadas en ellos y de utilidad para el desarrollo de la solución. Estas tablas servirán de enlace con el siguiente capítulo, donde ya describiremos la solución propiamente dicha, y profundizaremos en sus aportaciones con respecto a estas oportunidades de mejora identificadas.

3.1 Aspectos a analizar

Para llevar a cabo el estudio y la comparativa de las soluciones existentes hemos seleccionado una serie de aspectos, a partir del estado del arte realizado en el capítulo anterior, que nos servirán como base para la evaluación de cada solución. Estos aspectos son los siguientes:

1. Características generales

En este punto recogemos los aspectos más generales sobre la solución analizada. Estas características son:

- *Contexto*

Esta característica se centra en definir qué entidad o entidades han realizado la solución, la fecha de realización y otros proyectos relacionados que haya que tener en cuenta.

- *Plataforma*

Esta característica recoge tanto el dispositivo como el sistema operativo que se ha utilizado para el desarrollo de la solución.

- *Entorno*

Esta característica se centra en definir si el entorno en donde se ha desarrollado la plataforma es centralizado o distribuido.

2. Operación online/offline

En este punto describimos si la solución propuesta soporta la operación online/offline tanto orientado a datos como a servicios, junto con las técnicas utilizadas, en caso afirmativo.

- *Enfoque orientado a datos.*

Esta característica se centra en definir si la solución aporta alguna solución para el tratamiento de datos durante la desconexión, junto con una descripción de su funcionamiento.

- *Enfoque orientado a servicios.*

Esta característica se centra en definir si la solución aporta alguna solución para el tratamiento de las invocaciones a servicios durante la desconexión, junto con una descripción de las técnicas empleadas.

3. Gestión inteligente

Consiste en detallar cómo la solución lleva a cabo una gestión inteligente sobre las aplicaciones o los recursos.

- *Objeto*

Indica cuál va ser el objeto que va a ser gestionado, por ejemplo, el código de la aplicación, un vídeo, un documento de texto, etc.

- *Modelo de adaptación*

Indica cuál/es son los modelos de adaptación que soporta. Éstos pueden ser (ver apartado 2.3): libre, transparente a la aplicación y sensible a la aplicación. Dentro de esta última además, se puede subdividir en: aplicación con soporte a la adaptación y sin soporte a la adaptación.

- *Lógica*

- ▷ *Lugar*

En este aspecto detallamos en qué nivel se realiza la gestión de la adaptación junto con las técnicas utilizadas en cada nivel donde se aplica la adaptación de la solución analizada. Estas técnicas corresponderán a las definidas en los apartados 2.6 y 2.7. Dentro del nivel middleware, las podemos clasificar en función de cómo proporcionen la adaptación: como un servicio común (*common service*) o un servicio específico del dominio (*domain-specific service*) [Flinn+04]. Como un servicio común, la gestión se aplicaría a través de un mecanismo genérico igual para todas las aplicaciones. Mientras que como un servicio específico del dominio, la gestión de energía se vería limitada a resolver las necesidades de un cierto dominio de aplicaciones, por ejemplo, aplicaciones de *streaming*, proporcionando una gestión más personalizada para este tipo de aplicaciones.

▷ Gestión basada en políticas

En este aspecto recogemos si la solución analizada basa su funcionamiento en políticas que indiquen cuándo y cómo debe adaptarse para lograr que el objetivo perseguido. Además, en este aspecto indicamos si se usan *políticas estáticas*, es decir, definidas durante el diseño, la compilación o durante la carga de la aplicación, o si se usan *políticas dinámicas*, es decir, definidas en tiempo de ejecución. Además, debemos definir cuál es el ámbito de aplicación de la política: si son políticas definidas para todo el sistema, por aplicación o por usuario.

4. Gestión inteligente de la energía

En este aspecto nos centramos en describir características a alto nivel que incluye la solución analizada sobre la gestión de la energía.

- *Niveles de gestión de la energía*

En este aspecto recogemos en qué niveles han introducido la gestión de energía. Como ya hemos detallado, estos niveles serán: hardware, sistema operativo, middleware, aplicación.

- *Dispositivos gestionados*

Consiste en indicar sobre qué dispositivos se realiza la gestión de energía.

5. Gestión inteligente de las comunicaciones

En este aspecto detallamos si se lleva a cabo una gestión de las comunicaciones, indicando el lugar donde se realiza y las técnicas empleadas, entre las definidas en el apartado 2.7.

6. Soporte de varias aplicaciones

Hace referencia a si la solución soporta la ejecución de varias aplicaciones simultáneamente, teniendo en cuenta que podrán producirse conflictos internos, por ejemplo, entre aspectos diferentes como el rendimiento, la calidad y el consumo de energía y conflictos externos con otras aplicaciones. Estos conflictos pueden ser evaluados localmente, en el lado del cliente, o globalmente, evaluados por una entidad externa.

3.3 Comparativa

A continuación vamos a proceder a realizar una comparativa de todas estas soluciones indicando qué mejoras incorpora la solución propuesta en esta tesis.

- **Características generales**

Centrándose en la plataforma de prueba utilizada por las diferentes soluciones, existe un mayor número de proyectos que han sido desarrollados sobre portátiles (o PC de sobremesa) frente a los desarrollados sobre PDA. Aquellos que han utilizado portátiles disponen de un entorno con suficientes recursos para dar soporte a soluciones potentes sin tener que preocuparse de la carga que supone la ejecución de los mismos. Pero el principal reto es diseñar soluciones que sean igualmente eficientes para entornos tan limitados como las PDA. Éstas como se vio en el primer capítulo, cada vez se utilizan más, sobre todo para dar soporte a servicios empresariales, por lo que es un área a tener en cuenta para desarrollos futuros. En esta tesis, la solución está orientada completamente al uso de PDA para dar soporte a los servicios empresariales disponibles para los empleados. Como se puede ver en las tablas anteriores, varias soluciones optan por el uso de PDA en sus desarrollos, pero la principal carga de trabajo, normalmente, es delegada a un Proxy, quien se encarga de realizar la tarea de adaptación, enviándole los resultados a la PDA. De esta manera no se tienen que enfrentar a su limitado entorno a la hora de diseñar su solución. En esta tesis nos centramos en un entorno distribuido donde la adaptación se realiza en la propia PDA y donde a la hora de realizar el diseño de la solución se han tenido en cuenta los recursos tan limitados de los que se dispone.

La mayoría utilizan como sistema operativo Unix o Linux, pero éste no es el más extendido en esta plataforma, siendo Windows el que domina el mercado²³ y para el que existe un mayor número de aplicaciones desarrolladas. Además, diversos estudios indican que el uso de Unix en el entorno empresarial queda relegado principalmente a los servidores y no se prevé que se extienda a los terminales de usuario [Andago04], [Bozman+02]. De ahí, que en esta tesis utilicemos el sistema operativo Windows.

Normalmente, el uso de Unix o alguno de sus variantes como sistema operativo les viene impuesto por la necesidad de modificar el propio sistema operativo para desarrollar su solución de adaptación, por lo que requeriría de una nueva instalación del sistema operativo en cada dispositivo. Esta situación no es aconsejable y no es viable cuando cualquier cambio en el sistema operativo puede producir el mal funcionamiento

²³ Según Gartner, Windows Mobile ocupó el primer puesto en 2004, ya que el 43% de las PDAs vendidas contaban con su sistema operativo, mientras que el 20,7% se lo disputan principalmente Linux y Symbian.

de las propias aplicaciones que sobre él se ejecutan y puede conllevar a la reinstalación del resto de aplicaciones.

En cuanto al entorno para el que se ha diseñado las diferentes soluciones, existe un claro equilibrio entre los centralizados y los distribuidos. En esta tesis nos centramos en un entorno distribuido Cliente/Proxy/Servidor, pero buscamos optimizar el funcionamiento del cliente, de ahí que las adaptaciones se realicen en el middleware del cliente.

- **Operación online/offline**

Gracias a esta funcionalidad, se da soporte a las aplicaciones para que puedan trabajar en offline. Tal y como se vió en el capítulo anterior, la operación online/offline puede ser considerada desde dos puntos de vista: enfoque orientado a datos y enfoque orientado a servicios. De las soluciones analizadas aproximadamente la mitad incluyen esta funcionalidad en sus arquitecturas, mientras que el resto se centran principalmente en aspectos como la gestión de la energía del dispositivo. De entre las que sí proporcionan algún soporte a la operación online/offline, todas excepto una [Sairamesh+04], sólo dan soporte a un único enfoque y con diferentes niveles de profundidad.

La solución propuesta por IBM [Sairamesh+04] es la única que muestra, al igual que esta tesis, la necesidad de dar soporte a ambos enfoques, pero la profundidad que ofrece de estos enfoques es muy ligera. Además, esta solución se centra en dar soporte exclusivamente a procesos de negocio previamente definidos.

Con respecto al enfoque orientado a datos, las soluciones se centran, por un lado, en el uso de bases de datos móviles (Thor, IBM) y, por otro lado, en la sincronización de objetos definidos explícitamente para la solución (Replet System, SDOs). En esta tesis, hacemos uso de bases de datos móviles y su posterior sincronización, permitiendo el acceso a los datos a las aplicaciones durante la desconexión.

Con respecto al enfoque orientado a servicios, lo primero que hay que destacar es que las invocaciones son de diversos tipos (objetos RMI, CORBA, tráfico HTTP). De todas las soluciones, sólo dos (IBM, iMMS) destacan la importancia de la integración con los servicios empresariales, área en la que se centra esta tesis. La propuesta de IBM indica que para llevar a cabo la integración, la solución más adecuada sería el uso de una arquitectura orientada a servicios basada en servicios web, mientras que la arquitectura iMMS no hace uso de servicios web para la integración, sino que requiere del desarrollo de adaptadores específicos para lograr la conversión entre el protocolo de transporte MMS, utilizado por la plataforma, y el protocolo de comunicación utilizado por los sistemas empresariales. Si se utilizasen servicios web, el beneficio sería mayor ya que

cada vez más, los diseñadores de los sistemas empresariales proporcionan estos adaptadores.

- **Gestión inteligente**

En cuanto a la gestión, los objetos gestionados por las soluciones analizadas son muy variables incluyendo tareas, vídeo, tráfico web, tráfico de red, invocaciones, etc. Con respecto al modelo de adaptación, los más utilizados siguen el modelo transparente a la aplicación y sensible a la aplicación con soporte a la adaptación, aunque les sigue el modelo sensible a la aplicación sin soporte a la adaptación a corta distancia.

Examinando el modelo transparente a la aplicación, posee la ventaja principal de que la gestión es totalmente transparente para las aplicaciones, no requiriendo de ninguna modificación de las mismas. En las soluciones expuestas, normalmente se centran en servicios comunes que ofrece el sistema (PARM, ECOSystem, iMMS, SodaSync), como el planificador de tareas. Además, resaltar que para conseguir la transparencia, en la mayoría de las soluciones toda la adaptación se realiza en el proxy (Pagoda, PAWP).

En el modelo sensible a la aplicación sin soporte a la adaptación, la aplicación participa en la gestión proporcionando información sobre sus necesidades pero no realiza la adaptación, de ahí que no sea necesaria la modificación del código. Y por último, en el modelo sensible a la aplicación con soporte a la adaptación, la aplicación se involucra totalmente en la gestión realizando toda o parte de la adaptación. Su principal inconveniente es la necesidad de diseñar la aplicación de manera explícita para que realice las funciones de adaptación y que cualquier modificación implicará cambios en la misma. Su complejidad de diseño, implementación y mantenimiento será muy elevada.

En esta tesis proponemos implementar una solución híbrida, es decir, que sea tanto transparente a la aplicación como sensible a la aplicación sin soporte a la adaptación con respecto a la gestión inteligente. Aunque como hemos visto, la última sería la aproximación más deseable ya que permite una gestión más eficiente. Pero como no es posible obligar a toda aplicación a definir sus preferencias, sobre todo si ya está operativa, para este tipo de aplicación hemos optado por el modelo transparente a la aplicación, para que se beneficien de manera totalmente transparente de la gestión mediante el uso de una política por defecto.

Con respecto a la lógica, la mayoría de las soluciones se basan en una gestión basada en políticas con diferentes grados de complejidad y con una gran variedad de parámetros. Un aspecto muy importante es si las políticas son estáticas o dinámicas. La mejor opción son las dinámicas al no requerir tener que reiniciar el sistema para su modificación.

En esta tesis, la solución permite la utilización de políticas dinámicas e incluso se da la posibilidad a cada aplicación de poder aplicar excepciones a la política en tiempo real.

Otro aspecto crucial es el lugar donde reside esta lógica, por ejemplo, si reside en la propia aplicación, modificarla implica la modificación de la aplicación. Si reside en cualquier otro nivel, será o no fácilmente modificable en función del nivel donde resida. En estas soluciones analizadas, existen representantes de cada nivel tanto en el sistema operativo (ECOSystem, Cooperative I/O), como a nivel de middleware (PAWP, DSM) como a nivel de aplicación (Odyssey).

En esta tesis, la lógica reside en el middleware, de manera que se encuentra centralizada.

- **Gestión inteligente de la energía**

De las soluciones analizadas, un gran número de ellas se centran en gestionar de manera óptima la energía disponible. En contrapartida, de todas ellas, sólo dos se centran en dar soporte online/offline a las aplicaciones.

Con respecto al nivel donde se lleva a cabo esta gestión, un gran número de soluciones se centran en el nivel del sistema operativo, combinado o no con otros niveles. Muy pocas soluciones se centran en el nivel de aplicación, ya que su uso implica que las propias aplicaciones realicen la gestión, por ejemplo, adaptándose a cambios. Y, por último, el resto se centran en el nivel middleware.

Con respecto a los dispositivos gestionados, la mayoría centran su gestión en la CPU, seguido del disco duro y la interfaz de red y, por último, la pantalla. En esta tesis nos centraremos en la gestión del consumo de la interfaz inalámbrica.

- **Gestión inteligente de las comunicaciones**

De las soluciones analizadas que tratan este aspecto, la mayoría gestionan las comunicaciones como medio para reducir el consumo de energía. En cuanto al lugar donde se realiza, la mitad se realiza en el middleware del Proxy y la otra mitad se realiza en el sistema operativo del propio dispositivo móvil. En cuanto a las técnicas empleadas la que más predomina es la planificación del tráfico, que será la que también utilizaremos en esta tesis.

- **Soporte de varias aplicaciones**

La resolución de conflictos entre aplicaciones no es un tema que abarquen la mayoría de las soluciones, aunque sí es un aspecto a tener que en cuenta debido a que las adaptaciones que realice una aplicación pueden afectar al resto de aplicaciones. Para

- *PYMEs con filosofía de alta comunicación interna*: empresas donde el flujo de información entre los integrantes de la organización es alto. Estas políticas repercuten en gran medida en la capacidad de los integrantes de la organización para responder a imprevistos y su grado de implicación en la misma.
- *PYMEs con sistemas críticos (como puede ser el caso de la protección de datos)*: empresas que necesitan un acceso seguro a sus datos internos o confidenciales desde cualquier entorno.
- *PYMEs con tareas de mantenimiento*: empresas con parte de su plantilla fuera de la empresa realizando labores de mantenimiento, de equipos o instalaciones, que pueden enviar los datos resultantes de su visita al sistema central o requerir datos a este sistema para que les dé soporte en su labor de mantenimiento.

Tras un análisis de los casos de éxito de las soluciones existentes en el mercado, otra de las conclusiones extraídas del estudio es que las PYMEs con necesidades de movilización de sus servicios o aplicaciones empresariales son aquellas que necesitan disponer de datos en tiempo real. Por lo tanto, las PYMEs con necesidades de movilización podemos agruparlas en los siguientes tres subsectores de mercado:

- *SubSector 1*: Sistemas críticos: bolsa y finanzas, ambulancias, ferrocarril, etc.
- *SubSector 2*: Fuerzas de ventas o mantenimiento.
- *SubSector 3*: Ejecutivos desplazados con poder de decisión.

Análisis de la encuesta a empresas

Con el fin de extraer las necesidades de movilidad de las empresas, elaboramos un cuestionario, recogido en el Apéndice A, que realizamos a veinte empresas vascas. Antes de comenzar a analizar el resultado de la encuesta, comentar que previamente a la realización del cuestionario, realizamos una presentación a las empresas sobre qué son las soluciones de movilidad y los beneficios que podrían aportar.

Como resultado de la encuesta podemos extraer, brevemente, las siguientes conclusiones:

- Las empresas con más de 50 empleados, donde la mayoría se encuentran desplazados fuera de la oficina, son los más susceptibles a incorporar una solución de movilidad.
- La vía de comunicación entre clientes y empresa más utilizada es la comunicación directa, seguida del teléfono.

- La vía de comunicación entre los proveedores y empresa más utilizada es el teléfono, seguida por el correo electrónico.
- La vía de comunicación entre empleados y empresa más utilizada es el teléfono.
- A las empresas les gustaría mejorar, principalmente mediante las soluciones de movilidad, el modo de comunicación con sus empleados.
- Con respecto al equipamiento tecnológico de la empresa, el ordenador es el dispositivo más extendido, seguido del teléfono móvil. Sólo aquellas empresas con un gran número de empleados disponen de alguna PDA, ya que principalmente, con el software incorporado en ellos no les aporta un valor añadido. Después de la presentación realizada a las empresas, la mayoría estaría dispuesta a adquirir PDAs.
- Cada vez son más las empresas que disponen de una red WLAN en sus instalaciones debido a su facilidad despliegue.
- La mayoría de las empresas considera que los factores más importantes a valorar a la hora de adquirir una solución de movilidad son la reducción de costes (consiguiendo una mayor productividad de los empleados), la facilidad de uso, mantenimiento e integración con el resto de procesos y, por supuesto, el precio.
- Las actividades de su negocio en las que aplicarían una solución de movilidad serían fuerza de ventas, fuerza de trabajo móvil y transporte y logística.
- Para los empleados en movilidad de la empresa, las funcionalidades que incorporarían en una solución de movilidad son principalmente, el acceso y sincronización de datos que residen en los servidores de la empresa y el acceso y la integración con los servicios de la empresa (por ejemplo, el calendario).

A partir de estos resultados, podemos extraer que una solución de movilidad debería potenciar la comunicación entre empresa y empleado y, además, se consideran esenciales la facilidad de uso, mantenimiento e integración con el resto de sus procesos, por lo que en el diseño de la solución propuesta en esta tesis deberemos tener en cuenta estos factores (tal y como se recoge en los requisitos arquitectónicos planteados).

Hemos realizado un análisis sobre la funcionalidad ofrecida por las aplicaciones existentes en las empresas dentro de estas actividades y hemos llegado a la conclusión de que estas **aplicaciones susceptibles a ser movilizadas se pueden clasificar en dos tipos**: aquellas que centran su funcionamiento en los datos y aquellas que centran su funcionamiento en los servicios.

```

<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.mose.deusto.es/schema/moseML.xsd" elementFormDefault="qualified"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:mose="http://www.mose.deusto.es/schema/moseML.xsd">
  <element name="mosePolicyML">
    <complexType>
      <sequence maxOccurs="1" minOccurs="1">
        <element name="applicationSettings" maxOccurs="1" minOccurs="1">
          <complexType>
            <sequence>
              <element name="applicationID" type="string" maxOccurs="1" minOccurs="1"/>
              <element name="idleTime" type="time" maxOccurs="1" minOccurs="1"/>
            </sequence>
          </complexType>
        </element>
        <element name="dataConnectionRules" maxOccurs="1" minOccurs="0">
          <complexType>
            <sequence maxOccurs="unbounded" minOccurs="0">
              <element name="table">
                <complexType>
                  <sequence maxOccurs="1" minOccurs="1">
                    <element name="insert" maxOccurs="1" minOccurs="0">
                      <complexType>
                        <attribute name="timeout" type="time" use="required"/>
                      </complexType>
                    </element>
                    <element name="delete" maxOccurs="1" minOccurs="0">
                      <complexType>
                        <attribute name="timeout" type="time" use="required"/>
                      </complexType>
                    </element>
                    <element name="update" maxOccurs="1" minOccurs="0">
                      <complexType>
                        <attribute name="timeout" type="time" use="required"/>
                      </complexType>
                    </element>
                    <element name="noChange" maxOccurs="1" minOccurs="0">
                      <complexType>
                        <attribute name="timeout" type="time" use="required"/>
                      </complexType>
                    </element>
                    <element name="anyChange" maxOccurs="1" minOccurs="0">
                      <complexType>
                        <attribute name="timeout" type="time" use="required"/>
                      </complexType>
                    </element>
                  </sequence>
                  <attribute name="name" type="string" use="required"/>
                </complexType>
              </element>
            </sequence>
            <attribute name="syncMode" use="required">
              <simpleType>
                <restriction base="string">
                  <enumeration value="total"/>
                  <enumeration value="selective"/>
                </restriction>
              </simpleType>
            </attribute>
            <attribute name="defaultTimeout" type="time" use="required"/>
          </complexType>
        </element>
        <element name="serviceConnectionRules" maxOccurs="1" minOccurs="0">
          <complexType>
            <sequence maxOccurs="unbounded" minOccurs="1">
              <element name="service">
                <complexType>
                  <sequence maxOccurs="unbounded" minOccurs="0">
                    <element name="method">
                      <complexType>
                        <attribute name="name" type="string" use="required"/>
                        <attribute name="requestTime" type="time" use="optional"/>
                        <attribute name="responseTime" type="time" use="optional"/>
                        <attribute name="serviceTime" type="time" use="optional"/>
                      </complexType>
                    </element>
                  </sequence>
                  <attribute name="targetUri" type="string" use="required"/>
                  <attribute name="defaultRequestTime" type="time" use="required"/>
                  <attribute name="defaultResponseTime" type="time" use="required"/>
                </complexType>
              </element>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </complexType>
  </element>
</schema>

```

Figura 4.13 Esquema del lenguaje ANE-SML.

- **Tipo de sincronización (SynchronizationType):** especifica el tipo de sincronización deseada por la aplicación.

- **DataConnectionRules**

Esta sección corresponde con el enfoque orientado a datos. Aquí se definen las necesidades de las aplicaciones en referencia a sus requisitos de mantenimiento de los datos que residen en el dispositivo.

Como se ha visto, el mantenimiento de los datos se realiza a través del proceso de sincronización, con lo que sus necesidades se reducen a la especificación de las necesidades de sincronización de los datos. Para especificar dicha necesidad, proponemos modelarla mediante *reglas de sincronización* con diferentes niveles de granularidad.

Hemos definido diversas reglas de sincronización, donde el diseñador especifica las necesidades tanto a alto nivel como a bajo nivel, dependiendo de su criterio. Por ejemplo, se podría especificar a nivel de tabla o bien a nivel de operación realizada en cada tabla. De este modo, se consigue la flexibilidad sea mayor.

Concretamente, los elementos definidos en esta sección, haciendo uso de la base de datos como sistema de almacenamiento, son:

- **Modo de sincronización (syncMode):** Manifiesta el comportamiento de la sincronización. Se dispone de dos alternativas:
 - ▷ *total*: Actualiza todo lo disponible cada vez que se decide sincronizar.
 - ▷ *selective*: Únicamente se sincronizan las tablas que lo demandan según los tiempos establecidos en sus políticas.
- **Tiempo por defecto (defaultTimeout):** Indica el tiempo de espera para la sincronización por defecto. Se aplica a todos aquellos elementos que no tengan un tiempo de sincronización definido explícitamente facilitando de este modo, la labor del diseñador al no tener que especificar por cada tabla un tiempo concreto.
- Por cada tabla (**table**) contenida en la base de datos con necesidad de sincronización, se indica el nombre de la misma a través del atributo "name" y, dependiendo del nivel de profundidad deseado, se pueden definir los tiempos de sincronización en base a las operaciones realizadas sobre dichos datos. Para el caso de una base de datos sería:

Definición del algoritmo

La primera cuestión que tuvimos en cuenta a la hora de diseñar el algoritmo es que su funcionamiento debía girar en torno al tiempo. Las aplicaciones imponían unas restricciones de tiempo a sus peticiones que debían cumplirse por parte de este algoritmo y todas estas peticiones se centraban en el uso de las comunicaciones. Es por ello, que seleccionamos la aplicación de la técnica de planificación del tráfico como base para el desarrollo del algoritmo.

El primer paso fue analizar los **diferentes parámetros** con los que se debería contar a la hora de planificar:

- **Diferentes tipos de tráfico:** a la hora de planificar existirán diferentes tipos de peticiones para el uso de la interfaz de comunicaciones. En esta tesis, estos tipos son dos, aunque podrán incluirse nuevos tipos de datos en un futuro:
 - ▷ *Sincronización de datos:* cuya petición provendrá del módulo de Gestión de datos.
 - ▷ *Envío/recepción de invocaciones a servicios:* cuya petición provendrá del módulo de Gestión de servicios.

Cada tipo tendrá una exigencia de tiempos distinta y requerirá de un tratamiento diferenciado.

- **El estado de los recursos del dispositivo móvil:** otro de los parámetros a tener en cuenta es el estado de los distintos recursos del dispositivo móvil, que podrá influir en las decisiones del propio planificador. En esta tesis, los recursos que hemos tenido en cuenta son:
 - ▷ *La batería:* el estado de la batería influirá en las decisiones tomadas, concretamente, si su estado es crítico, provocará el envío inmediato del tráfico pendiente, puesto que si se esperase a la expiración del tiempo, se agotaría la batería y el tráfico pendiente quedaría en el dispositivo móvil sin ser procesado.
 - ▷ *Las comunicaciones:* el estado de la interfaz de red también influirá en las decisiones, por ejemplo, si no hay cobertura, se procederá a retrasar todo el proceso, con lo que implicará la modificación de las decisiones tomadas.
- **La planificación varía de forma dinámica durante el tiempo de ejecución:** una vez realizada una planificación inicial, ésta se verá afectada durante la ejecución debido

a la solicitud de envío de nuevo tráfico proveniente del usuario. Por ejemplo, una nueva invocación a un servicio con tiempo más restrictivo que las planificadas hasta ahora o un cambio en la base de datos que requiere de una sincronización, provocará la replanificación continua.

- **Las restricciones de tiempo provienen de diversas fuentes:** cada tipo de tráfico dispone de una serie de restricciones de tiempo. Éstas pueden ser definidas:
 - ▷ en tiempo de diseño, a través de la política definida y especificada en el lenguaje ANE-SML para la aplicación.
 - ▷ en tiempo de ejecución, donde el usuario especifica para una solicitud concreta, las necesidades deseadas. De esta forma, se proporciona un mecanismo a través del cual se puede obviar la política para una solicitud concreta.

- **Maximizar el ahorro de energía:** tal y como se vio anteriormente, el algoritmo no sólo persigue el cumplimiento de las necesidades de las aplicaciones, sino que se persigue la gestión óptima de los recursos. Como se vio, este objetivo se iba a centrar en la Gestión inteligente de la energía, buscando el máximo ahorro en el consumo de estos recursos. Para ello, los fabricantes proporcionan software que permite aplicar diferentes técnicas de adaptación sobre el recurso y así poder modificar el estado y el funcionamiento del mismo, situándolo en un estado de bajo consumo que permitirá ahorrar energía, que es el objetivo de esta tesis. Hasta ahora, estas técnicas son utilizadas por el sistema operativo siguiendo una política de ahorro de energía general y predefinida, sin tener en cuenta las necesidades de las aplicaciones, lo cual produciría retrasos y el incumplimiento de las necesidades de las mismas. Si se tienen en cuenta las necesidades de las aplicaciones, se conseguirá por un lado, el cumplimiento de las necesidades de las mismas y, por otro, mayores ahorros de energía.

- **Soporte a diferentes modelos de adaptación de las aplicaciones con respecto a la gestión inteligente:** tal y como especificamos en el requisito operacional RO5, en esta tesis perseguimos dar soporte a diferentes modelos de adaptación. Entre los más utilizados se encuentran:
 - ▷ *Adaptación transparente a la aplicación:* posee la ventaja principal de que la gestión es totalmente transparente para las aplicaciones, no requiriendo de ninguna modificación de las mismas.

- ▷ *Adaptación sensible a la aplicación sin soporte a la adaptación:* en este caso, la aplicación participa en la gestión proporcionando información sobre sus necesidades pero no realiza la adaptación, de ahí que no sea necesario la modificación del código.
- ▷ *Adaptación sensible a la aplicación con soporte a la adaptación:* la aplicación se involucra totalmente en la gestión realizando toda o parte de la adaptación. Su principal inconveniente es la necesidad de diseñar la aplicación de manera explícita para que realice las funciones de adaptación y cualquier modificación implicará cambios en la misma. Su complejidad de diseño, implementación y mantenimiento será muy elevada.

En esta tesis hemos optado por evitar modificar las aplicaciones para permitir el uso de la solución diseñada con cualquier aplicación ya sea previamente existente o una nueva, como sucede en el entorno empresarial. En esta tesis hemos diseñado una **solución híbrida**, es decir, que sea tanto transparente como sensible a la aplicación sin soporte a la adaptación con respecto a la gestión. Aunque como se ha visto, la última sería la aproximación más deseable ya que permite una gestión más eficiente. Como no es posible obligar a toda aplicación a definir sus preferencias, sobre todo si ya está operativa. Para este tipo de aplicación optamos por el modelo transparente a la aplicación, para que se beneficien de manera totalmente transparente de la gestión mediante el uso de una política por defecto.

Concretamente, dentro del diseño para incorporar el modelo transparente a la aplicación, hemos definido una política por defecto que será utilizada para aquellas aplicaciones que no quieren definir ninguna política concreta. Para dar soporte al modelo sensible a la aplicación sin soporte a la adaptación, cada aplicación podrá utilizar el lenguaje ANE-SML para especificar sus necesidades que serán tenidas en cuenta en el algoritmo. De esta manera, no se exige a la aplicación tener que modificar su código para implementar técnicas de adaptación, aspecto impensable para las aplicaciones empresariales.

En segundo lugar, procedimos a analizar **los resultados** que debía proporcionar el algoritmo, éstos eran los siguientes:

- **Planificación del tráfico.**

El resultado más importante que debe proporcionar el algoritmo es una planificación óptima de las comunicaciones que cumpla los requisitos de cada aplicación.

Para ello, debemos tener en cuenta que se deben gestionar diferentes tipos de tráfico, ser consciente del estado de los recursos, que la planificación variará de forma dinámica y que las restricciones de tiempo provendrán de diversas fuentes.

La técnica de planificación de tráfico debe ser aplicada tanto al tráfico entrante como al saliente. Como tráfico entrante consideramos las respuestas a las invocaciones a servicios mientras que como tráfico saliente consideramos tanto la sincronización de datos (lo consideramos saliente ya que este proceso es iniciado siempre por el dispositivo móvil tal y como se especifica en el apartado 4.4.1) como el envío de las invocaciones a servicios realizadas por el usuario.

- **Gestión inteligente de la energía basada en técnicas de adaptación a aplicar a los recursos.**

Centrándose en el algoritmo, éste debe proporcionar como resultado el conjunto de técnicas a aplicar a los recursos, que podrán ser modificadas, al igual que la planificación, de forma dinámica durante la ejecución. La selección de las técnicas estará orientada a maximizar el ahorro de energía.

Esta tesis se ha centrado en gestionar un único recurso, la interfaz WLAN, pero podría ser escalable a otros recursos como la CPU, disco duro, pantalla, etc. La selección de este recurso la realizamos teniendo en cuenta que el algoritmo se centraba en la gestión de las comunicaciones de las diversas aplicaciones y que estas comunicaciones en los dispositivos móviles se realizan normalmente a través de este tipo de interfaz inalámbrica.

La técnica de adaptación utilizada sobre la interfaz inalámbrica consiste en hacer uso de los estados de bajo consumo de dicha interfaz cuando durante un período de tiempo determinado no se va a hacer uso de la misma. Ese período de tiempo viene calculado por la planificación del tráfico, aunque variará de forma dinámica.

El tercer paso, una vez analizados los parámetros de entrada y los resultados a producir, fue **diseñar** en profundidad **el algoritmo**. En primer lugar, vamos a profundizar en el diseño de la planificación del tráfico y, a continuación, en la gestión de los recursos.

Arquitectura del algoritmo

Para definir el funcionamiento del algoritmo, hicimos en primer lugar una descomposición en módulos de su lógica, que se muestra en la Figura 4.18.

4.4.2.2 Gestión de servicios

Este módulo implementa el enfoque orientado a servicios. Para llevar a cabo la gestión de los servicios hemos tenido en cuenta los requisitos arquitectónicos, definidos en el apartado 4.1, además de los requisitos funcionales propios de este módulo. Al igual que antes, procederemos a dividir el apartado en dos partes: una, centrada en el cliente y otra, en el servidor.

Cliente MOSE

Siguiendo el modelo de componentes propuesto para el enfoque orientado a servicios, los tres componentes a diseñar en la parte cliente son:

- Sistema de almacenamiento.
- Mecanismo de almacenamiento de invocaciones.
- Mecanismo de envío/recepción de invocaciones.

Sistema de almacenamiento

Este componente pretende almacenar las invocaciones cuando no hay conectividad para, posteriormente, enviar la invocación. Además, deberá almacenar las invocaciones incluso cuando haya conectividad para poder relacionar las peticiones con las respuestas, puesto que dichas respuestas no tienen porqué llegar instantáneamente, ya que puede producirse la desconexión cuando se está esperando la respuesta.

El primer paso que realizamos para diseñar este componente fue crear una representación genérica de las invocaciones a los servicios para cualquier tipo de servicio y, posteriormente, diseñar la gestión de su ciclo de vida.

Para su diseño hemos definido las siguientes clases representadas en el diagrama de clases de la Figura 4.38.

A continuación detallamos la función de cada una de ellas en base al área al que contribuyen.

Representación de las invocaciones

Los objetos ServiceRequest representan las invocaciones que el usuario quiere realizar a un servicio externo concreto. Los parámetros necesarios para llevar a cabo una invocación a un servicio son los siguientes:

- **TargetUri:** ubicación del servicio en la red.
- **MethodName:** nombre del método que se quiere invocar.
- **Parameters:** lista de parámetros que recibe el servicio.
- **Location:** ubicación de la descripción del servicio.
- **ServiceName:** nombre del servicio concreto que se quiere invocar.
- **ResponseType:** tipo de la respuesta (necesario para la deserialización).
- **Response:** respuesta recibida a la invocación.

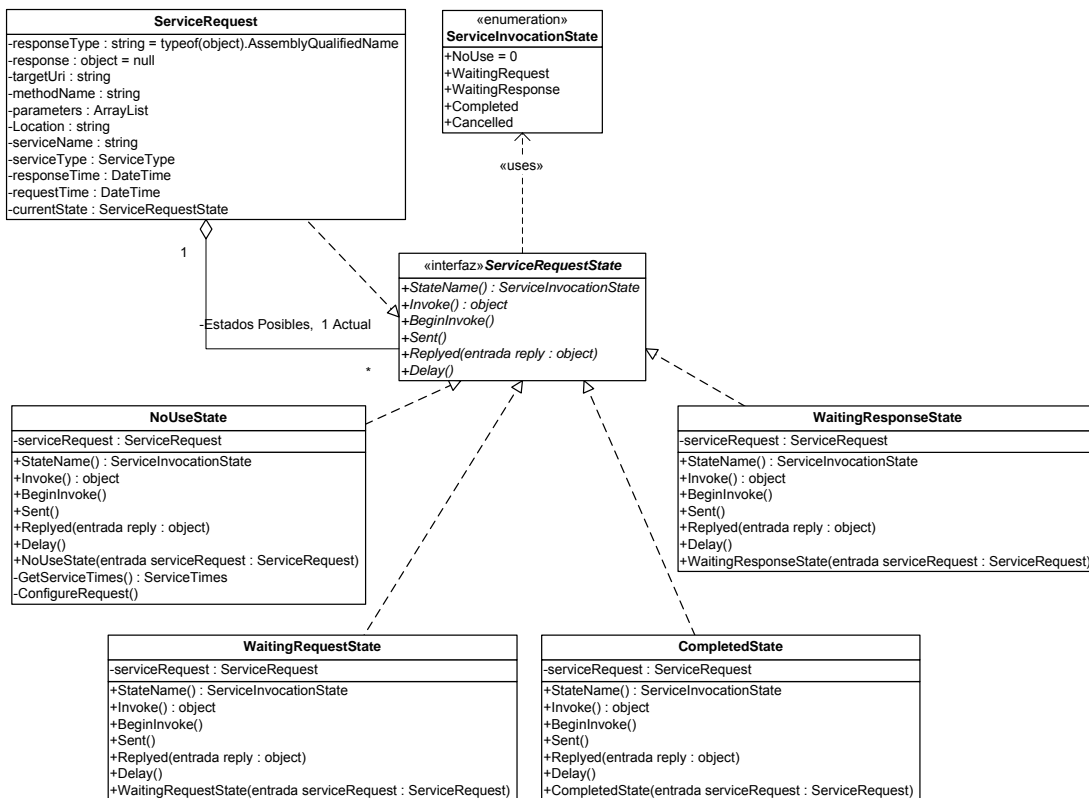


Figura 4.38 Diagrama de clases del sistema de almacenamiento.

Gestión del ciclo de vida de las invocaciones

Una invocación puede pasar a través de diversos estados, éstos son:

- **NoUse:** una invocación permanecerá en este estado desde que el objeto de invocación se crea hasta que se invoca.

- **WaitingRequest:** la petición de invocación a un servicio pasa a este estado cuando se solicita la invocación de la petición a través de la plataforma. Este estado se ha creado explícitamente para definir una situación transitoria que necesitará el núcleo de la plataforma.
- **WaitingResponse:** las peticiones se encuentran en este estado cuando están esperando a la respuesta del servidor.
- **Completed:** cuando se recibe la respuesta, la petición ya está completa.

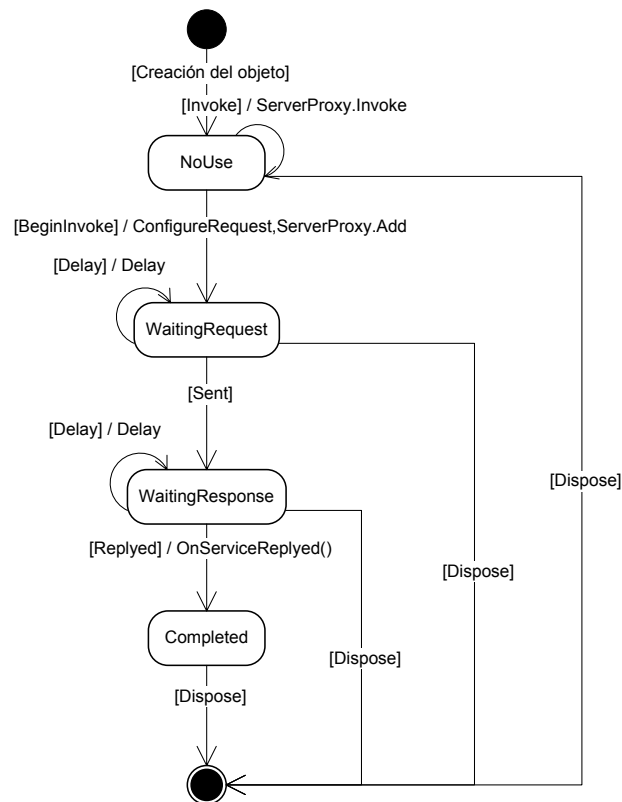


Figura 4.39 Diagrama de estados de las invocaciones a servicios.

Decidimos diseñar el comportamiento de las invocaciones siguiendo el patrón de máquina de estados, ya que se adapta perfectamente a los objetivos de esta clase. El diagrama de estados que detalla dicho comportamiento se muestra en la Figura 4.39.

Mecanismo de almacenamiento de invocaciones

Este componente proporciona a las aplicaciones una interfaz para solicitar la invocación a un servicio. Para ello, hemos definido la clase `ApplicationManager`, cuyos objetivos son: actuar como fachada de la plataforma de cara al cliente y coordinar los diferentes componentes que

la forman. En la Figura 4.40, se muestra el diagrama de clases correspondiente a este componente.

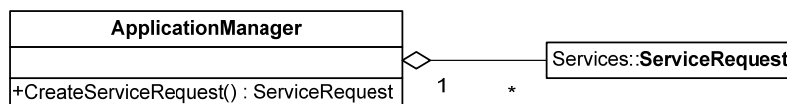


Figura 4.40 Diagrama de clases del mecanismo de almacenamiento de invocaciones.

ApplicationManager proporciona un único punto de entrada a las funciones del middleware (siguiendo el patrón de diseño Singleton y Fachada) para todas las aplicaciones, de esta manera, el ApplicationManager gestiona las invocaciones de múltiples aplicaciones simultáneamente, con lo que se cumple con el requisito RO6.

Para permitir a una aplicación solicitar la invocación de un servicio, esta clase dispone de un método CreateServiceRequest que almacena la petición en un objeto ServiceRequest y lo añade a la lista de peticiones realizadas previamente.

Mecanismo de envío y recepción de invocaciones

Una vez introducidas las invocaciones en el sistema de almacenamiento, será necesario un mecanismo que se encargue de:

- Proceder a la invocación de las peticiones almacenadas cuando haya conexión.
- Recoger de las respuestas a dichas invocaciones.

El conjunto de clases que componen este mecanismo se encuentran reflejadas en la Figura 4.41.

La clase ServerProxy es la encargada de enviar y recibir las peticiones de invocación (ServiceRequest). A la hora de enviar, comprobará si la conexión es posible, si es así, obtendrá todas las invocaciones que se encuentran esperando a ser enviadas y las cambiará de estado a *WaitingResponse*. Además, aprovechará que la conexión está abierta para recibir las respuestas que serán adjuntadas a su invocación correspondiente y se les procederá a cambiar de estado a *Completed*.

El ServerProxy hará uso de la clase ServiceGateway que será la encargada de realizar la comunicación con el servidor. Se encargará de implementar el mecanismo de comunicación usado por el servidor. De esta manera, si dicho mecanismo cambia, sólo habrá que modificar esta clase con lo que se consigue que la solución sea transparente al mecanismo de comunicación utilizado.

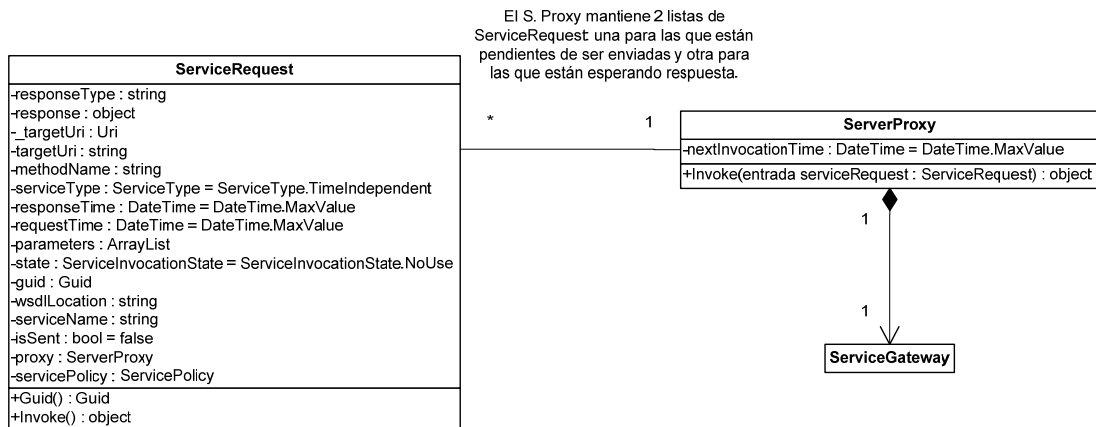


Figura 4.41 Diagrama de clases del mecanismo de envío y recepción de invocaciones.

Servidor MOSE

En la parte servidora, tal y como detallamos en el apartado 4.3.1.2, en el enfoque orientado a servicios, se pueden incorporar componentes adicionales (en base a la funcionalidad que se desee proporcionar). Si no se desea añadir ningún componente adicional, el enfoque orientado a servicios sólo requeriría de la parte cliente, ya explicada.

En esta tesis, hemos considerado esencial garantizar que las aplicaciones puedan recibir las respuestas aún cuando estén en un período de desconexión, por lo que haremos uso de un sistema de almacenamiento intermedio, es decir, un servidor Proxy. En él, también podría residir el servidor de sincronización del enfoque orientado a datos.

Otro de los aspectos que hemos tenido en cuenta son los requisitos RA1 y RA7, es decir, que debería ser integrable con otras soluciones y escalable. Esto, dentro de este módulo, lo hemos traducido en garantizar **que los servicios serán independientes de la tecnología subyacente**: es decir, que los servicios gestionados por la solución no estén ligados a la tecnología de los mismos (Servicios web, EJB, RMI, CORBA, etc.). Por lo tanto, un dispositivo móvil puede acceder a cualquier servicio empresarial puesto que se realiza la adaptación a la tecnología final en el servidor Proxy.

Por lo tanto, los objetivos perseguidos en el diseño del servidor proxy serán:

- Gestión de la respuesta enviada a las aplicaciones de un dispositivo móvil.
- Gestión transparente de los servicios.

Gestión transparente de los servicios

Como hemos mencionado, uno de los objetivos perseguidos es poder gestionar cualquier invocación a un servicio remoto independientemente de su naturaleza. Para ello, hemos optado, en primer lugar, por realizar un mecanismo genérico de representación de las invocaciones independiente del tipo de servicio (a través de la clase `ServiceRequest`). En segundo lugar, hemos optado por el uso de una pasarela de servicios en el servidor proxy, que reciba estas peticiones genéricas y que, finalmente, realice las invocaciones a los servicios remotos.

Pero, ¿por qué utilizar una pasarela cuando se podrían invocar directamente, con el consiguiente ahorro de tiempo? Si analizamos en profundidad el comportamiento deseado de la solución, llegamos a la conclusión de que esa opción no es del todo válida para la problemática planteada. Hemos detectado varias situaciones en las que resulta beneficioso, sino imprescindible, el uso de un elemento intermedio (la pasarela) para solucionar problemas derivados del tipo de sistemas finales a los que va dirigida el middleware MOSE, los cuales suelen tener limitaciones, sobre todo, de recursos de computación.

El uso de la pasarela queda entonces justificado en los siguientes puntos:

- **Soporte de servicios heterogéneos**

Gracias a la pasarela, será posible la invocación a servicios cuyo protocolo o mecanismo de comunicación no esté soportado o suponga una carga excesiva para un dispositivo móvil. Los servicios empresariales serán de diversa naturaleza y puede que no todos los dispositivos sean capaces de invocarlos directamente. Por ejemplo, una aplicación empresarial basada en CORBA o RMI no se podrá invocar desde un dispositivo móvil. Sin embargo, la plataforma MOSE proporciona un mecanismo para que los servicios que estén implantados en la empresa sean accesibles y se puedan utilizar indirectamente desde las aplicaciones movilizadas.

- **Necesidad de funcionamiento asíncrono**

La mayoría de los servicios soporta un mecanismo de funcionamiento síncrono que no permitiría que el dispositivo se desconectara mientras espera la respuesta, aspecto clave para una aplicación móvil.

- **Necesidad de funcionamiento pull**

La mayoría de los mecanismos de comunicación con servicios remotos suponen que los dispositivos son direccionables a nivel de red, es decir, que el servidor puede invocar directamente al cliente. Como se vio en el apartado 4.4.1, esto no siempre es posible,

Protocolo de comunicación

El siguiente paso en el diseño fue seleccionar qué protocolo de comunicación utilizar. Analizamos diversas alternativas que son expuestas a continuación.

- **Invocaciones a servicios mediante el motor de sincronización**

En esta alternativa, el mecanismo empleado para transportar las invocaciones a los servicios al cliente, sería el motor de sincronización. Para ello, la información residente en la invocación del servicio se mapea sobre una tabla de la base de datos, y cuando el middleware estime necesario se procede a la sincronización de la misma. De esta forma, toda la información sobre las invocaciones se encontraría en la base de datos consolidada en el servidor.

A continuación, una vez que el servidor termina la sincronización, los datos se transforman al formato original de la invocación del servicio para invocar realmente al servicio remoto. Por lo tanto, el servidor de sincronización es el elemento que desarrolla la función de Proxy invocando al servicio. No obstante, la comunicación entre el servidor de sincronización y los servidores remotos se supone siempre activa, es decir, no hay problemas de conectividad.

Una vez recibida la respuesta en el servidor de sincronización procedente del proveedor del servicio, para poder retornarla al cliente solicitante, se tiene que ajustar la respuesta a la tabla dedicada a los servicios. Por lo tanto, en un nuevo proceso de sincronización, los datos se almacenan en la base de datos del cliente. Lógicamente, los datos se entregarán a la aplicación movilizadora solicitante.

La principal ventaja de este enfoque es que se podría reutilizar en cierta medida el mecanismo de sincronización que ya se utiliza para el enfoque orientado a datos. Sin embargo, hemos descartado este enfoque por las siguientes razones:

- Se obligaría a utilizar una base de datos, tanto en el cliente como en el servidor. Con esto se excluyen de la solución posibles aplicaciones que no puedan o quieran utilizar bases de datos.
- La integración de las llamadas a servicios en la base de datos cliente implicaría que las aplicaciones deben tener obligatoriamente una serie de tablas específicas de control y que la solución debería conocer. Que cada aplicación tenga sus invocaciones en su base de datos dificulta, por un lado, la gestión global por parte del middleware MOSE, ya que tendría que acceder a múltiples bases de datos. A su vez, si se unifican en una única base de datos, la plataforma MOSE requeriría

siempre de un motor de base de datos en el cliente, lo cual no cumple con el requisito de ser lo más ligera posible.

- La integración de las invocaciones en la sincronización implica que la arquitectura del servidor se complica, ya que cada base de datos consolidada debería ser capaz de invocar a dichos servicios, con lo que por cada una sería necesario replicar la lógica de invocación de los servicios o, en el mejor de los casos, redireccionar a un componente que sepa realizarla con la consiguiente sobrecarga que supondría.
- Las características de las invocaciones hace que resulte relativamente sencillo integrarlas en la base de datos. Sin embargo, la necesidad de configurar un proceso de sincronización para estas tablas no compensa ni en tiempo (se tardaría más en configurar cada aplicación) ni en rendimiento (un proceso de sincronización es más costoso computacionalmente).

Por estas razones hemos optado por un enfoque más abierto y flexible para el tratamiento de las invocaciones a los servicios que detallamos a continuación.

- **Invocaciones a servicios mediante servicios web**

A diferencia de la alternativa anterior, las sucesivas invocaciones a servicios procedentes de las aplicaciones residirían en el propio middleware MOSE, con lo que se facilita la gestión de las mismas. El almacenamiento es persistente y el borrado de las invocaciones y las respuestas correspondientes a los servicios solicitados se lleva a cabo cuando los resultados son entregados a la aplicación demandante.

Cuando el dispositivo móvil recupera la conectividad, se transmiten al servidor Proxy de la plataforma MOSE todas las invocaciones de servicios solicitadas y se reciben las respuestas de invocaciones precedentes. El servidor Proxy de la plataforma MOSE realiza la adaptación correspondiente al tipo de servicio empresarial.

Los resultados de los recursos empresariales se almacenan en el servidor Proxy de la plataforma MOSE hasta que los clientes lo soliciten. Lógicamente, es el dispositivo móvil quien comprueba si las respuestas de los servicios están disponibles.

Aunque similar en concepto, este enfoque en relación al anterior, proporciona una serie de ventajas:

- El uso de un servidor intermedio que reciba las invocaciones permite la separación total de la base de datos empresarial.

- No depender de una base de datos en el cliente hace que la plataforma sea más adaptable a cualquier tipo de sistema.
- El mecanismo por el cual se envíen y reciban las invocaciones entre los dispositivos y el servidor puede ser optimizado para cada caso, ya que es independiente del resto de módulos.

En base a estas reflexiones, hemos decidido implementar **la comunicación mediante servicios web por HTTP**, debido a los beneficios que plantean [Alonso+04] [Mnaouer+04] [Muñoz04] [Carbonell+03] como la sencillez y adaptación a dispositivos limitados y el soporte por la industria, basado en estándares (cumpliendo así con el requisito RA4), etc. Concretamente, haremos uso de un servicio web que servirá como punto de entrada de la pasarela. Le hemos denominado *ServiceGateway* y dispone de un método principal, que será el encargado de recibir las invocaciones y encolarlas para su posterior invocación. Además, consultará las respuestas de invocaciones procedentes de las aplicaciones que recibe como parámetro.

Hemos optado por concentrar las dos operaciones en una (solicitud de invocación y respuesta) para ahorrar conexiones y optimizar el tráfico entre cliente y servidor. De hecho, la operación está justificada, ya que siempre que se invoca hay que consultar las respuestas y viceversa, con lo que no tiene sentido realizar dos conexiones separadas.

Procesador de invocaciones

El procesador ha sido diseñado mediante una única clase y contiene dos colas: una, para las peticiones en espera de ser atendidas y otra, con las respuestas recibidas que representan al sistema de almacenamiento utilizado en el servidor Proxy.

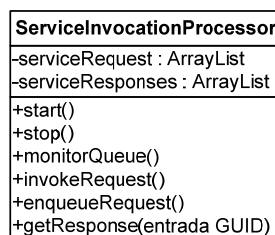


Figura 4.43 Diagrama de clases del procesador de invocaciones.

Esta clase es la encargada de dar servicio a las invocaciones que se encuentran encoladas. Sondeará la cola de invocaciones y en función del tipo, creará procesos hijos que sean capaces de despacharlas. El número de procesos a crear simultáneamente será proporcional a la capacidad de procesamiento que tenga el servidor donde se instale la solución.

Adaptadores de invocación

Las invocaciones pueden ser de distinta naturaleza en función del sistema empresarial final, con lo que será necesario adaptar la representación de una invocación utilizada en MOSE a una invocación real. Serán necesarias distintas implementaciones en función del tipo de servicio.

La pasarela de servicios es la encargada de llevar a cabo las invocaciones reales a los servicios empresariales, es decir, si el servicio final es un servicio web, es la que debe conocer el funcionamiento de los mecanismos de comunicación de los servicios web. Para ello, hemos definido en la pasarela adaptadores específicos para cada tipo de servicio. De tal manera que conozca cuál es el mecanismo de comunicación a utilizar para cada tipo de servicio.

Por otro lado, por cada tipo de servicio, no se pueden conocer de antemano qué servicios se invocarán, ya que esto limitaría en la escalabilidad del sistema. Si se conociesen de antemano los servicios que se van a usar, el diseño se simplificaría al poder guardar en cada adaptador de la pasarela, la definición detallada de cada servicio, con sus métodos, atributos, etc. Por ejemplo, con servicios web se materializaría con el almacenamiento del fichero WSDL en tiempo de diseño. Pero, como hemos mencionado, esto limitaría la escalabilidad de la solución, con lo que buscamos que la invocación a los servicios finales sea realizada de manera dinámica para conseguir un rendimiento óptimo.

Resumiendo, la invocación dinámica de un servicio consiste en conseguir comunicarse con él sin la existencia previa de una descripción del mismo. Por lo tanto, la solución en el servidor deberá crear dinámicamente un *proxy o representante del servicio* a invocar a partir de la descripción del mismo obtenida durante el tiempo de ejecución. En el caso de los servicios web, se debería crear un Proxy basándose en la definición WSDL del mismo.

Por lo tanto, la pasarela debe partir de la definición del servicio final, construir dinámicamente un Proxy para comunicarse con él y adaptar los parámetros para llevar a cabo la invocación y recepción de la respuesta.

Tipos de datos de las invocaciones

Uno de los problemas de la invocación de servicios a través del middleware son los tipos de datos. Conocer la naturaleza de un servicio, sus parámetros, métodos y tipos de datos de entrada y salida es responsabilidad de la aplicación movilizada, así como de la pasarela de servicios. Sin embargo, el middleware MOSE cliente no tiene por qué conocer el servicio final que va a ser invocado, simplemente, tiene la responsabilidad de gestionar la representación de dicha invocación.

Por lo tanto, ¿cómo garantizar que los parámetros viajan correctamente desde la aplicación hasta el servicio final pasando a través de MOSE? La clave está en la manera de representar los datos. La solución planteada es utilizar la *serialización*, es decir, obtener una representación neutra del tipo de datos a transportar, por ejemplo, el proceso de envío de un parámetro de un tipo concreto, por ejemplo, el objeto de tipo “Persona”. Hemos optado por representarlo en XML al ser el lenguaje más ampliamente utilizado para este fin.

A continuación, detallamos el diseño de los adaptadores. Lo primero que tuvimos en cuenta fue la escalabilidad (RA7), pudiéndose añadir nuevos adaptadores en función de las necesidades futuras.

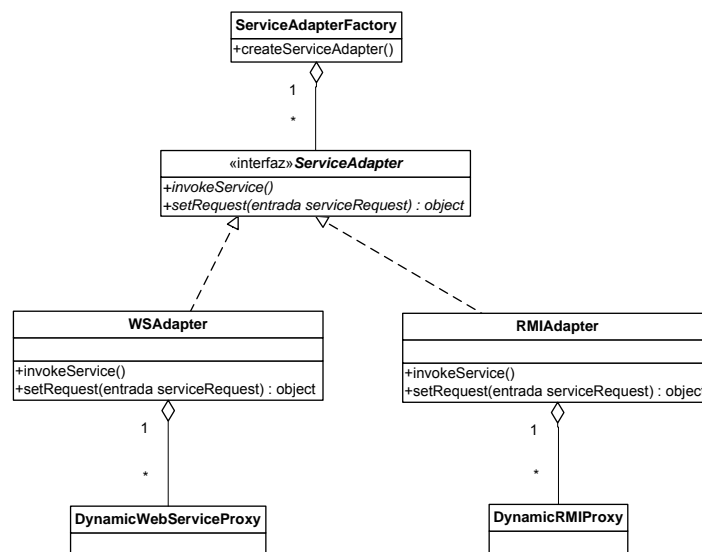


Figura 4.44 Diagrama de clases de los Adaptadores de invocación.

Como se puede observar en la Figura 4.44, hemos utilizado el patrón Factory para obtener el adaptador más conveniente en función del tipo de servicio. Todos los adaptadores implementan la interfaz `ServiceAdapter` que permite invocar cualquier tipo de servicio.

Las clases definidas son:

- **ServiceAdapterFactory**

Los servicios que van a ser invocados pueden ser de diferentes naturalezas. Para invocar cada tipo de servicio hará falta un adaptador específico. Esta clase será la encargada de proporcionar dichos adaptadores para cada tipo de servicio, actuando de factoría de los mismos.

- **ServiceAdapter**

Esta interfaz contempla los métodos que debe tener un adaptador para invocar un servicio.

- **WSAdapter y RMIAdapter**

Corresponden a implementaciones concretas para servicios web y RMI, respectivamente.

- **DynamicWebServiceProxy**

Esta clase será la encargada de crear dinámicamente un Proxy de un servicio web a invocar a partir de la descripción del mismo.

- **DynamicRMIProxy**

Esta clase será la encargada de crear dinámicamente un Proxy de un servicio RMI a invocar a partir de la descripción del mismo.

El funcionamiento de este componente estará guiado por las siguientes actividades principales:

- Determinar el servicio remoto al que hay que invocar a través del campo URI y methodName de la clase invocación.
- Adaptación de los parámetros a un tipo reconocible por el servicio final.
- Realización de la invocación. Este proceso será normalmente automático, para lo cual se requerirá la creación de un Proxy dinámico que invoque el servicio final.
- Recepción de la respuesta y adaptación a un tipo reconocible por la plataforma cliente.

4.4.3 Núcleo

Durante el diseño hemos tenido en cuenta los requisitos operacionales y los requisitos arquitectónicos definidos en el apartado 4.1. Para describir el diseño detallado realizado, lo vamos a dividir en los componentes definidos previamente que se muestran en la Figura 4.45.

En primer lugar, definimos una clase `ApplicationManager` que será el punto de entrada del gestor de aplicaciones, además de servir como punto de entrada al middleware y de coordinación de los diferentes componentes que lo forman. En la Figura 4.47, se incluye un diagrama de clases general sobre este elemento.

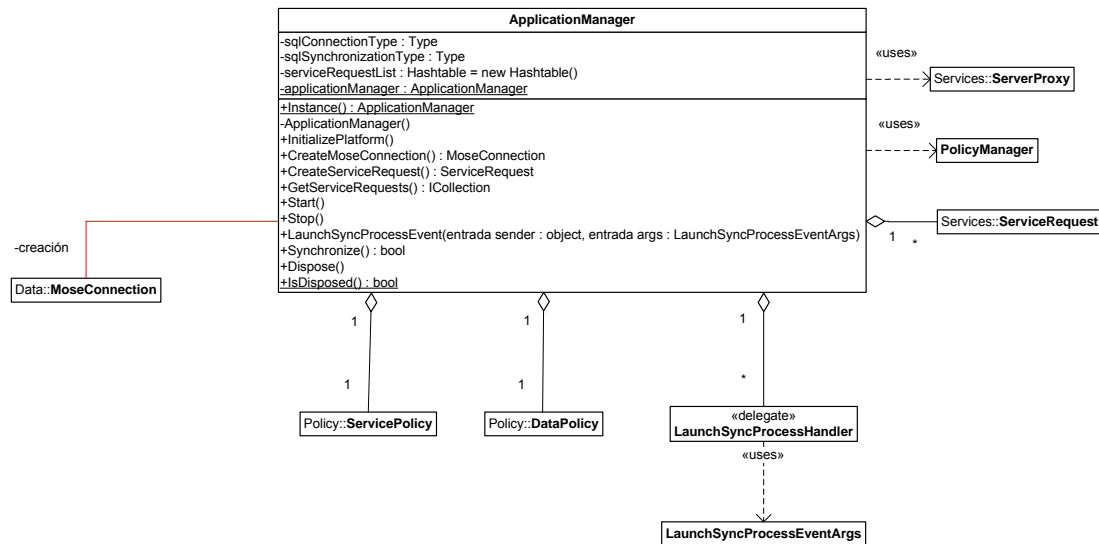


Figura 4.47 Diagrama de clases del gestor de aplicaciones.

Esta clase recibe el evento “LaunchSyncProcess” enviado por el gestor central para que inicie el proceso de comunicaciones. El `ApplicationManager` a su vez notificará a cada uno de los componentes para que procedan a enviar su tráfico pendiente.

A continuación, definimos los diferentes componentes del gestor de aplicaciones.

Gestor de políticas de datos

Este componente es el encargado de gestionar la parte de sincronización de datos. Este componente tiene como cometido calcular cuál será el siguiente tiempo de sincronización en base a los tiempos recogidos en la política o los tiempos especificados en tiempo de ejecución por el usuario y deberá notificarlo al gestor central. Esta lógica se ha materializado en un conjunto de clases que se muestran en el siguiente diagrama de clases general.

Describiendo cada una de estas clases:

- *DataPolicy*: esta clase es clave en este componente debido a que implementa la lógica del mismo. Para llevar a cabo su funcionamiento, se basa en una serie de clases que se describen a continuación.

- *DataRulesHandler*: esta clase recoge el conjunto de reglas de sincronización definidas en el fichero de políticas para la aplicación y que DataPolicy usará para obtener información sobre las necesidades de las aplicaciones.
- *DataRule*: esta clase representa una regla concreta de sincronización.
- *SyncMode*: es una clase de apoyo que define los tipos de sincronización usados.

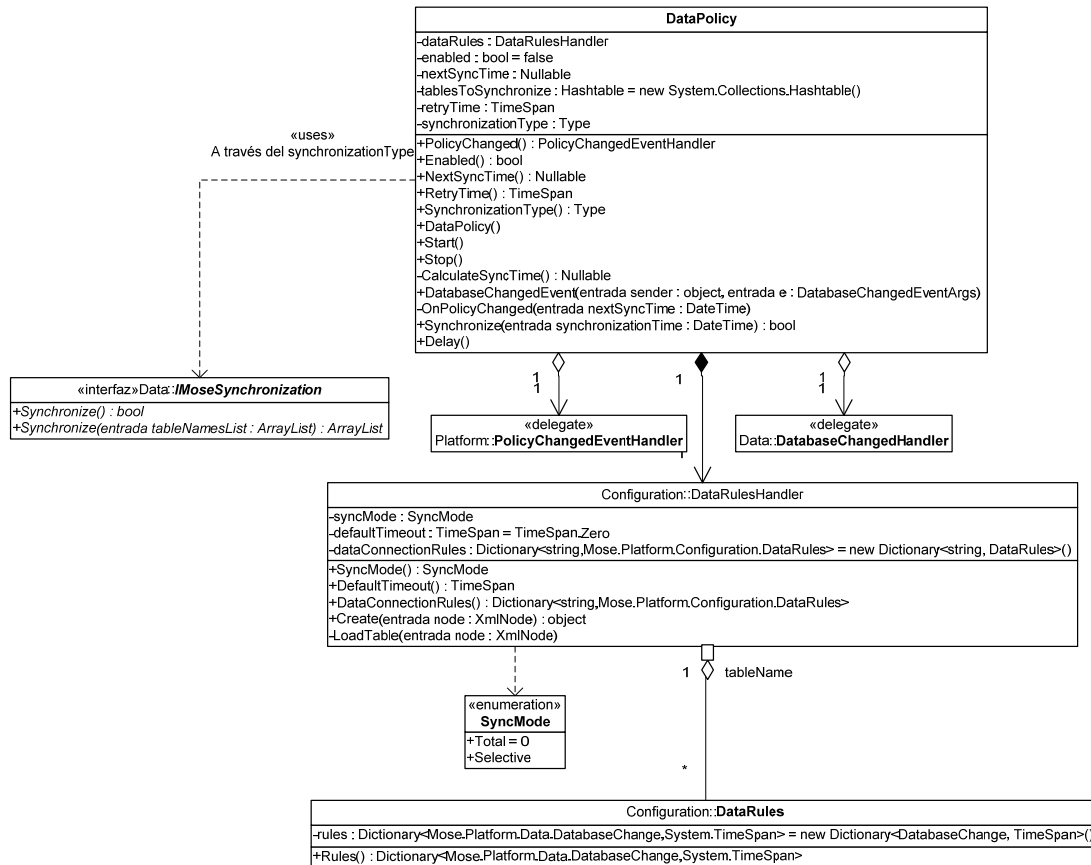


Figura 4.48 Diagrama de clases del gestor de políticas de datos.

Además, su funcionamiento, al igual que el gestor central, se basa en el uso de eventos, este componente gestiona dos tipos de eventos:

- *DatabaseChanged*: este evento será enviado por el módulo de gestión de datos del “Soporte a la operación online/offline” al gestor de aplicaciones para indicar que se ha producido un cambio en la base de datos y que DataPolicy pueda actuar en consecuencia en función del algoritmo definido en el modelo teórico.
- *PolicyChanged*: será enviado por este componente al gestor central para notificar sobre los cambios en sus necesidades de tiempo.

Gestor de políticas de servicios

Es el encargado de gestionar la parte de servicios. Este componente tiene como cometido calcular el tiempo en el que deberán ser invocados los servicios para que se cumplan los tiempos que están definidos en cada petición, además de notificarlo al gestor central.

A la hora de afrontar el diseño de este componente, pensamos en diseñarlo como el componente de gestión de políticas de datos, es decir, utilizar una clase central responsable del cálculo de los tiempos y el control de los cambios en los servicios además de hacer uso de la reglas de sincronización. Sin embargo, los servicios son algo más complejos de gestionar ya que, por una parte, permiten un mayor nivel de configuración y, por otra parte, las peticiones son independientes entre sí, además de pasar por una serie de estados.

Si la responsabilidad de gestión recayera sobre una única clase, el diseño se complicaría y surgiría el peligro de crear un cuello de botella. Por ello, hemos optado por el enfoque en el que sean las propias invocaciones las responsables de sus estados y existirá una clase *ServicePolicy* que se encargará de la gestión de la reglas extraídas de la política y a la que las invocaciones consultarán cuando necesiten un dato de la política. Además, existirá una clase independiente *ServerProxy* que será la que realice el cálculo del tiempo de invocación de todas las invocaciones. Por tanto, la lógica la hemos dividido en tres funcionalidades:

- Gestión de las reglas de envío/recepción de invocaciones contenidas en la política.
- Gestión del ciclo de vida de las invocaciones.
- Cálculo y notificación del siguiente tiempo de invocación.

A continuación vamos a proceder a presentar el diseño detallado de estas tres funcionalidades:

- **Gestión de la reglas de envío/recepción de las invocaciones.**

En la Figura 4.49 se muestra un diagrama de clases general de esta funcionalidad.

La función de cada una de estas clases es:

- *ServicePolicy*: esta clase es la encargada de proporcionar el acceso a la reglas sobre los servicios recogidos en la política.
- *ServiceRulesHandler*: se encarga de gestionar las diferentes reglas.
- *ServiceRules*: representa las reglas de envío/recepción para un servicio completo identificado por su targetURI.

- *ServiceTimes*: recoge los diferentes tiempos que se pueden definir a nivel de servicio.

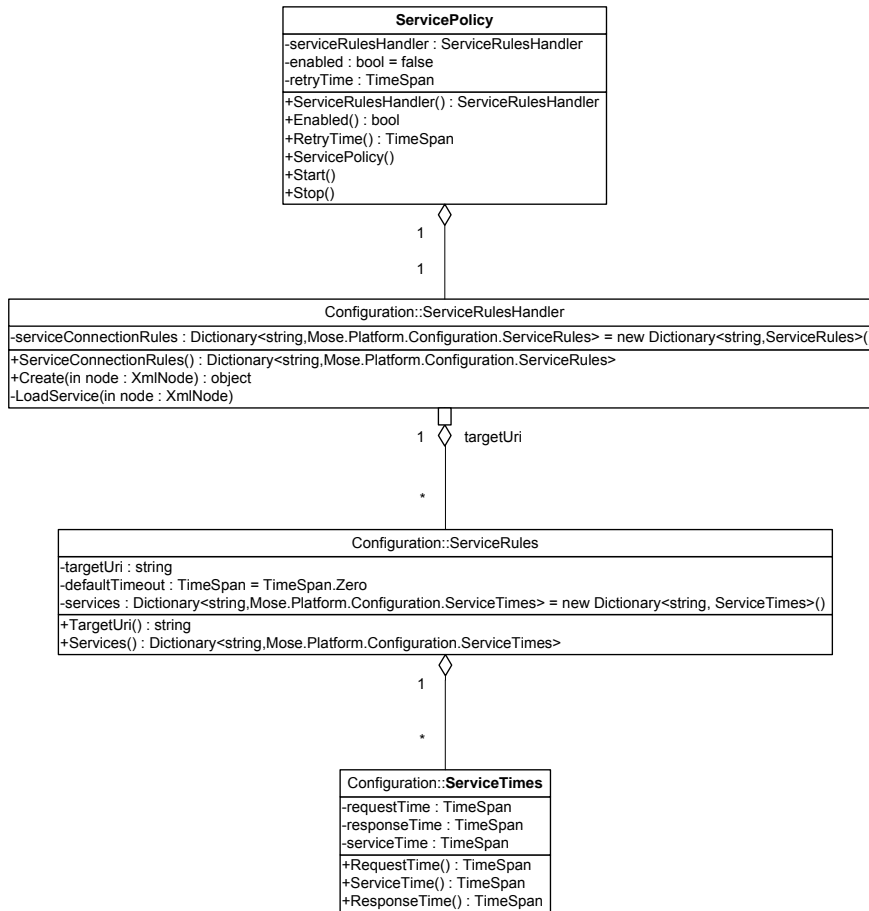


Figura 4.49 Diagrama de clases de la gestión de las invocaciones.

- **Gestión del ciclo de vida de las invocaciones.**

En el diseño, el primer aspecto a afrontar era cómo representar las invocaciones a los servicios y cómo gestionar su ciclo de vida. Para ello, tomamos como base la representación genérica y el ciclo de vida definido en el componente de gestión de servicios del módulo “Soporte a la operación online/offline”, puesto que no tendría sentido volver a definir nuevas clases que realicen la misma funcionalidad. De ahí que el diagrama de clases será el mostrado en la Figura 4.50.

A continuación, detallamos aquellas clases que son más representativas:

- *ServiceRequest*: Esta clase representa las invocaciones que el usuario quiere realizar a un servicio externo concreto. Además, para su funcionamiento dentro de la plataforma es necesario que cada invocación tenga definidos:

- ▷ **Guid:** identificador único de cada petición de invocación (generado internamente).
- ▷ **ServiceType:** tipo de invocación a servicio en base al comportamiento de las políticas en función de la clasificación de las invocaciones a servicios propuesta en el modelo teórico (ver apartado 4.3.2.2).
- ▷ **RequestTime, ResponseTime:** tiempos de invocación y respuesta, que se reconfiguran automáticamente en función del estado por el que pase la invocación y del tipo de la misma.

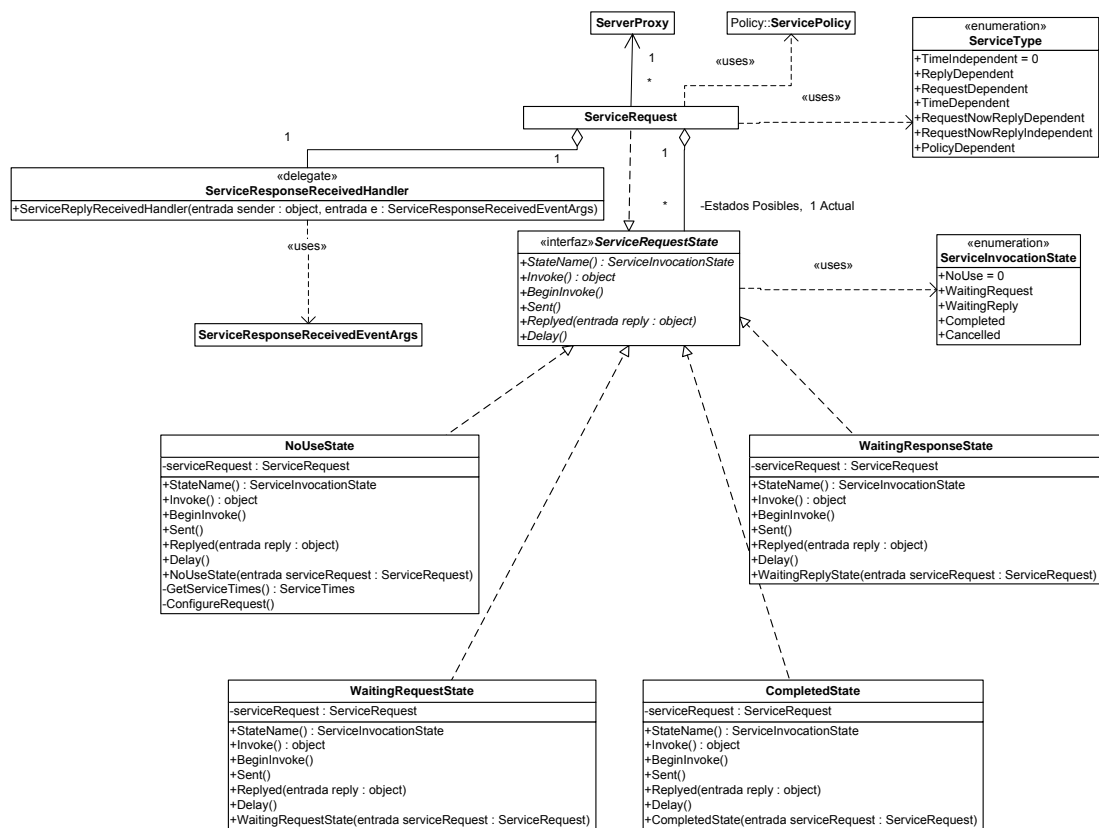


Figura 4.50 Diagrama de clases de la gestión del ciclo de vida de las invocaciones.

Además de los atributos básicos, cada invocación mantiene referencias a los componentes que necesita para gestionar su estado, es decir, el `ServicePolicy`, que le indica la configuración de los tiempos, y al `ServerProxy`, que es el que permite llevar a cabo la invocación hacia el servidor de las invocaciones.

- **ServiceRequestState:** esta clase representa los estados por los que pasa una invocación (ver apartado 4.4.2.2). Decidimos diseñar el comportamiento de las invocaciones siguiendo el patrón de máquina de estados, ya que se adapta perfectamente a los

objetivos de esta clase. El diagrama de estados que detalla dicho comportamiento se muestra en la siguiente figura:

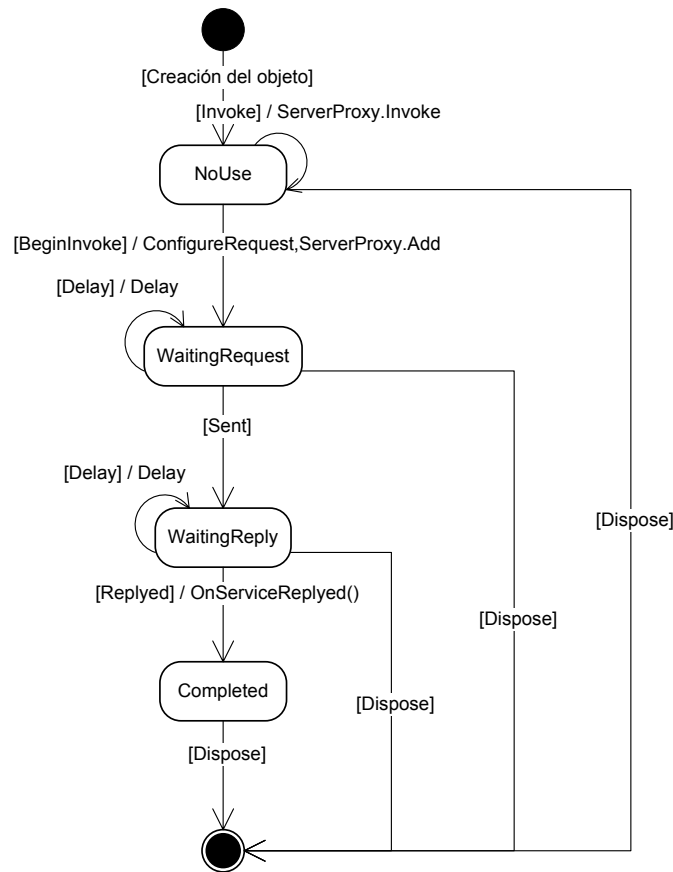


Figura 4.51 Diagrama de estados de las invocaciones a servicios.

Las partes más relevantes de esta clase como objeto individual son las transiciones de cambio de estado. Cada transición implica pocas acciones, pero todas ellas clave para el correcto funcionamiento de la plataforma. El patrón de máquina de estados permite modularizar las transiciones y delegar responsabilidades en cada estado, haciendo más fácil la comprensión y la extensión de las mismas. Las acciones posibles, transiciones y responsables de provocarlas las resumimos en la siguiente lista:

- ▷ **Invocación directa (Invoke):**
 - ▶ *Responsable:* el programador, después de crear una petición de servicio y haberla configurado, se utiliza para enviar invocaciones sin hacer uso de la plataforma.
 - ▶ *Acción:* Se invoca la petición a través del ServerProxy.

- ▶ *Transición:* No provoca transición, pero sólo es válida desde el estado NoUse.
- ▷ **Invocación desde la plataforma (BeginInvoke):**
 - ▶ *Responsable:* el programador, después de crear y configurar una petición.
 - ▶ *Acción:* se configuran los tiempos de la petición en función del tipo que haya definido el programador y se añade a la lista del ServerProxy para que la considere en las sincronizaciones.
 - ▶ *Transición:* provoca transición de NoUse a *WaitingRequest*.
- ▷ **Petición enviada (Sent):**
 - ▶ *Responsable:* el objeto ServerProxy cuando envía la petición al servidor.
 - ▶ *Acción:* ninguna, simplemente se produce la transición.
 - ▶ *Transición:* de *WaitingRequest* a *WaitingResponse*
- ▷ **Respuesta recibida (Replied):**
 - ▶ *Responsable:* el objeto ServerProxy, cuando recibe la respuesta.
 - ▶ *Acción:* se lanza el evento de respuesta recibida.
 - ▶ *Transición:* de *WaitingResponse* a *Completed*.
- ▷ **Retrasar la petición (Delay):**
 - ▶ *Responsable:* el ServerProxy, cuando no puede invocar las peticiones.
 - ▶ *Acción:* se calcula el nuevo tiempo, sumándole al tiempo actual el tiempo de reintento.
 - ▶ *Transición:* no provoca transición de estado.
- ▷ **Destruir la petición (Dispose):**

- ▶ *Responsable*: El programador, para comunicar que la petición ya no es útil, bien porque la cancela o bien porque ya ha recogido su resultado.
- ▶ *Acción*: se liberan los recursos y las referencias. Si se encuentra en estado *WaitingRequest* o *WaitingResponse*, además se elimina la petición de invocación del *ServerProxy* para que ya no la tenga en cuenta.
- ▶ *Transición*: provoca transición al estado final.

Para diseñar esta máquina de estados hemos incorporado una clase para cada posible estado que será la responsable de calcular sus tiempos en función de su estado y de consultar a *ServicePolicy* sobre la política cuando sea necesario. Las clases definidas son *NoUseState*, *WaitingRequestState*, *WaitingResponseState* y *Completed* y todas ellas implementan la interfaz *ServiceRequestState*.

Además, la clase *ServiceRequest* envía el evento “*ServiceResponseReceived*” a la aplicación que se haya registrado para indicar la llegada de la respuesta de una invocación.

- **Cálculo y notificación del siguiente tiempo de invocación.**

A la hora de diseñar esta funcionalidad hemos optado por asignarla a una única clase denominada *ServerProxy*. Ésta es la encargada de recibir las peticiones de invocación y se encarga de gestionar cuándo hace falta invocar al servidor, bien para enviar nuevas solicitudes o bien para recibir las respuestas de las que estén pendientes. Además, para soportar la invocación inmediata de servicios, también es capaz de forzar la invocación de una petición concreta cuando así se lo soliciten.

En resumen, sus responsabilidades son las siguientes:

- Recibir objetos *ServiceRequest* y encargarse de calcular el tiempo en el que deberán ser invocados para que se cumplan los tiempos que estén marcados en la configuración de cada petición, tanto para el envío como para la recepción.
- Comunicarse con el servidor, transformando las peticiones y respuestas desde la plataforma hacia el servidor y viceversa.

En la Figura 4.52 se muestra un diagrama de clases general de esta funcionalidad.

- ▷ *levelLow*: valor de referencia por debajo del cual la batería se considera en estado bajo.

Esta clase es la encargada de parsear la configuración del fichero XML para la sección del gestor de batería.

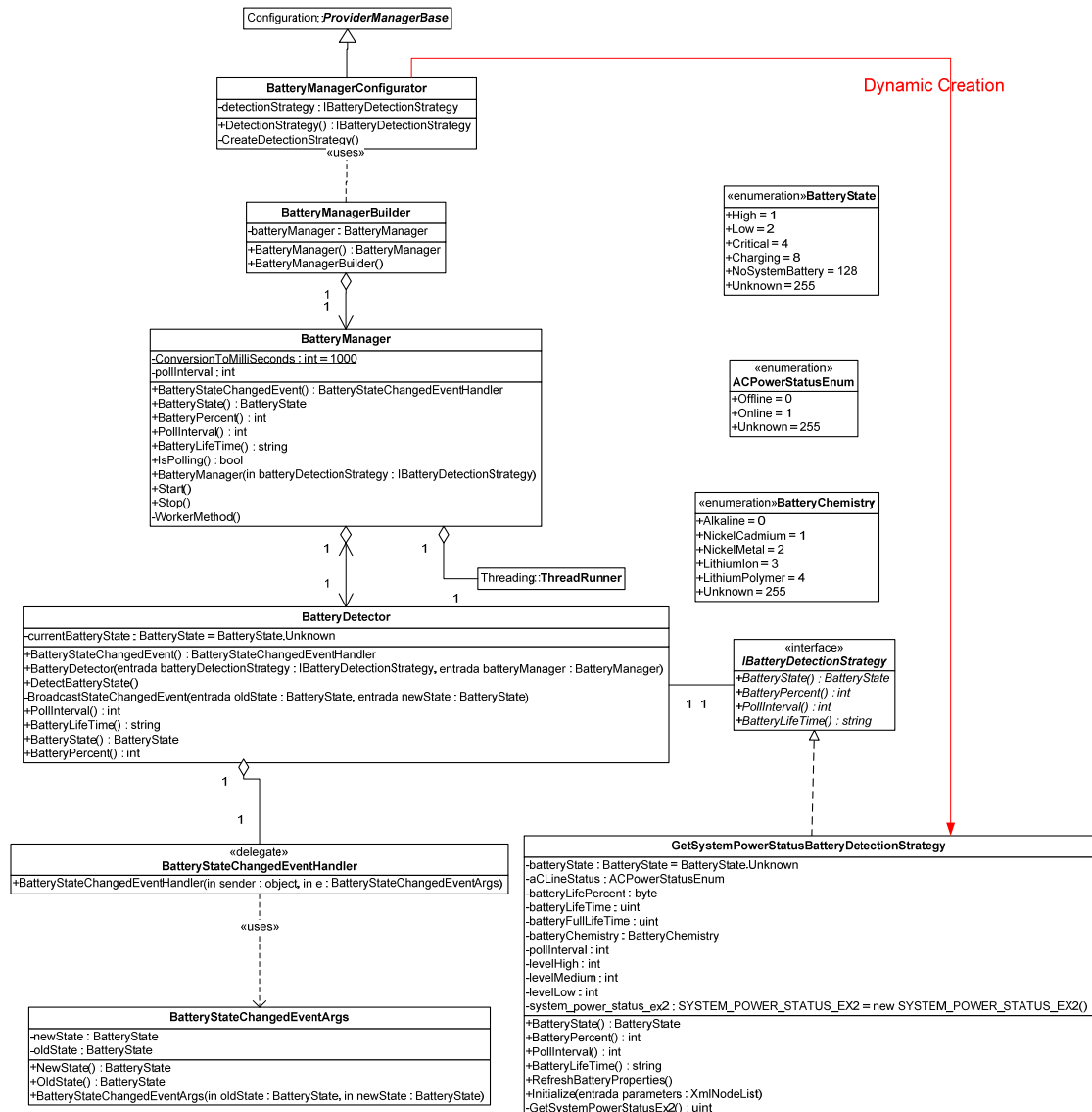


Figura 4.53 Diagrama de clases del Gestor de batería.

- *BatteryDetector*: es el encargado de obtener el estado de la batería a partir de la estrategia que se haya definido en la configuración y comprobar si ha cambiado o no. En caso de que haya cambiado será el responsable de lanzar el evento “BatteryStateChanged” que recibirá el gestor central. Dicho evento es accesible desde el *BatteryManager*, pero está delegado en el *BatteryDetector*.

- *IBatteryDetectionStrategy*: interfaz que deben implementar las estrategias de detección de batería, que serán las responsables de acceder a las funciones internas del sistema operativo y devolver los valores actuales de la batería.

4.4.4.2 Gestor de comunicaciones

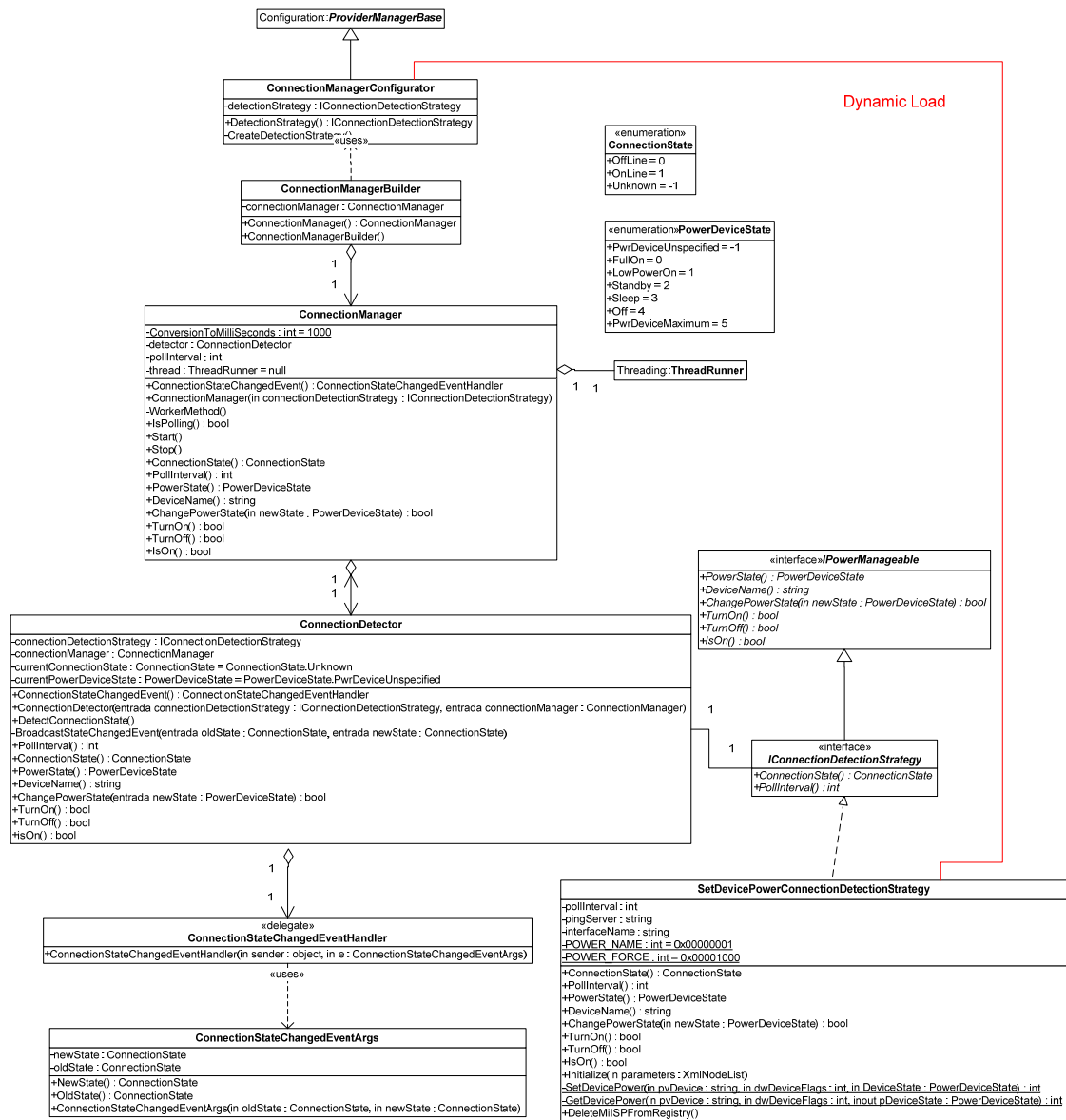


Figura 4.54 Diagrama de clases del Gestor de comunicaciones.

Las responsabilidades del gestor de comunicaciones son las siguientes:

- Monitorizar de forma periódica el estado de la interfaz de comunicaciones.
- Notificar mediante eventos el cambio de dicha interfaz.

Capítulo 5

M2A: Metodología de Movilización de Aplicaciones

Con el fin de proveer de un medio más adecuado para la aplicación práctica de la solución de movilidad definida en esta tesis, hemos desarrollado una metodología simple para la movilización de aplicaciones empresariales. Esta metodología ha sido evaluada mediante el desarrollo de un prototipo que permite la verificación de la hipótesis planteada.

5.1 Definición de la Metodología de Movilización de Aplicaciones basada en el middleware MOSE (M2A)

Mediante esta metodología vamos a proponer un modo de llegar desde la idea de movilizar una aplicación hasta la implementación de dicha idea. M2A (*Metodología de Movilización de Aplicaciones*) es una metodología para la movilización de aplicaciones empresariales basadas en el middleware MOSE.

Esta metodología se puede aplicar en dos ámbitos: por un lado, para el desarrollo de una nueva aplicación móvil que accede a sistemas empresariales existentes y, por otro lado, para dotar de funcionamiento offline y de una gestión de aplicaciones y recursos óptima a una aplicación móvil existente.

Desde este punto de vista, M2A establecerá los pasos que deben seguirse desde el diseño conceptual hasta la implementación para dotar de movilidad y de una gestión más óptima a una aplicación basándose en la solución definida en esta tesis. Esta metodología se debería aplicar durante las **fases de análisis y diseño** de la aplicación a movilizar.

Las características básicas de M2A son:

- Es *simple*, se compone de una serie de pasos secuenciales que facilitan el diseño de la aplicación.
- Es *asistido*, proporcionamos una serie de fichas a rellenar por el diseñador de la aplicación que facilitan la aplicación de la metodología.
- Es *compatible con la metodología RUP*³¹, que se utiliza ampliamente en los proyectos de construcción de componentes software [Castillo05] con lo que se consigue que la curva de aprendizaje de los participantes de los proyectos de desarrollo del software sea mínimo.
- Es *compatible con el lenguaje de modelado UML* [Rumbaugh+05], con lo que facilita el uso de la metodología en desarrollos orientados a objetos.

5.2 Fases de M2A

M2A define una serie de fases a seguir para la movilización de una aplicación. Las fases a incluir en el diseño para dotar de movilidad de una aplicación empresarial son las siguientes:

1. Clasificación de la aplicación a movilizar por su forma de operar.
2. Adaptación de la aplicación al enfoque orientado a datos (si la aplicación es *orientada a servicios*, no será necesaria realizar ninguna actividad en la fase de diseño).
3. Modelizar el comportamiento de la aplicación mediante el lenguaje ANE-SML.

En la Figura 5.1 se muestra en qué fase del ciclo de vida de desarrollo de software tradicional se aplicaría esta metodología. Concretamente, se procedería a aplicarla dentro de la fase de diseño, después de realizar el diseño tradicional.

5.2.1 Clasificación de la aplicación a movilizar

En esta fase, procedemos a clasificar la aplicación en función de cuál es su comportamiento deseado ante la ausencia de conexión. Este comportamiento lo definirá el diseñador de la

³¹ Rational Unified Process.

Las dos primeras técnicas dependen de la funcionalidad que proporciona el sistema de gestión de base de datos elegido. Por el contrario, la utilización de pools de claves primarias es un método más general y adaptable a la mayoría de bases de datos.

El procedimiento de utilización y mantenimiento de un pool de claves primarias requiere la realización de una serie de acciones sobre la base de datos consolidada:

1. Añadir una nueva tabla a la base de datos donde se reflejen las claves primarias disponibles para cada usuario remoto. Dicha tabla contará con dos columnas, una que indica la clave disponible y otra que indica el usuario remoto que puede utilizarla.
2. Dicha tabla será también sincronizable y las aplicaciones remotas tendrán que utilizarla a la hora de insertar nuevas filas. Cada vez que inserten una nueva fila, deberán obtener la clave primaria de la tabla de pool, insertarla en la tabla correspondiente y después borrar la clave que han utilizado de la tabla de pool.
3. El mantenimiento de la tabla pool requiere que se vaya actualizando su contenido de manera automática según se van utilizando. Este problema se puede abordar de varias formas. Puede ser tarea del administrador de manera manual o se puede programar de manera automática desde fuera o dentro de la base de datos (mediante un trigger). La elección de un método u otro dependerá de la política de cada aplicación respecto a las claves primarias.

Técnicas de sincronización

Tal y como se definió en el capítulo dos, la técnica de sincronización es la definición de las condiciones y mecanismos por los cuales se va a llevar a cabo la sincronización de los datos entre el servidor y la base de datos remota. De manera natural surgen dos técnicas, la *sincronización basada en timestamp*, que podríamos resumirla en “actualiza lo que tenga una fecha más reciente a los datos que hay en remoto” y la *sincronización basada en snapshot*, que resumimos en “vuelve a cargar todos los datos”. Sobre estas técnicas básicas de sincronización podemos aplicar a su vez la técnica de sincronización selectiva, que apoyándose en alguna de las anteriores, trata de optimizar el proceso de sincronización evitando tener que sincronizar todas las tablas.

La elección de la técnica de sincronización está condicionada por el sistema de sincronización elegido, ya que no ofrecen todas las técnicas y, además, deberemos considerar dos aspectos: la frecuencia en la que los datos son sincronizados y el número de cambios que se producen en los datos entre sincronizaciones.

A continuación explicamos más en detalle en qué consisten y las implicaciones que tiene la elección de cada una de ellas.

Sincronización basada en timestamp

El método basado en *timestamp* es la técnica general más empleada para llevar a cabo una sincronización de datos eficiente. Esta técnica implica controlar el último instante en el que el usuario sincronizó sus datos. Mediante esta información es posible controlar las filas descargadas en cada base de datos remota para transferir solamente aquellas filas que hayan cambiado desde la última sincronización.

Esta sincronización es muy aconsejable en comunicaciones inalámbricas ya que reduce la transferencia de datos al mínimo consiguiendo así incrementar la eficiencia, reducir el consumo de energía y los costes de las comunicaciones.

En concreto, para aplicar este tipo de sincronización, es necesario modificar el esquema de la base de datos consolidada añadiendo una nueva columna en cada tabla para que registre el instante de la última modificación de la fila. Sin embargo, en la base de datos remota no es necesario incluir esta columna, ya que se controlaría de manera automática desde el servidor.

Además, todo servidor de sincronización define una serie de funciones de sincronización que se aplican durante las fases de sincronización (fase de descarga y fase de subida) que deberán ser modificadas para dar soporte a este tipo de sincronización.

La opción de añadir una columna resulta muy agresiva para la base de datos consolidada, ya que implica modificar el esquema de todas las tablas que se quieran sincronizar, por eso, la técnica basada en el uso de tablas de tiempos resulta la mejor opción. En dicha tabla de tiempos se almacenarán las claves primarias de la tabla original y una marca de tiempo que indique el último acceso a dicha fila. El mantenimiento correcto de esta tabla implica definir los triggers necesarios para que cada vez que se inserte, actualice o eliminen filas de la tabla, dichos cambios se reflejen en las marcas de tiempo de cada fila. Como consecuencia, se complica la lógica del servidor pero, sin embargo, se compensa con la mejora del rendimiento de los flujos de sincronización.

El servidor de sincronización deberá mantener automáticamente la marca de tiempo de la última de sincronización de cada usuario, lo cual facilita la obtención de las marcas de tiempo.

En resumen, los pasos para implementar esta técnica de sincronización son los siguientes:

1. Añadir una columna a la tabla a sincronizar (o una nueva tabla a la base de datos) que mantenga el valor de la marca de tiempo de la última actualización sufrida por cada fila. El tipo de datos de esta columna tendrá que ser compatible con el valor de la marca de tiempo del servidor de sincronización.
2. Configurar las funciones de sincronización de la fase de descarga en el servidor de sincronización para que comprueben dicho valor.
3. Definir una función para que el valor de la marca de tiempo se actualice de manera automática en la base de datos consolidada cuando se insertan nuevas filas o se modifican durante la fase de subida.

Sincronización basada en snapshot

Como alternativa a la sincronización descrita en el apartado anterior, existe la sincronización basada en “snapshot”. Esta técnica implica descargar completamente todas las filas de una tabla, aun habiendo sido descargadas previamente. Lógicamente, se trata de un método simple que puede limitar el rendimiento en algunos casos, por lo que está recomendado cuando las tablas tienen las siguientes características:

1. Relativamente escaso número de filas, para que no afecte excesivamente al rendimiento.
2. Filas que se modifican frecuentemente, con lo que se justificaría el volver a obtenerlas todas, ya que con alta probabilidad casi siempre serán distintas.

La sincronización basada en snapshot no obliga a modificar el esquema de las tablas, por lo que puede ser más recomendable en ciertos casos, siendo asumible la diferencia de rendimiento con respecto al método anterior.

En resumen, para implementar la sincronización basada en “snapshot” habrá que configurar las funciones de sincronización correspondientes a la fase de descarga para que seleccionen todas las filas de cada tabla.

Sincronización selectiva

Las bases de datos consolidadas y las remotas generalmente no disponen de los mismos datos. En la práctica, las bases de datos de los dispositivos móviles tienen un subconjunto de filas, campos y tablas. Por lo tanto, resulta interesante disponer de la posibilidad de realizar una sincronización selectiva y, en consecuencia, sincronizar los datos que realmente resultan interesantes para la aplicación a movilizar. Incluso la base de datos local del dispositivo

puede incorporar tablas adicionales con información exclusiva del cliente sin necesidad de realizar una réplica o sincronizar contra la base de datos consolidada.

En la solución definida en esta tesis hacemos uso de la sincronización selectiva persiguiendo sincronizar sólo los datos necesarios ahorrando de esta manera energía, coste de la comunicación y recursos mediante una gestión inteligente basada en políticas.

Fragmentación de la base de datos consolidada

No siempre es necesario que las bases de datos remotas sean una copia exacta de la base de datos consolidada, es decir, puede haber columnas dentro de las tablas que no sean necesarias en los clientes remotos.

A la técnica de diseñar la base de datos remota para que incluya solamente las tablas, columnas y filas estrictamente necesarias para sí se le denomina **fragmentación** (apartado 2.5.1.2).

La fragmentación puede ser de tres tipos: **fragmentación horizontal** (sólo se seleccionan las filas necesarias), la **fragmentación vertical** (solamente las columnas necesarias) o **fragmentación mixta** cuando se combinan las dos anteriores.

La implementación de una fragmentación, previo diseño de los esquemas correspondientes de las bases de datos, implica redefinir las funciones de sincronización de manera que tengan en cuenta las correspondencias entre los esquemas, principalmente utilizando condiciones en cada una de las sentencias SQL de sincronización.

Tratamiento de los conflictos

Tal y como se vio en el apartado 2.5.1.3, durante la sincronización se pueden producir diferentes conflictos (por ejemplo, conflictos de inserción, conflictos de actualización,...) debido a cambios realizados por múltiples usuarios sobre el mismo conjunto de datos.

Para el tratamiento de estas situaciones especiales habrá que definir una serie de mecanismos que ayuden, por una parte, a detectar cuándo se ha producido un conflicto y, por otra, a resolverlos para cada tabla determinada.

El concepto de conflicto puede confundirse con otras situaciones similares, por ejemplo, en la Tabla 5.1 se muestran las posibles situaciones anómalas y cómo las podría considerar el servidor de sincronización.

- ▷ **requestTime:** Indica el tiempo máximo que se puede esperar antes de realizar la invocación de un método.
 - ▷ **responseTime:** Indica el tiempo máximo que se puede esperar antes de obtener la respuesta del método.
 - ▷ **requestSpecificTime:** Este campo permite al diseñador definir un tiempo específico para la invocación de una determinada solicitud de invocación durante la ejecución de la misma y así poder ignorar la política definida. Este campo no se incluye en la política pero tendrá que tenerlo en cuenta durante la fase de implementación.
 - ▷ **responseSpecificTime:** Este campo permite al diseñador definir un tiempo específico para la recepción de una respuesta de una determinada solicitud de invocación durante la ejecución de la misma y así poder ignorar la política definida. Este campo no se incluye en la política pero tendrá que tenerlo en cuenta durante la fase de implementación.
- e) Conversión de la ficha anterior al formato del lenguaje definido para MOSE.

5.2.3.3 Definición de la política con respecto a la configuración global de la aplicación

El diseñador deberá definir las opciones de configuración específicas para la aplicación utilizando el lenguaje definido. Entre los atributos a definir se encuentran el identificador de la aplicación (*applicationID*) y el tiempo de inactividad (*idletime*), ambos obligatorios. Además, deberá definir otros elementos opcionales como el motor de sincronización (*databaseEngine*), la fuente de datos (*connectionString*),...

5.3 Aplicación de la metodología M2A al desarrollo de una aplicación móvil concreta

A la hora de seleccionar la aplicación prototipo sobre la que íbamos a realizar la validación de la hipótesis planteada haciendo uso del middleware MOSE y de la metodología definida para dicho middleware, se tuvieron en cuenta los resultados del estudio sobre las necesidades de movilidad de las empresas (apartado 4.3.1.1).

En él se delimitaba el área de aplicación de estas soluciones a las siguientes áreas: fuerza de ventas, fuerza de trabajo móvil y transporte y logística. De ahí que hayamos seleccionado

una aplicación que se enmarca dentro de una de estas áreas, concretamente, el área de fuerza de trabajo móvil.

La aplicación móvil seleccionada permite acceder a un sistema central de Operación y Mantenimiento (O&M) para la gestión de incidentes, enfocada en el servicio del Metro Bilbao [MetroBilbao07]. Esta aplicación móvil fue una de las aplicaciones desarrolladas dentro del ámbito del proyecto WALLIP (*WiFi-2.5G: Vertical handover and location based service and applications*) [Lago+06] financiado a través del programa INTEK del departamento de Industria, Comercio y Turismo del Gobierno Vasco [Intek07] con número I-CN03DS02.

Esta aplicación móvil fue diseñada para su funcionamiento en online, lo que pretendemos es dotarle de funcionamiento online/offline a través de la solución de movilidad. Para ello, el primer paso que realizamos fue a aplicarle la metodología M2A para adaptarla al middleware definido.

5.3.1 Descripción del sistema de O&M y de la aplicación a movilizar

5.3.1.1 Propósito del sistema

El punto de partida de esta aplicación consistía en un típico escenario de O&M y su objetivo era permitir la gestión de las incidencias que se produzcan en un sistema y poder asignarlas remotamente en función del lugar donde se encuentren los operarios. Los requisitos de esta aplicación los obtuvimos a partir del funcionamiento real de un sistema concreto, como es el del Metro Bilbao, por lo que algunas decisiones de diseño están orientadas a dichos requisitos. Además, aprovechando el Servicio de Localización de la plataforma WALLIP [Lago+06], implantamos un sistema automático de asignación de incidencias en función de la cercanía de los operarios y de la carga de trabajo que tuvieran asignada en ese momento.

Este sistema tiene como función principal la gestión de las incidencias que pueden ocurrir en las diferentes estaciones del metro. A cada operador se le va a dotar de una PDA que le permita gestionar las incidencias que le son asignadas y recibir las notificaciones desde la oficina central de nuevas incidencias que hayan sucedido dentro del área donde se encuentra actualmente.

El sistema dispone de varios componentes cuya función se especifica a continuación:

- *Gestor de incidencias*: este componente lleva a cabo todas las operaciones que pueden ser realizadas sobre las incidencias. Para facilitar la comunicación con otros componentes ofrece una interfaz basada en servicios web.

- **Servidor web empresarial 1**

Este servidor contiene el servicio empresarial al que acceden las aplicaciones móviles. Puesto que su interfaz de acceso era mediante servicios web, instalamos como servidor web, el IIS de Microsoft.

6.1.3 Incidencias encontradas

- **Sobre el Gestor de batería.**

A la hora de implementar el gestor de batería para el dispositivo móvil, vimos que la versión 2.0 del Compact Framework .NET, utilizada para su implementación no poseía clases para monitorizar el estado de la batería ni los cambios que se producían en ella. En cambio, para ordenadores de sobremesa, el framework sí incluye algunas clases que realizan estas funcionalidades. Es por ello, que tuvimos que implementar el gestor de batería a un bajo nivel directamente a través de los drivers proporcionados por el fabricante del dispositivo móvil.

- **Sobre el Gestor de comunicaciones**

A la hora de llevar a cabo la implementación de este componente surgieron algunos aspectos con respecto al manejo de la interfaz inalámbrica de la Pocket PC que detallamos a continuación.

Sin entrar en mucho detalle, en los Pocket PC, el propio sistema operativo a nivel interno lleva la gestión de los distintos dispositivos de red a través de un componente que se llama `Connecti onManager`.

Dicho componente tiene control total sobre los diferentes interfaces y dispositivos y es el que crea las conexiones automáticamente para las aplicaciones. Manejar este componente desde .NET es complicado, ya que requiere mucho control a nivel de sistema operativo. Lo que sí se dispone es de algunas llamadas a sistema que satisfacen las necesidades de esta plataforma, como son, consultar el estado del interfaz (si está conectada la antena inalámbrica) y poder cambiarlo cuando se necesite. Estas llamadas las hemos implementado en la estrategia de gestión de la conexión (`SetDevicePowerConnectionDetectionStrategy`).

Windows CE divide internamente los interfaces con capacidad de gestión de energía en ciertos estados, declarados en la enumeración `PowerDevi ceState`. Después, en función de cómo cada fabricante implemente su driver, se permitirá un control mayor o menor

sobre la gestión de energía del dispositivo. Es decir, no todos los dispositivos con gestión de energía tienen que tener todos los estados o ser capaces de cambiar entre ellos.

De las pruebas realizadas con el interfaz de la Pocket PC PDA2K hemos comprobado que únicamente posee los estados de FullOn, Sleep y Off, por lo que será con los que trabajemos en las pruebas de la plataforma.

Como hemos comentado anteriormente, Windows CE está pensado para que la gestión de los interfaces se lleve a cabo en el núcleo del sistema operativo y no desde una aplicación externa. Esto implica que si con la estrategia que se ha implementado se cambia el estado de la interfaz “manualmente” el sistema deja de tener control sobre ella. Es decir, si se apaga la interfaz por programa, se deberá encenderla por programa o el sistema ya no la reconocerá. Por ello, es necesario devolver el control al sistema cuando ya no necesitemos gestionar la interfaz. Esto se consigue cambiando el estado de la interfaz a PwrDevilUnspecified.

6.2 Evaluación

6.2.1 Aplicaciones desarrolladas

Para validar el funcionamiento de la solución propuesta así como la metodología M2A, desarrollamos dos aplicaciones móviles junto con su servicio empresarial. Para que la evaluación fuese lo más completa posible, optamos por desarrollar dos tipos de aplicaciones distintas:

- Aplicación móvil para la Operación y Mantenimiento del metro.

Esta aplicación fue introducida en el capítulo cinco, al tomarse como base para la evaluación de la metodología. Eligimos esta aplicación puesto que era un ejemplo de adaptación la movilidad de un sistema empresarial existente.

- Aplicación móvil Detective.

Esta aplicación sirve como ejemplo de desarrollo de una aplicación móvil desde cero. Su área de aplicación se encuentra en el dominio de los detectives privados.

6.2.1.1 Aplicación móvil Detective

La segunda aplicación desarrollada para llevar a cabo las pruebas de la plataforma consiste en un gestor de datos y servicios aplicado en el dominio de los detectives privados.

- ▷ *Evaluación de la adaptación de la técnica DPM.*

Para llevar a cabo esta evaluación, la primera actividad que realizamos fue el cálculo del idleTime para la interfaz WLAN. Una vez calculado este tiempo, comprobamos si se aplicaba la técnica (cambio de estado de la interfaz) cuando verdaderamente era necesaria siguiendo el algoritmo descrito.

- ▷ *Evaluación de la técnica para maximizar el ahorro de energía basado en garantizar el cambio a un estado de bajo consumo.*

Para ello, ante cada expiración del temporizador, calculamos el tiempo de relleno (Tstuf) adecuado basándose en el valor del idleTime y el tiempo de procesamiento estimado. Para el cálculo de este último tiempo hicimos uso de la técnica de la media exponencial.

Comprobamos durante la ejecución de los escenarios que estos aspectos eran evaluados y que, verdaderamente, conseguimos, gracias a la aplicación de este algoritmo, el ahorro de energía cumpliendo con las necesidades de las aplicaciones.

- **Módulo de gestión de recursos**

Este módulo debía permitir la monitorización del estado de los recursos y gestionar las peticiones para la aplicación de la técnica de adaptación sobre el recurso (RO1). Como en esta tesis hemos utilizado dos gestores, debemos evaluarlos por separado:

- *Gestor de batería:* comprobamos si monitorizaba el estado de la misma y notificaba el cambio de estado adecuadamente.
- *Gestor de comunicaciones:* comprobamos si monitorizaba el estado de la misma, notificaba el cambio de estado y si se aplicaba la técnica de adaptación sobre la interfaz WLAN.

Para ello, definimos varios escenarios en los que la batería cambiaba de estado y, además, provocamos la necesidad de aplicar la técnica de adaptación sobre la interfaz.

En cada uno de los escenarios vamos a incluir los siguientes elementos que resumen el resultado de las pruebas realizadas sobre el mismo:

- Una tabla con los parámetros que se pretendían evaluar en el escenario.
- Una figura que muestra la línea de ejecución general de las distintas actividades realizadas por el usuario dentro del escenario con su PDA.

- Un fragmento del fichero de especificación de las necesidades utilizado para el escenario basado en el lenguaje ANE-SML, con las secciones que se consideran de mayor interés.
- Resultados obtenidos de las pruebas realizadas sobre el escenario:

Sobre los escenarios hemos realizado tres pruebas distintas que permitieron validar la hipótesis presentada en esta tesis:

- ▷ **Prueba A:** Esta prueba consistió en la ejecución del escenario sin la solución de movilidad, por lo que la interfaz WLAN siempre se encontraba en estado *working*. Cada actividad fue realizada en el momento en el que el usuario la solicitó ya que no se rige por ninguna política de tiempos.
- ▷ **Prueba B:** Esta prueba consistió en la ejecución sobre la solución de movilidad pero con la aplicación exclusiva de la técnica de ahorro basada en la adaptación de la técnica DPM. Esta prueba se basó en el algoritmo definido para la gestión inteligente de los recursos y las aplicaciones. Basa su funcionamiento en los tiempos definidos en la política de la aplicación y en el estado de los recursos para realizar una óptima planificación del tráfico y de los recursos, en este caso, de la energía, situando la interfaz WLAN en su estado de bajo consumo (*sleep*) para lograr un mayor ahorro, pero no aplica la técnica para maximizar el ahorro de energía.
- ▷ **Prueba C:** Esta prueba consistió en la ejecución sobre la solución de movilidad completa, incluyendo la técnica de adaptación de la técnica DPM junto con la técnica para maximizar el ahorro de energía definida en esta tesis (ver apartado 4.3.2.3). Este escenario se basó en el algoritmo definido para la gestión inteligente de los recursos y las aplicaciones. Su funcionamiento se basó en los tiempos definidos en la política de la aplicación y en el estado de los recursos para realizar una óptima planificación del tráfico y de los recursos, en este caso, de la energía, situando la interfaz WLAN en su estado de bajo consumo (*sleep*) para lograr un mayor ahorro.

De cada una de estas pruebas incluimos los siguientes elementos:

- ▷ Una figura que muestra la línea de ejecución real seguida durante la prueba, mostrando el comportamiento de la interfaz WLAN durante la ejecución, así como los tiempos definidos en la política para cada actividad, si procede.

En resumen, la principal aportación del presente trabajo de investigación es una solución para la gestión inteligente de aplicaciones móviles y recursos sensible a las necesidades de las aplicaciones que presenta diversas mejoras en varios aspectos frente a las soluciones existentes, tal y como se puede observar en la Tabla 7.3.

7.1.3 Proyectos de investigación relacionados y publicaciones

El área de trabajo abordado en esta tesis ha dado lugar a la aprobación de dos proyectos de investigación con subvención externa, que corresponden a la convocatoria INTEK, de la Consejería de Industria del Gobierno Vasco. Éstos datan de los periodos 2003/2004 y 2004/2006, tienen un carácter industrial, fueron desarrollados en consorcio con empresas del País Vasco (Euskaltel, CBT, Nextel, Fundación Robotiker, Deusto Sistemas y Fagor Electrónica) y el papel desempeñado por la autora de la tesis fue el de proporcionar el conocimiento científico sobre las soluciones de movilidad y sus requisitos.

Los dos proyectos fueron finalizados de una forma satisfactoria y aunque las soluciones obtenidas no coinciden exactamente con la presentada en esta memoria, dado el diferente carácter de las convocatorias asociadas a los proyectos citados y la naturaleza de una tesis doctoral, sus similitudes hicieron que estos proyectos fuesen muy interesantes a la hora de refinar ciertas partes de la solución desarrollada y contrastar su aplicabilidad en la industria del software.

Asimismo, los resultados de estos proyectos y la evolución de la solución a lo largo de los años ha dado lugar a un total de ocho publicaciones en congresos nacionales e internacionales directamente vinculadas con el tema de la tesis [Lago+06, Lago+06a, Lago+06b, Lago+06c, Lago+06d, Trueba+06, Lago+05, Vazquez+05]. Seis de estas publicaciones se han realizado en el 2006, lo cual es un indicador más del interés vigente entre la comunidad científica en este tema.

7.2 Líneas de investigación futuras

Por extraño que pueda parecer, después de más de una década, se podría decir que estamos en los orígenes de la Computación móvil. Desde luego parece razonable pensar, a tenor de las innumerables posibilidades que día a día vemos que los dispositivos móviles, Internet, las redes inalámbricas nos ofrecen y de su evolución en los últimos años, que el uso que en la actualidad se hace de este medio puede ser considerado muy pequeño si lo comparamos con su previsible explotación en un futuro.

La utilización de Internet y de aplicaciones móviles críticas sobre dispositivos móviles no hace sino crecer aceleradamente, y nuestra sociedad alcanza continuamente mayores niveles de dependencia respecto de todo tipo de medios de comunicación. La información se ha convertido en el elemento fundamental del desarrollo económico en nuestros días y la disponibilidad de la misma en cualquier lugar y en cualquier instante, es esencial.

Nadie discute que el porvenir de la Computación móvil es muy prometedor. Es seguro que el número de usuarios continuará creciendo, así como la diversidad y criticidad de los servicios ofertados. Pero debido a las limitaciones de los dispositivos móviles (principalmente de la batería, cuya tecnología progresa lentamente), a las limitaciones de la movilidad y a las limitaciones de las redes inalámbricas, la importancia de desarrollar soluciones de movilidad que gestionen de manera inteligente estas limitaciones y las aplicaciones serán fundamentales para que el crecimiento de la demanda de este tipo de servicios se produzca.

Con esta tesis pretendemos realizar una pequeña aportación contribuyendo de este modo al desarrollo de dichos servicios, pero son necesarias nuevas líneas de investigación que afronten diversos aspectos que no han sido abordados por esta tesis.

A modo de ejemplo, esta tesis se ha centrado en dar soporte a las aplicaciones móviles empresariales quedando excluidas otro tipo de aplicaciones como aplicaciones multimedia en las que la calidad de servicio es un elemento clave. Se podrían analizar las necesidades de estas aplicaciones y adaptar la solución a los requisitos que planteen. Además, se podrían crear nuevas extensiones al lenguaje de especificación de necesidades ANE-SML para adaptarse a este tipo de aplicaciones.

Otras áreas que han quedado excluidas del ámbito de esta tesis y que serían interesante incorporarlas serían el uso de interfaces adaptativas, la seguridad y el soporte multiplataforma.

En relación con el núcleo, hemos definido un algoritmo de gestión inteligente de aplicaciones y recursos sensible a las aplicaciones. La gestión de recursos se ha centrado en la gestión inteligente de las comunicaciones y de la energía, persiguiendo minimizar el consumo de energía gestionando las técnicas de adaptación proporcionadas por una interfaz inalámbrica.

Por otro lado, se debería profundizar en el diseño de una gestión inteligente de la energía integrada que abarque todos los niveles: a nivel de hardware, una versión más compleja, debería soportar la gestión del consumo de energía de todos los recursos de los dispositivos móviles (CPU, disco duro, etc.) haciendo uso de las técnicas de adaptación proporcionadas por los fabricantes. A nivel del sistema operativo, aplicando técnicas como *performance*

scaling. A nivel de middleware, complementándolo con nuevas técnicas como las de compresión, ejecución remota de tareas, streaming de video, técnicas de prefetching y caching, etc.

Dentro de esta área, se podría explorar la combinación de las técnicas de ahorro de energía planteadas en esta tesis, con el área de asignación de energía a recursos para cada aplicación en función de sus prioridades y de la energía restante.

Con respecto a la nueva técnica de ahorro de energía planteada, se podrían incorporar otras variables en su aplicación que no han sido tenidas en cuenta en esta tesis.

Referencias Bibliográficas

- [Abukmail+04] Abukmail, A. and Helal, A. S. "Power Awareness and Management Techniques". In *Mobile Computing Handbook*, I. Mahgoub and M. Ilyas (Eds), CRC Press, 2004, pp. 731-746.
- [Acosta+01] Acosta, J., Morales, J. and Esparza, P. *Aprovechamiento electroquímico de la energía*. Servicio de publicaciones de la Universidad de la Laguna, 2001.
- [ACPI07] ACPI. *ACPI - Advanced Configuration and Power Interface Home Page*. Available at <http://www.acpi.info/>, 2007.
- [Adams02] Adams, H. *Asynchronous operations and Web services*. IBM. Available at <http://www-128.ibm.com/developerworks/library/ws-asynch1.html>
- [Agrawal+01] Agrawal, S. and Singh, S. "An experimental study of TCP's energy consumption over a wireless link". In *Proceedings of the 4th European Personal Mobile Communications Conference (EPMCC'01)*, Vienna, Austria, February 2001.
- [AHCJET05] AHCJET. *Las Telecomunicaciones y la Movilidad en la Sociedad de la Información*. Telefónica I+D, 2005. Disponible en http://www.tid.es/html/libros_sector_telecomunicaciones.html
- [Ahuja+00] Ahuja, S. Agarwal, J., Singh P. and Shorey, R. "Performance of TCP over different Routing Protocols in Mobile Ad-hoc Networks". In *Proceedings of IEEE Vehicular Technology Conference*, vol. 3, 2000, pp. 2315 - 2319.
- [Alonso+04] Alonso, G., Casati, F., Kuno, H., Machiraju, V. *Web Services: Concepts, Architecture and Applications*. Springer Verlag, 2004.
- [AMD07] Advanced Micro Devices Home Page. *Comparaciones Competitivas del procesador AMD Athlon™ 64 X2*. AMD, Inc., 2006. Available at http://www.amd.com/es-es/Processors/ProductInformation/0,,30_118_9485_13041%5E13042,00.html
- [Anand+04] Anand, M., Nightingale, E.B. and Flinn, J. "Ghosts in the machine:

- Interfaces for better power management". In *Proceedings of the Second International Conference on Mobile Systems, Applications, and Services (MOBISYS'04)*, June 2004.
- [Anand+05] Anand, M., Nightingale, E.B. and Flinn, J. "Self-Tuning Wireless Network Power Management". *Wireless Networks*, No. 11, Vol. 4, 2005, pp. 451-469.
- [Andago04] Ándago. *Segundo Informe Ándago sobre el Uso de Linux y Software Libre en el Entorno Corporativo Español*. Disponible en <http://andago.com/>.
- [Angin+98] Angin, O., Campbell, A. T., Kounavis M. E. and Liao R. R.-F. "The Mobiware Toolkit: Programmable Support for Adaptive Mobile Networking". *IEEE Personal Communications Magazine*, August 1998.
- [Apshankar+02] Apshankar, K. et al. *Web Services Business Strategies and Architectures*. P. Fletcher and M. Waterhouse (Eds.), Expert Press Ltd, 2002.
- [Avison+99] Avison, D., Lau, F., Myers, M. and Nielsen, A. "Action research". *Communications of the ACM*, 42(1), 1999, pp. 94-97.
- [Baggio98] Baggio, A. *Design and Early Implementation of the Cadmium Mobile and Disconnectable Middleware Support*. INRIA Research Report 3515, RR-3515, 1998. Available at <http://www-sor.inria.fr/projects/cadmium/>
- [Balan+03] Balan, R. K., Satyanarayanan, M., Park, S. and Okoshi, T. "Tactics-Based Remote Execution for Mobile Computing". In *Proceedings of 1st International Conference on Mobile Systems, Applications, and Services (MobiSys '03)*, San Francisco, CA, May 2003.
- [Bansal+06] Bansal, S., Shorey, R., Gupta, R. and Misra, A. "Energy Efficiency and Capacity of TCP Traffic in Multi-Hop Wireless Networks", *ACM/Kluwer Journal of Wireless Networks*, February 2006.
- [Barbará99] Barbara, D. "Mobile Computing and Databases - A Survey". *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, No. 1, Jan/Feb 1999, pp. 108-117.
- [Belloso00] Belloso, F. "The benefits of event-driven accounting in power-sensitive systems". In *Proceedings of the 9th ACM SIGOPS European Workshop*, September 2000.
- [Benini+99] Benini, L., Bogliolo, A., Paleologo, G.A. and De Micheli, G. "Policy optimization for dynamic power management". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 18, No. 6, June 1999, pp. 813-833.

- [Benini+00] Benini, L., Bogliolo, A. and De Micheli, G. "A survey of design techniques for system-level dynamic power management". *IEEE Transactions on VLSI Systems*, Vol. 8, June 2000, pp. 299-316.
- [Bevis02] Bevis, D. and Patterson, L. *Extending enterprise applications to mobile users*. IBM. July 2002.
- [Blount+04] Blount, M., Perret, V., Yeh, D., Purakayastha, A., Moser, M., Duponchel, Y., Bourges-Waldegg, D. and Graf, M. "Managed Portal Appliance: An Experiment in Extending the Reach of Web Applications". *IEEE International Conference on Mobile Data Management (MDM'04)*, 2004, pp.158.
- [BluetoothSIG07] Bluetooth SIG, Inc. *The Bluetooth Special Interest Group (SIG)*. Available at <http://www.bluetooth.com/Bluetooth/SIG/>, 2007.
- [Bolchini+04] Bolchini, C., Schreiber, F.A. and Tanca, L. "A Context-Aware Methodology for Very Small Data Base Design". *SIGMOD Record*, Vol. 33, No. 1, March 2004.
- [Bolchini+04a] Bolchini, C., Schreiber, F.A. and Tanca, L. *VSDB: A design methodology*. Technical Report 5.2.1, Politecnico di Milano, May 2004.
- [Booch+99] Booch, G., Rumbaugh, J. and Jacobson, I. *The unified modeling language user guide*. Addison Wesley Longman Inc., 1999.
- [Booch+05] Booch, G., Rumbaugh, J. and Jacobson, I. "The unified modeling language user guide (2nd Edition)". Addison Wesley Longman Inc., 2005.
- [Boudreau+01] Boudreau, M.C., Gefen, D. and Straub, D. "Validation in IS Research: A State-of-the-Art Assessment". *MIS Quarterly*, Vol. 25, No. 1, 2001, pp. 1-16.
- [Bozman+02] Bozman, J. et al. *Windows 2000 Versus Linux in Enterprise Computing. An Assessment of Business Value for Selected Workloads*. IDC (Internacional Data Corporation). Available at <http://www.idc.com>
- [Braynard+06] Braynard, R., Silberstein, A. and Ellis, C.S. "Extending Network Lifetime Using an Automatically Tuned Energy-Aware MAC Protocol". *In Proceedings of the 3rd European Workshop on Wireless Sensor Networks*, February 2006.
- [Brooks+00] Brooks, D., Tiwari, V. and Martonosi, N. "Wattch: A framework for architectural-level power analysis and optimizations". *In Proceedings of the 27th Annual International Symposium on Computer Architecture (ISCA-27)*, June 2000.

Referencias Bibliográficas

- [Buchmann01] Buchmann, I. *Batteries in a Portable World: A Handbook on Rechargeable Batteries for Non-Engineers, Second Edition*. Cadex Electronics Inc., May 2001.
- [Burmakin+02] Burmakin, E. M. and Tuominen, J.O. "Development of Mobile Distributed Applications". In *Proceedings of the M-Business International Conference on Mobile Computing*, Athens (Greece), July 2002.
- [Burriss03] Burriss, P. *The Business Value Roadmap to Mobilized Software Solutions*. Online book. FreeMARKETpress, 2003. Available at http://www.ianywhere.com/promos/intel_ebook.html
- [Burriss04] Burriss, P. *The Developer's Roadmap to mobilized software solutions*. Online book. FreeMARKETpress, 2004. Available at http://www.ianywhere.com/promos/intel_ebook.html
- [Cai+05] Cai, L. and Lu, Y-H. "Energy Management Using Data Buffers", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, February 2005, pp.141-152.
- [Campadello03] Campadello, S. *Middleware Infrastructure for Distributed Mobile Applications*. PhD. Thesis, Department of Computer Science, University of Helsinki, Finland, 2003.
- [Cano+00] Cano, J.C. and Manzoni, P. "A Performance Comparison of Energy Consumption for Mobile Ad Hoc Networks Routing Protocols". In *Proceeding of IEEE/ACM Eighth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'00)*, San Francisco (USA), August 2000.
- [Cantera05] Cantera, J. M. "Soluciones de movilidad empresariales: Tecnologías, componentes y aplicaciones". *Comunicaciones de Telefónica I+D*. Telefónica Investigación y Desarrollo, No. 36, Junio 2005.
- [Cao04] Cao, G. "Power-aware Cache Management in Mobile Environments". In *Mobile Computing Handbook*, I. Mahgoub and M. Ilyas (Eds.), CRC Press, 2004, pp. 403-419.
- [Carbonell+03] Carbonell, J., Congosto, M.L., Lucio, J. and Siles, J. *La Empresa en la Red*. Telefónica, S.A., 2003.
- [Cardei+04] Cardei, I. and Du, D-Z. "Energy Efficient Wireless Networks". In *Mobile Computing Handbook*, I. Mahgoub and M. Ilyas (Eds.), CRC Press, 2004, pp. 689-706.
- [Castillo05] Castillo de Mora, R. *La mejora del sistema de calidad*, Nextel Engineering

- SIM SL, Noticias.com, Mayo 2005. Disponible en <http://www.noticias.com/notaprensa/20-05-2005/sim/mejora-sistema-calidad-5dg.html>
- [Castro+04] Castro, P., Giraud, F., Konuru, R., Purakayastha, A. and Yeh, D. "A Programming Framework for Mobilizing Enterprise Applications". In *Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'04)*, 2004, pp. 196-205.
- [Causen+03] Causen, T. and Jacquet, P. "Optimized Link State Routing Protocol (OLSR)". RFC 3626, October 2003. Available at <http://www.ietf.org/rfc/rfc3626.txt>.
- [Chan+03] Chan, D. and Roddick, F. Context-Sensitive Mobile Database Summarisation. ACM International Conference Proceeding Series, Proceedings of the twenty-sixth Australasian conference on Computer science: research and practice in information technology, vol. 35, Australia, 2003, pp. 139-149.
- [Chandra02] Chandra, S. "Wireless network interface energy consumption implications of popular streaming formats". In *Proceedings of Multimedia Computing and Networking (MMCN'02)*, San Jose, CA, January 2002, pp. 85-99.
- [Cheng+04] Cheng, W.-C. and Pedram, M. "Power minimization in a backlit TFT-LCD display by concurrent brightness and contrast scaling". *IEEE Trans. Consumer Electronics*, Vol. 50, No.1, February 2004, pp. 25-32.
- [Cheung05] Cheung, R. "An Adaptive Middleware Infrastructure". In *Proceedings of the 14th International World Wide Web Conference*, ACM Press, May 2005.
- [Chuang+03] Chuang, S.N., Chan, A.T.S., Cao, J. and Cheung, R. "Dynamic Service Reconfiguration for Wireless Web Access". In *Proceedings of the Twelve International World Wide Web Conference*, ACM Press, May 2003, pp. 58-67.
- [Chuang+04] Chuang, S.N., Chan, A.T.S., Cao, J. and Cheung, R. "Actively Deployable Mobile Services for Adaptive Web Access". *IEEE Internet Computing*, March 2004.
- [Chughtai+04] Chughtai, S. and Patterson, L. "Robust mobile-computing products delivering business value to your enterprise. IBM WebSphere Everyplace Access and IBM Workplace Client Technology, Micro Edition". *IBM Pervasive Computing*, October 2004.
- [Chung+99] Chung, E.-Y., Benini, L. and De Micheli, G. "Dynamic power management using adaptive learning tree". In *Proceedings of the 1999*

- IEEE/ACM international conference on Computer-aided design*, San Jose, California, (United States), November 1999, pp.274-279.
- [Chung+02] Chung, E.-Y., Benini, L., Bogliolo, A., Lu, Y-H. and De Micheli, G. "Power Management for Nonstationary Service Requests". *IEEE Transactions on Computers*, Vol. 51, No. 11, November 2002, pp. 1345-1361.
- [Chung+06] Chung, S., Pan, J.R. and Davalos, S. "A Special Web Service Mechanism: Asynchronous .NET Web Services". In *Proceedings of International Conference on Internet and Web Applications and Services/Advanced International Conference on Telecommunications (AICT-ICIW '06)*, 2006.
- [Cignetti+00] Cignetti, T. L., Komarov, K. and Ellis, C. "Energy estimation tools for the palm". In *Proceedings of the 9th ACM Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2000)*, August 2000.
- [CMT04] CMT. *Informe Anual 2004*. Comisión del Mercado de las Telecomunicaciones (CMT), 2004. Disponible en http://www.cmt.es/cmt/centro_info/publicaciones/index.htm
- [CMT05] CMT. *Informe Trimestral. Julio-Septiembre 2005*. Comisión del Mercado de las Telecomunicaciones (CMT). Disponible en http://www.cmt.es/cmt/centro_info/publicaciones/index.htm
- [CMT07] CMT. *Comisión del Mercado de las Telecomunicaciones (CMT) Home Page*. Disponible en at <http://www.cmt.es/cmt/index.htm>, 2007.
- [Coda07] *Coda File System Home Page*. Available at <http://www.coda.cs.cmu.edu/doc/html/index.html>, 2007.
- [Conolly+05] Conolly, T.M. y Begg, C.E. *Sistemas de bases de datos: un enfoque práctico para diseño, implementación y gestión*. Pearson-Addison Wesley, 2005.
- [Corner+01] Corner, M., Noble, B. D. and Wasserman, K.M. "Fugue: Time Scales of Adaptation in Mobile Video". In *Proceedings of the SPIE Multimedia Computing and Networking Conference (MMCN '01)*, San Jose, CA, January 2001.
- [Coulouris+01] Coulouris, G., Dollimore, J., Dormido, S. y Kindberg, T. *Sistemas Distribuidos: Conceptos y Diseño*. 3ª edición, Addison-Wesley, 2001.
- [Date01] Date, C. J. *Introducción a los sistemas de bases de datos*. Pearson Educación, S. A., 2001.
- [DeLara+05] De Lara, E., Choopra, Y., Vaghela, N., Kumar, R., Wallach, D. and Zwaenepoel, W. "Iterative adaptation for mobile clients using existing APIs". *IEEE Distributed Systems Online*, Vol. 6, No 9, September 2005.

- [DISA05] Defense Information Systems Agency (DISA). *Wireless. Security Technical Implementation Guide*. Developed by DISA for the Department of Defense of USA, October 2005, version 4, release 1. Available at www.csrc.nist.gov/pcig/STIGs/wireless-stig-v4r1.pdf
- [Dix+98] Dix, A., Finlay, J., Abowd, G. and Beale, R. *Human-Computer Interaction, second edition*. Prentice Hall, 1998.
- [Dougkis+95] Dougkis, F., Krishnan, P. and Bershad, B. "Adaptive Disk Spin-Down Policies for Mobile Computers". In *Proceedings of the Second Usenix Sump. Mobile and Location-Independent Computing (MLICS'95)*, Usenix, 1995, pp.121-137.
- [Du+04] Du, Y. and Gupta, S.K.S. "Cache Mangement in Wireless and Mobile Computing Environments". In *Mobile Computing Handbook*, I. Mahgoub and M. Ilyas (Eds.), CRC Press, 2004, pp. 337-360.
- [Dutt+07] Dutt, N., Banerjee, K., Benini, L., Lahiri, K. and Pasricha, S. "Tutorial 5: SoC Communication Architectures: Technology, Current Practice, Research, and Trends". In *Proceedings of the 20th International Conference on VLSI Design held jointly with 6th International Conference on Embedded Systems (VLSID'07)*, 2007.
- [Ebert+01] Ebert, J.-P. and Wolisz, A. "Combined Tuning of RF Power and Medium Access Control for WLANs". *Mobile Networks & Applications*, Vol. 6, No. 5, September 2001, pp. 417-426.
- [Edmonds01] Edmonds, T. *Adaptation for Mobile Systems*. PhD thesis, University of Cambridge, 2001.
- [Elbashir+05] Elbashir, K., Deters, R. "Transparent Caching for Nomadic WS Clients". In *Proceedings of the IEEE International Conference on Web Services (ICWS'05)*, 2005, pp. 177-184.
- [Ellis+99] Ellis, C.S., Lebeck, A. and Vahdat, A. *System Support for Energy Mangement in Mobile and Embedded Workloads : A White Paper*. Department of Computer Science Duke University Durham, October 1999.
- [Ellis+99a] Ellis, C. S. "The Case for Higher-Level Power Management". In *Workshop on Hot Topics in Operating Systems*, 1999, pp. 162-167.
- [Elmasri+02] Elmasri, R. y Navathe, S.B. *Fundamentos de sistemas de bases de datos. Tercera edición*. Pearson Educación, S.A., 2002.
- [ENTEL04] ENTEL. "El teletrabajo. Una opción con futuro". *Jornada sobre el teletrabajo "Trabajar y vivir en la red"*, 2004.

- [EPFL+02] EPFL, et al. *Mobile Databases: a Report on Open Issues and Research*. Technical Report, National Center for Scientific Research in France (CNRS), 2002.
- [Erradi+05] Erradi, A., Maheshwari, P. "A broker-based approach for improving web services reliability". In *Proceedings of the 2005 IEEE international conference on web services*, 2005, pp. 355 – 362.
- [EuroCom05] European Commission. *10th Report on the Implementation of the Telecommunications Regulatory Package – 2004*, 2005. Available at http://europa.eu.int/information_society/policy/ecom/doc/implementation_enforcement/annualreports/10threport/com20040759en.pdf
- [Eustat06] Eustat. *Encuesta sobre la Sociedad de la Información (ESI) Empresas 2005*. 2006. Disponible en www.eustat.es
- [Faerber04] Faerber, M. *Application-Specific Power Management for Wireless Networks*. Master Thesis. Operating System Group, University of Erlangen, January 2004.
- [Feeney+01] Feeney, L. M. and Nilsson, M. "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment". In *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'01)*, April 2001.
- [Fernandez+05] Fernández, J.A., Pazos, J.J. and Fernández, A. "Web Services Performance Using Cache". In *Proceedings of IEEE International Conference on Next Generation Web Services Practices (NWeSP'05)*, June 2005, pp. 157-162.
- [Fei+04] Fei, Y., Zhong, L. and Jha, N. K. "An Energy-aware Framework for Coordinated Dynamic Software/Management in Mobile Computers". In *Proceedings of IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOT'04)*, 2004, pp. 306-317.
- [Fielding+99] Fielding, R., Gettys, J., Mogul, J.C., Frystyk, H., Masinter, L., Leach, P. J. and Berners-Lee, T. "RFC 2616: Hypertext transfer protocol – HTTP/1.1," June 1999. Available at <http://www.ietf.org/rfc/rfc2616.txt>
- [Flarion03] Flarion Technologies. *OFDM for Mobile Data Communications*. Flarion Technologies, Inc., 2003. Available at http://www.flarion.com/products/whitepapers/OFDM_Mobile_Data_Communications.pdf
- [Flinn+99] Flinn, J. and Satyanarayanan, M. "Energy-aware adaptation for mobile

- applications". In *Proceedings of the 17th ACM Symposium on Operating Systems Principles*, December 1999.
- [Flinn+99a] Flinn, J. and Satyanarayanan, M. "PowerScope: A Tool for Profiling the Energy Usage of Mobile Applications". In *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications*, 1999.
- [Flinn01] Flinn, J. *Extending Mobile Computer Battery Life through Energy-Aware Adaptation*. Phd. Thesis. School of Computer Science Computer Science Department Carnegie Mellon University, Pittsburgh, December 2001.
- [Flinn+01] Flinn, J., de Lara, E., Satyanarayanan, M., Wallach, D. S. and Zwaenepoel, W. "Reducing the Energy Usage of Office Applications". In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, Heidelberg, Germany, November 2001.
- [Flinn+01a] Flinn, J., Narayanan, D. and Satyanarayanan, M. "Self-Tuned Remote Execution for Pervasive Computing". In *Proceedings of the 8th IEEE Workshop Hot Topics in Operating Systems*, 2001.
- [Flinn+04] Flinn, J. and Satyanarayanan, M. "Managing Battery Lifetime with Energy-Aware Adaptation". *ACM Transactions on Computer Systems*, Vol. 22, No. 2, May 2004, pp. 137-179.
- [Frankel+06] Frankel, S., Eydt, B., Owens, L. and Kent, K. *Guide to IEEE 802.11i: Robust Security Networks*. Draft Special Publication 800-97, National Institute of Standards and Technology (NIST), June 2006. Available at <http://csrc.nist.gov/publications/drafts/Draft-SP800-97.pdf>
- [French+96] French, W.L. and Bell, C.H. *Organisational Development: Behavioral Science Interventions for Organization Improvement*. Prentice-Hall, London, 1996.
- [Gabriel03] Gabriel, L. *Enterprise Mobility Applications*. Tellispark, Inc., April 2003.
- [Gaddah+03] Gaddah, A. and Kunz, T. *A survey of middleware paradigms for mobile computing*. Technical Report SCE-03-16, Department of Systems and Computer Engineering, Carleton University, July 2003. Available at <http://kunz-pc.sce.carleton.ca/Thesis/Middleware.pdf>
- [Gaptel03] Grupo de Análisis y Prospectiva del Sector de las Telecomunicaciones (GAPTEL). *WI-FI, Análisis, diagnóstico y políticas públicas*. Red.es, Noviembre 2003.
- [Gaptel05] Grupo de Análisis y Prospectiva del Sector de las Telecomunicaciones (GAPTEL). *Comunicaciones móviles e inalámbricas*. Red.es, Septiembre 2005.

- [Gartner] Gartner. Gartner Home Page. Available at <http://www.gartner.com>.
- [Gast02] Gast, M. 802.11 ® *Wireless Networks: The Definitive Guide*. O'Reilly & Associates, Inc., April 2002.
- [Gauthier+96] Gauthier, P., Harada, D. and Stemm, M. "Reducing Power Consumption for the Next Generation of PDAs: It's in the Network Interface". In *Proceedings of the International Workshop on Mobile Multimedia Communications (MoMuC'96)*, September 1996.
- [Giguère01] Giguère, E. "Mobile data management: Challenges of wireless and offline data access". In *Proceedings of 17th International Conference on Data Engineering (ICDE)*, 2001.
- [Gitzenis+04] Gitzenis, S. and Bambos, N. "Transmitter Power Control in Wireless Computing". In *Mobile Computing Handbook*, I. Mahgoub and M. Ilyas (Eds.), CRC Press, 2004, pp. 805-836.
- [Glass+02] Glass, R., Vessey, I. and Ramesh, V. "Research in software engineering: an analysis to the literature". *Information and Software Technology*, 44, Elsevier Science B.V., 2002, pp. 491-506.
- [Goñi+00] Goñi, G. and Illarramendi, A. "Mobile Computing: Data Management Issues". In *Advanced Database Technology and Design*, Artech House Inc., 2000.
- [Graham+02] Graham, S. et al. *Building Web Services with Java™. Making Sense of XML, SOAP, WSDL, and UDDI*. SAM Publishing, 2002.
- [Grunwald+00] Grunwald, D., Levis, P. and Farkas, K. I. "Policies for dynamic clock scheduling". In *Proceedings of the 4th Symposium on Operating Systems Design and Implementation (OSDI)*, 2000, pp. 73-86.
- [Gundlach+04] Gundlach, M., Doster, S., Yan, H., Lowenthal, D.K., Watterson, S.A. and Chandra, S. "Dynamic, power-aware scheduling for mobile clients using a transparent proxy". In *Proceedings of the International Conference on Parallel Processing (ICPP'04)*, 2004, pp. 557-565.
- [Gunther+01] Gunther, S. H., Binns, F., Carmean, D. M. and Hall, J. C. "Managing the impact of increasing microprocessor power consumption". *Intel Technology Journal*, Vol.5, No. 1, 2001.
- [Gurumurthi+03] Gurumurthi, S., Sivasubramaniam, A., Kandemir, M. and Franke, H. "DRPM: Dynamic Speed Control for Power Management of Server Class Disks". In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, June 2003, pp. 169-179.

- [Gurun+03] Gurun, S. and Krintz, C. *Addressing the Energy Crisis in Mobile Computing with Developing Power Aware Software*. Technical Report #2003-15, UCSB Computer Science Department, 2003.
- [Gurun+04] Gurun, S., Krintz, C. and Wolski, R. "NWSLite: A Light-Weight Prediction Utility for Mobile Devices". In *Proceedings of 2nd ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'04)*, 2004.
- [Haid+03] Haid, J., Weiss, R., Schoegler, W. and Manninger, M. "Design of an Energy-Aware System-in-Package for Playing MP3 in Wearable Computing Devices". In *Proceedings of IEEE International SOC Conference (SOCC03)*, Portland, Oregon, September 2003.
- [Harris+04] Harris, A., Sengul, C., Kravets, R. and Ratanchandani, P. "Energy-Efficient Multimedia Communications in Lossy Multi-hop Wireless Networks". In *Proceedings of 6th IFIP IEEE International Conference on Mobile and Wireless Communication Networks (MWCN)*, October 2004.
- [Hassmann+02] Hansmann, U., Mettälä, R., Purakayastha, A., Thompson, P. and Kahn, P. *SyncML®: Synchronizing and Managing Your Mobile Data*. Prentice Hall PTR, September 2002.
- [Havinga00] Havinga, P.J.M. *Mobile Multimedia Systems*. PhD thesis, University of Twente, February 2000.
- [Heath+02] Heath, T., Pinheiro, E., Hom, J., Kremer, U. and Bianchini, R. "Application transformations for energy and performance-aware device management". In *Proceedings of the Eleventh Conference on Parallel Architectures and Compilation Techniques (PACT'02)*, September 2002.
- [Hemant+99] Hemant, S. and Badrinath, B. R. "Conflict Resolution and Reconciliation in Disconnected Databases". *DEXA Workshop*, 1999.
- [Hill+04] Hill, D. et al. "Occasionally Connected Smart Clients". In *Smart Client Architecture and Design Guide*, chapter 4, MSDN (Microsoft Developer Network), Microsoft Corporation, June 2004. Available at <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag/html/SCAG-CH04.asp>
- [Hitachi07] Hitachi Global Storage Technologies Home Page. *Deskstar 7K80 hard disk drives specifications*. Available at <http://www.hgst.com/hdd/support/7k80/7k80.htm>
- [Hodges+03] Hodges, D. A., Saleh, R. and Jackson, H. G. *Analysis and Design of Digital Integrated Circuits in Deep Submicron Technology*. McGraw Hill, 3rd

- edition, July 2003.
- [Holliday+02] Holliday, J., Agrawal, D. and El Abbadi, A. "Disconnection modes for mobile databases". *Wireless Networks*, vol. 8, Issue 4, Kluwer Academic Publishers, July 2002.
- [Holt05] Holt, K. "Wireless LAN: Past, Present, and Future". In *Proceedings of the Design, Automation and Test in Europe (DATE'05)*, Vol. 3, 2005, pp. 92-93.
- [Horstmann+97] Horstmann, M. and Kirtland, M. *DCOM Architecture*. Microsoft Developer Network (MSDN), July 1997. Available at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/msdn_dcomhb.asp
- [Huang+03] Huang, Y. and Venkatasubramanian, N. "Supporting Mobile Multimedia Services with Intermittently Available Grid Resources". In *Proceedings of the 10th International Conference High Performance Computing (HiPC 2003)*, December 2003, pp. 238-247.
- [Huang+05] Huang, Y., Mohapatra, S. and Venkatasubramanian, N. "An Energy-Efficient Middleware for Supporting Multimedia Services in Mobile Grid Environments". In *Proceedings of International Symposium on Information Technology: Coding and Computing (ITCC 2005)*, Vol. 2, April 2005, pp.220-225.
- [Hwang+97] Hwang, C.-H. and Wu, A.C. "A Predictive System Shutdown Method for energy Saving of Event-Driven Computation". In *Proceedings of International Conference on Computer-Aided Design, 1997*, pp. 28-32.
- [IBM07] IBM Software. "Rational Unified Process". Available at <http://www-306.ibm.com/software/awdtools/rup/>, 2007.
- [IDC] IDC. *Internacional Data Corporation Home Page*. Available at <http://www.idc.com>.
- [IEEE80211] IEEE. *IEEE 802.11 Wireless Local Area Networks - The Working Group for WLAN Standards*. Available at <http://www.ieee802.org/11/>
- [IEEE80215] IEEE. *IEEE 802.15 WPAN™ Task Group 4 (TG4)*. Available at <http://ieee802.org/15/pub/TG4.html>
- [IEEE80216] IEEE. *The IEEE 802.16 Working Group on Broadband Wireless Access Standards*. Available at <http://www.ieee802.org/16/>
- [IEEE80220] IEEE. *IEEE 802.20 Mobile Broadband Wireless Access (MBWA)*. Available at <http://www.ieee802.org/20/>
- [IETF] IETF. "IP Mobility Support". RFC 2002. Available at

- <http://www.ietf.org/rfc/rfc2002.txt>
- [IETFa] IETF. *Mobile Ad-hoc Networks (MANET) Charter*. Available at <http://www.ietf.org/html.charters/manet-charter.html>
- [Ilyas+04] Ilyas, M. and Mahgoub, I. *Mobile Computing Handbook*. Auerbach publications, CRC Press, June 2004.
- [Intek07] *Programa INTEK del Departamento de Industria, Comercio y turismo del Gobierno Vasco*. Disponible en <http://www.spri.es/>, 2007.
- [Intel03] Intel. *Effects of wireless mobile technology on employee productivity*. Intel Corporation, 2003.
- [Intel03a] Intel. *Intel ® Mobile Application Architecture Guide*. Intel Corporation, 2004.
- [Intel04] Intel. *Mobilizing your field force: The next wave in business transformation*. Intel Corporation, 2004.
- [Intel07] Intel. *Intel QuickStart Technology*. Available at <http://www.intel.com/cd/products/services/EMEA/ENG/notebook/processors/index.htm>, 2007.
- [Irani+05] Irani, S., Singh, G., Shukla, S. and Gupta, R. "An overview of the competitive and adversarial approaches to designing dynamic power management strategies". *IEEE Transactions on Very Large Scale Integration (VLSI)*, Vol.13, No.12, 2005, pp. 1349-1361.
- [IRDA07] IrDA. *The Infrared Data Association Home Page (IrDA)*. Available at <http://www.irda.org>, 2007.
- [Irmak+05] Irmak, U. and Suel, T. "Hierarchical substring caching for efficient content distribution to low-bandwidth clients". In *Proceedings of the 14th international conference on World Wide Web (WWW 2005)*, 2005, pp. 43-53.
- [Israelsohn01] Israelsohn, J. "Battery Management Included". *EDN Magazine*, 2001, pp. 65-74. Available at www.ednmag.com/ednmag/reg/2001/01182001/02cs.htm
- [IWE-X06] IWE-X. "Estadísticas wireless en España – Octubre 2006". Disponible en http://www.gowex.com/images/observatorio/informe_6_2006-10.pdf, 2006.
- [Iyer+03] Iyer, S., Luo, L., Mayo, R. and Ranganathan, P. "Energy-Adaptive Display System Designs for Future Mobile Environments". In *Proceedings of the First International Conference on Mobile Systems, Applications, and*

- Services*, San Francisco, CA (USA), May 2003, pp. 245 -258.
- [Jermaine+06] Jermaine, C., Omiecinski, E. and Yee, W. G. "The Partitioned Exponential File for Database Storage Management". *VLDB Journal*, Springer, 2006.
- [Jing+99] Jing, J., Helal, A. S. and Elmagarmid, A. "Client-Server Computing in Mobile Environments". *ACM Computing surveys*, Vol. 31, No. 2, June 1999, pp. 117-157.
- [Jones+01] Jones, C. E., Sivalingam, K.M., Agrawal, P. and Chen, J. "A survey of energy efficient network protocols for wireless networks". *Wireless Networks*, Vol. 7, No. 4, 2001, pp. 343-358.
- [Johnston+04] Johnston, D.B., Perkins, C. and Arkko, J. *Mobility Support in IPv6 – RFC 3775*, IETF, June 2004.
- [Joseph+95] Joseph, A., deLespinasse, A., Tauber, J., Gifford, D. and Frans Kaashoek, M. "Rover: A Toolkit for Mobile Information Access". *In Proceedings of the Fifteenth ACM Symposium on Operating System Principles (SOSP)*, December 1995.
- [Karl03] Karl, H. *An Overview of Energy-Efficiency Techniques for Mobile Communication Systems*. Technical Report, TKN Technical Report Series, Telecommunication Networks Group, Technical University Group, September 2003.
- [Karlin+94] Karlin, A., Manasse, M., McGeoch, L. and Owichki, S. "Competitive Randomized Algorithms for Nonuniform Problems". *Algorithmica*, Vol. 11, No. 6, June 1994, pp. 542-571.
- [Karygiannis+02] Karygiannis, T. and Owens, L. *Wireless Network Security. 802.11, Bluetooth and Handheld Devices*. NIST Special Publication 800-48, National Institute of Standards and Technology (NIST), November 2002. Available at http://csrc.nist.gov/publications/nistpubs/800-48/NIST_SP_800-48.pdf
- [Kendall05] Kendall, P. *WiMAX - "Last Mile", WiFi Backhaul, Fixed Broadband Wireless Standard, or 3G Killer?*, Industry Report, Strategy Analytics, 2005. Available at <http://www.strategyanalytics.net/default.aspx?mod=ReportAbstractViewer&a0=2336>
- [Kien+05] Kien T. N., Erradi A. and Maheshwari P. "WSMB: a middleware for enhanced web services interoperability". *In Proceedings of First International Conference on Interoperability of Enterprise Software and Applications (Interop-ESA'05)*, 2005.
- [Kovari+01] Kovari, P., Van Acker, B., Marino, A., Ryan, J., Lun Tang, K. and Weiss,

- C. *Mobile Applications with IBM WebSphere Everyplace Access Design and Development*. IBM Redbook, November 2001, pp. 4. Available at <http://www.redbooks.ibm.com/abstracts/SG246259.html?Open>
- [Krashinsky+05] Krashinsky, R. and Balakrishnan, H. "Minimizing energy for wireless web access with bounded slowdown". *Wireless Networks*, Vol. 11, No. 1-2, 2005, pp. 135-148.
- [Kravets+00] Kravets, R. and Krishnan, P. "Application-driven power management for mobile communication". *Wireless Networks*, Vol. 6, No. 4, 2000, pp. 263 - 277.
- [Krunchten03] Krunchten, P. *The Rational Unified Process: An Introduction*. 3rd ed. Pearson Education Inc., 2003.
- [Kuhn+02] Kuhn, R., Tracy, M. and Frankel, S. *Security for Telecommuting and Broadband Communications*. NIST Special Publication 800-46, National Institute of Standards and Technology (NIST), August 2002. Available at http://csrc.nist.gov/publications/nistpubs/800-46/NIST_SP_800-46.pdf
- [Kunz+05] Kunz, T., Gaddah, A. "Adaptive Mobile Applications". *Encyclopedia of Information Science and Technology (I)*, 2005, pp. 47-52.
- [Kurose+04] Kurose, J.F. y Ross, K.W. *Redes de computadores. Un enfoque descendente basado en Internet. Segunda edición*. Pearson Educación, S.A., 2004.
- [Lago+05] Lago, A.B., Larizgoitia, I., Vázquez, I. y Bañales, R. "Desarrollo de la plataforma de servicios WALLIP". *Actas de las Jornadas Científico-Técnicas de Servicios Web (JSWEB 2005) de la W3C*, September 2005.
- [Lago+06] Lago, A.B., Larizgoitia, I. y Vázquez, I. "WALLIP: Desarrollo de una plataforma de servicios de valor añadido". *Actas de la I Conferencia Ibérica de Sistemas y Tecnologías de la Información (CISTI 2006)*, Portugal, 2006.
- [Lago+06a] Lago, A.B., Larizgoitia, I. and Trueba, R. "An application-aware approach to mobile operation optimization". In *Proceedings of 2nd International Workshop on Ubiquitous Computer & Ambient Intelligence (WUCAmI'06)*, November 2006.
- [Lago+06b] Lago, A.B., Larizgoitia, I. y Trueba, R. "Middleware para la movilización de servicios empresariales: MOSE". *Actas del IV Congreso Iberoamericano de Telemática (CITA 2006)*, Mayo 2006.
- [Lago+06c] Lago, A.B., Larizgoitia, I. y Vázquez, I. "Plataformas de servicios WALLIP". *Actas del II Congreso Iberoamericano sobre Computación ubicua (CICU 2006)*, Junio 2006.

- [Lago+06d] Lago, A.B., Larizgoitia, I. y Vázquez, I. "Plataforma de servicios de valor añadido: WALLIP". *Actas del IV Congreso Iberoamericano de Telemática (CITA 2006)*, Mayo 2006.
- [Lara+05] de Lara, E., Chopra, Y., Kumar, R., Vaghela, N., Wallach, D.S. and Zwaenepoel, W. "Iterative Adaptation for Mobile Clients Using Existing APIs". *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, Vol. 16, No. 10, October 2005.
- [Layne99] Layne, P. *Software Architecture for Mobile Computing*. Master Thesis. University of Jyväskylä, 1999.
- [Lavela+99] Lavela, P. and Tirado, J. L. *Baterías avanzadas*. Servicio de Publicaciones de la Universidad de Córdoba, 1999.
- [Liu02] Liu, J. *Mobile Map: A Case Study in the Design & Implementation of a Mobile Application*. Master Thesis. Carleton University, 2002.
- [Lorch+98] Lorch, J. R. and Smith, A. J. "Software strategies for portable computer energy management". *IEEE Personal Communications Magazine*, Vol. 5, No. 3, June 1998, pp.60 - 73.
- [Lorch+99] Lorch, J. R. and Smith, A. J. *Energy Consumption of Apple Macintosh Computers*. Technical report, EECS Department, University of California, Berkeley Berkeley, CA 94720-1776, February 1999.
- [Lorch+03] Lorch, J.R. and Smith, A.J. "Operating System Modifications for Task-Based Speed and Voltage Scheduling". *In Proceedings of the First International Conference on Mobile Systems, Applications, and Services (MobiSys'03)*, Usenix, 2003, pp. 215-230.
- [Loy01] Loy, R.J. *Active Mode Power Management For Mobile Devices*. Master Thesis. University of Florida, 2001.
- [Lu+00] Lu, Y-H., Chung, E-Y., Simunic, T., Benini, L. and De Micheli, G. "Quantitative comparison of power management algorithms". *In Proceedings of the Conference on Design Automation and Test in Europe (DATE'00)*, 2000.
- [Lu+00a] Lu, Y-H., Benini, L. and De Micheli, G. "Operating-system directed power reduction". *In Proceedings of the International Symposium on Low-Power Electronics and Design (ISLPED'00)*, July 2000, pp. 37-42.
- [Lu+01] Lu, Y-H. and De Micheli, G. "Comparing system-level power management policies". *IEEE Design & Test of Computers*. Special issue on Dynamic Power Management of Electronic Systems, March/April 2001,

- pp. 10-19.
- [Madrid04] Madrid, J.M. "Seguridad en redes inalámbricas 802.11". *Sistemas & Telemática*, No. 3, Universidad ICESI, Enero-Junio 2004.
- [Maheshwari+04] Maheshwari, P., Tang, H. and Liang, R. "Enhancing Web Services with Message-Oriented Middleware". In *Proceedings of IEEE International Conference on Web Services (ICSW'04)*, June 2004.
- [Mallick03] Mallick, M. *Mobile and Wireless Design Essentials*. Wiley and Sons Inc., 2003.
- [Martin96] Martin, T. L. and Siewiorek, D. P. "A Power Metric for Mobile Systems". In *Proceedings of International Symposium on Low Power Electronics and Design*, Monterey (California), 1996, pp. 37 - 42.
- [Martin99] Martin, T.L. *Balancing batteries, power, and performance: system issues in CPU speed-setting for mobile computers*. PhD thesis, Carnegie Mellon University, 1999.
- [Martin+03] Martin, T. and Siewiorek, D. "Non ideal Battery Properties and Their Impact on Software Design for Wearable Computers". *IEEE Transactions on Computers*, August 2003, pp. 79-84.
- [Martin+03a] Martin, T., Siewiorek, D., Smailagic, A., Bosworth, M., Ettus, M. and Warren, J. "A case study of a system-level approach to power-aware computing". *ACM Transactions in Embedded Computing Systems*. Vol.2, No.3, 2003, pp. 255-276.
- [Martin+04] Martin, T., Siewiorek, D.P., Smailagic, A. and Warren, J. "Power Management for Mobile Computers". In *Mobile Computing Handbook*, I. Mahgoub and M. Ilyas (Eds.), CRC Press, 2004, pp. 709-730.
- [MetroBilbao07] Metro Bilbao. *Página oficial de Metro Bilbao*. Disponible en <http://www.metrobilbao.com>, 2007.
- [Microsoft06] Microsoft. *SQL Server Mobile Edition Books Online*. Microsoft Corporation, 2006. Available at <http://www.microsoft.com/technet/prodtechnol/sql/2005/technologies/sqlmobile.mspx>
- [Microsoft06a] Microsoft. *SQL Server 2005 Books Online*. Microsoft Corporation, April 2006. Available at <http://www.microsoft.com/technet/prodtechnol/sql/2005/library/techref.mspx>
- [Microsoft06b] Microsoft. *Advantages of Microsoft Merge Replication for Mobile and*

- Distributed Applications*. Microsoft Corporation, February 2006.
- [Miyoshi+02] Miyoshi, A., Lefurgy, C., Hensbergen, E. V., Rajamony, R. and Rajkumar, R. "Critical powerslope: Understanding the runtime effects of frequency scaling". In *Proceedings of International Conference on Supercomputing*, June 2002, pp. 35–44.
- [Mnaouer+04] Mnaouer, A., Shekhar, A. and Liang, Z. "A Generic Framework for Rapid Application Development of Mobile Web Services with Dynamic Workflow Management". In *Proceedings of the 2004 IEEE International Conference on Services Computing (SCC'04)*, IEEE CS Press, 2004, pp. 165–171.
- [Mockocki+07] Mochocki, B., Rajan, D., Sharon, X., Poellabauer, C., Otten, K. and Chantem, T. "Network-Aware Dynamic Voltage and Frequency Scaling". In *Proceedings of the 13th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'07)*, April 2007.
- [Modiano99] Modiano, E. "An adaptive algorithm for optimizing the packet size in wireless ARQ protocols". *Wireless Networks*, Vol. 5, 1999, pp. 279-289.
- [Mohapatra+02] Mohapatra, S. and Venkatasubramanian, N. *Optimizing Power using Reconfigurable Middleware*. Technical Report DSM_TR03-04, CSN, University of California, Irvine, Nov 2002.
- [Mohapatra+03] Mohapatra, S. and Venkatasubramanian, N. "PARM: Power Aware Reconfigurable Middleware". In *Proceedings of IEEE International Conference on Distributed Computer Systems (ICDCS-23)*, 2003.
- [Mohapatra+03a] Mohapatra, S., Cornea, R., Dutt, N., Nicolau, A. and Ventakatasubramanian, N. "Integrated Power Management for Video Streaming to Mobile Handheld Devices". In *Proceedings of the Eleventh ACM International Conference Multimedia (MM'03)*, Berkeley (California), November 2003, pp. 582-591.
- [Mountain+04] Mountain, H.M. et al. *SOAP Version 1.2 Email Binding*. W3C Note, July 3, 2002. Available at <http://www.w3.org/TR/soap12-email>
- [Multinet07] Multinet Consortium. *Multinet Home Page*. Available at <http://www.ist-multinet.org/project.htm>.
- [Myers97] Myers, M.D. "Qualitative Research in Information Systems". *MIS Quarterly* (21:2), June 1997, pp. 241-242. www.misq.org/discovery/MISQD_isworld/
- [Natis03] Natis, Y.V. *Service-Oriented Architecture Scenario*. Gartner, April 2003.

- Available at <http://www.gartner.com/DisplayDocument?id=391595>.
- [Neugebauer+01] Neugebauer, R. and McAuley, D. "Energy is just another resource: Energy accounting and energy pricing in the Nemesis OS". In *Proceedings of the Eighth Workshop on Hot Topics in Operating Systems (HotOS'01)*, May 2001.
- [NIST06] National Institute of Standards and Technology (NIST). *Computer Security Division - 2005 Annual Report*. NIST IR 7285, February 2006. Available at <http://csrc.nist.gov/publications/nistir/7285/nistir-7285-CSD-2005-Annual-Report.pdf>
- [Noble98] Noble, B. D. *Mobile Data Access*. PhD thesis, Carnegie Mellon University, May 1998. Available at <http://www.cs.cmu.edu/afs/cs/project/coda/Web/coda.html>, CMU-CS-98-118
- [Nokia06] Nokia. "Nokia for Business City, soluciones de movilidad para la empresa en un entorno real". Disponible en <http://www.nokia.com/A4213038?newsid=-1823>, Noviembre de 2006.
- [Oasis02] Oasis. *UDDI Version 2.04 API Specification*. UDDI Committee Specification, July 2002. Available at <http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm>.
- [OMG04] Object Management Group. *Common Object Request Broker Architecture: Core Specification Version 3.0.3*, OMG, March 2004. Available at <http://www.omg.org/docs/formal/>.
- [Olsen+06] Olsen, C.M. and Narayanaswami, C. "PowerNap: An Efficient Power Management Scheme for Mobile Devices". *IEEE Transactions on Mobile Computing*, Vol. 5, No. 7, July 2006, pp. 816-828. [syncmlindex.html#V101](#)
- [Olson+03] Olson, M., and Ogbuji, U. *Sending and Receiving SOAP Requests over SMTP*. March 2003. Available at <http://www-106.ibm.com/developerworks/webservices/library/wspyth12.html>.
- [OMA01] Open Mobile Alliance. *SyncML Sync Protocol, version 1.0.1*. June 2001. Available at http://www.syncml.org/docs/syncml_protocol_v101_20010615.pdf
- [OMA01a] Open Mobile Alliance. *SyncML Representation Protocol, version 1.0.1*. June 2001. Available at http://www.syncml.org/docs/syncml_represent_v101_20010615.pdf

- [OMA07] Open Mobile Alliance. *SyncML Initiative*. Available at <http://www.openmobilealliance.org/tech/affiliates/syncml/>, 2007.
- [Oracle05] Oracle. "Oracle® Database Lite Administration and Deployment Guide 10g (10.2.0)". Part No. B15921-01, Oracle, 2005. Available at <http://www.oracle.com/technology/documentation/database10gr2.html>
- [Oracle05a] Oracle. "Oracle® Database Lite Developer's Guide 10g (10.2.0)". Part No. B15920-01, Oracle, 2005. Available at <http://www.oracle.com/technology/documentation/database10gr2.html>
- [Oracle05b] Oracle. "Oracle® Database Concepts 10g Release 2 (10.2)". Part No. B14220-02, Oracle, 2005. Available at <http://www.oracle.com/technology/documentation/database10gr2.htm>
- [Oracle05c] Oracle. "Oracle® Database Lite 10g. Technical White Paper". Oracle, May 2005.
- [Oracle05d] Oracle. "Oracle® Database Lite 10g. Business White Paper". Oracle, May 2005.
- [Ort05] Ort, E. *Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools*. Sun, April 2005. Available at <http://java.sun.com/developer/technicalArticles/WebServices/soa2/>
- [Paakkonen+05] Paakkonen, P., Pakkala, D. and Sihvonen, M. "An Optimized Message-Oriented Middleware Solution for Extending Enterprise Services to the Mobile Domain". In *Proceedings of Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services (ICAS-ICNS'05)*, October 2003, pp. 23.
- [Papazoglou+03] Papazoglou, M.P. and Georgakopoulos, D. "Service-Oriented Computing". *Communications of the ACM*, Vol. 46, No. 10, October 2003.
- [Patel+06] Patel, K., Benini, L., Macii, E. and Poncino, M. "STV-Cache: a leakage energy-efficient architecture for data caches". In *Proceedings of ACM Great Lakes Symposium on VLSI 2006*, 2006, pp. 404-409.
- [Peddemors+04] Peddemors, A., Zandbelt, H. and Bargh, M. "A Mechanism for Host Mobility Management supporting Application Awareness". In *Proceedings of the Second International Conference on Mobile Systems, Applications, and Services (MobiSys'04)*, June 2004.
- [Peddemors+05] Peddemors, A., Eertink, H. and Niemegeers, I. "Communication Context for Adaptive Mobile Applications". In *Workshop Proceedings of the 3rd Conference on Pervasive Computing - Middleware Support for Pervasive*

- Computing Workshop (PerWare'05)*, March 2005.
- [Pettis+06] Pettis, N., Cai, L. and Lu, Y.-H. "Statistically Optimal Dynamic Power Management for Streaming Data". *IEEE Trans. Computers*, No. 55, Vol. 7, 2006, pp. 800-814.
- [Pettis+06a] Pettis, N., Ridenour, J. and Lu, Y.-H. "Automatic run-time selection of power policies for operating systems". In *Proceedings of the conference on Design, automation and test in Europe (DATE 2006)*, 2006, pp. 508-513.
- [Perkins+03] Perkins, C., Belding-Royer, E. and Das, S. *Ad hoc On-Demand Distance Vector (AODV) Routing*. RFC 3561, July 2003. Available at <http://www.ietf.org/rfc/rfc3561.txt>.
- [Poellabauer+04] Poellabauer, C. and Schwan, K. "Energy-aware traffic shaping for wireless real-time applications". In *Proceedings of the Tenth Real-Time and Embedded Technology and Applications Symposium (RTAS'04)*, May 2004.
- [Ponzo+04] Ponzo, J., Hasson, L. D., George, J., Thomas, G., Gruber, O., Konuru, R., Purakayastha, A., Johnson, R. D., Colson, J and Pollak, R. A. "On Demand Web-client technologies". *IBM Systems Journal*, Vol. 43, No. 2, 2004.
- [Pouwelse03] Pouwelse, J. A. *Power Management for Portable Devices*. Phd. Thesis. Technische Universiteit Delft, October 2003.
- [Przybilski05] Przybilski, M. *Distributed Processing and Energy Awareness*. Technical Report. Helsinki Institute for Information Technology Basic Research Unit, 2005.
- [Rangarajan+03] Rangarajan, A., Sharma, V. and Singh, S. K. "Information-theoretic and Communication-theoretic power allocation for fading channels". In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2003.
- [Ratner+01] Ratner, D., Reiher, P., Popek, G.J. and Kuenning, G.H. "Replication requirements in mobile applications". *Mobile Networks and Applications*, Vol. 6, Issue 6, November 2001, pp. 525 – 533.
- [Rodriguez+05] Rodriguez, J.R. et al. *IBM Workplace Client Technology Micro Edition Version 5.7.1: Application Development and Case Study*. IBM Redbook, June 2005. Available at <http://www.redbooks.ibm.com>
- [Rodriguez+05a] Rodriguez, J.R. et al. *IBM WebSphere Everyplace Access V5 Handbook for Developers and Administrators Volume III: E-Mail and Database Synchronization*. IBM Redbook, April 2005. Available at

<http://www.redbooks.ibm.com>

- [Rosu+04] Rosu, M.C., Olsen, C. M., Narayanaswami, C. and Luo, L. "PAWP: A Power Aware Web Proxy for Wireless LAN Clients". In *Proceedings of Sixth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'04)*, 2004, pp. 206-215.
- [Rudenko+98] Rudenko, A., Reiher, P., Popek, G. and H. Kuenning, G. *The Remote Processing Framework for Portable Computer Power Saving*. ACM, 1998.
- [Ruiz+02] Ruiz, F., Polo, M. y Piattini, M. "Utilización de Investigación-Acción en la Definición de un Entorno para la Gestión del Proceso de Mantenimiento del Software". *Actas del I Workshop en Métodos de Investigación y Fundamentos Filosóficos en Ingeniería del Software y Sistemas de Información (MIFISIS'2002)*, El Escorial (España), Noviembre 2002, pp. 48-64.
- [Sailhan+04] Sailhan, F. and Issarny, V. "Energy-aware Web Caching over Hybrid Networks". In *Mobile Computing Handbook*, I. Mahgoub and M. Ilyas (Eds), CRC Press, 2004, pp. 779-804.
- [SAMSUNG07] Samsung Home Page. Available at http://www.samsung.com/Products/HardDiskDrive/SpinPointVSeries/HardDiskDrive_SpinPointVSeries_SV0802N.asp?page=Specifications
- [Sankar04] Sankar, R. "Routing and Mobility Management in Wireless Ad Hoc Networks". In *Mobile Computing Handbook*, I. Mahgoub and M. Ilyas (Eds), CRC Press, 2004, pp. 637-662.
- [Satyanarayanan96] Satyanarayanan, M. *Fundamental Challenges in Mobile Computing*. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1996.
- [Satyanarayanan01] Satyanarayanan, M. "Pervasive Computing: Vision and Challenges", *IEEE Personal Communications*, August 2001, pp. 10-17.
- [Satyanarayanan02] Satyanarayanan, M. "The Evolution of Coda". *ACM Transactions on Computer Systems*, Vol. 20, No. 2, May 2002.
- [Saygin04] Saygin, Y. "Hoarding in Mobile Computing Environments". In *Mobile Computing Handbook*, I. Mahgoub and M. Ilyas (Eds), CRC Press, 2004, pp. 389-402.
- [Schurgers03] Schurgers, C., Raghunathan, V. and Srivastava, M.B. "Power Management for Energy-Aware Communications Systems". *ACM Transactions on Embedded Computing Systems*, Vol. 2, No. 3, August 2003, pp. 431-447.

- [ServiceArchi] *Web Services and Service-Oriented Architectures Home Page*. Available at <http://www.service-architecture.com/index.html>.
- [Shapiro+04] Shapiro, M., Preguiça, N. M. and O'Brien, J. "Rufis: Mobile Data Sharing Using a Generic Constraint-Oriented Reconciler". In *Proceedings of IEEE International Conference on Mobile Data Management (MDM04)*, 2004.
- [Sharma+04] Sharma, V. and Chockalingam, A. "Battery Power Management in Portable Devices". In *Mobile Computing Handbook*, I. Mahgoub and M. Ilyas (Eds), CRC Press, 2004, pp. 893-920.
- [Shen+05] Shen, J., Sun, P., Yin, Y. and Song, S. "Delivering Mobile Enterprise Applications on iMMS Framework", In *Proceedings of ACM 6th International Conference on Mobile Data Management (MDM05)*, 2005.
- [Schultz03] Schultz, F. *Synchronisation von Informationsobjekten zwischenheterogenen Systemen*. Master Thesis. University of Harburg, 2003.
- [Schneider+04] Schneider, H., Lee, V. and Schell, R. *Mobile Applications: Architecture, Design, and Development*. Prentice Hall PTR, April 2004.
- [Shih+02] Shih, E., Bahl, P. and Sinclair, M.J. "Wake on wireless: An event driven energy saving strategy for battery operated devices". In *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking (MOBICOM'02)*, September 2002.
- [Smit+97] Smit, G.J.M. and Havinga, P.J.M. *A survey of energy saving techniques for mobile computers*. Technical Report, 1997, Available at <http://www.cs.utwente.nl/~havinga/papers/energy.ps>.
- [Simunic+00] Simunic, T., Benini, L. and De Micheli, G. "Dynamic Power Management for Portable Systems". In *Proceedings of the 6th International Conference in Mobile Computing and Networking (MOBICOM'00)*, ACM, Boston, Mass., 2000, pp. 11-19.
- [Simunic+00a] Simunic, T., Vikalo, H., Glynn, P. and De Micheli, G. "Energy Efficient Design of Portable Wireless Systems". In *Proceedings of the International Symposium on Low Power Electronics and Design*, 2000, pp. 49-54.
- [Simunic01] Simunic, T. *Energy Efficient System Design and Utilization*. PhD. Thesis, Stanford University, 2001.
- [Simunic02] Simunic, T. "Dynamic management of power consumption". In *Power Aware Computing*, R. Graybill and R. Melhem (Eds.), Kluwer Academic Publishers, 2002.

Referencias Bibliográficas

- [Simunic05] Simunic, T. "Power Saving Techniques for Wireless LANs". In *Proceedings of the conference on Design, Automation and Test in Europe*, Vol. 3, 2005, pp. 96-97.
- [Simunic+05] Simunic, T., Quadeer, W. and De Micheli, G. "Managing heterogeneous wireless environments via Hotspot servers". In *Proceedings of Twelfth Annual Multimedia Computing and Networking (MMCN'05)*, 2005, pp. 110-123.
- [Sivalingam+00] Sivalingam, K.M., Chen, J-C., Agrawal, P. and Srivastava, M. B. "Design and analysis of low-power access protocols for wireless and mobile ATM networks". *ACM Journal of Wireless Networks*. Vol. 6, No, 1, 2000, pp. 73-87.
- [Snowdon+05] Snowdon, D.C., Petters, S.M. and Heiser, G. "Power measurement as the basis for power management". In *Proceedings of the 2005 Workshop on Operating System Platforms for Embedded Real-Time Applications*, July 2005.
- [Soriano+06] Soriano, J., Jimenez, M., Cantera, J. M. and Hierro, J. J. "Delivering Mobile Enterprise Services on Morfeo's MC Open Source Platform". In *Proceedings of the 7th International Conference on Mobile Data Management (MDM'06)*, Vol. 0, pp.139, IEEE Computer Society, 2006.
- [Sousa+02] Sousa, J. and Garlan, D. "Aura: an architectural framework for user mobility in ubiquitous computing environments". In *Proceedings of the third Working IEEE/IFIP Conference on Software Architecture*, 2002, pp.29-43.
- [Srivastava+96] Srivastava, M.B., Chandrakasan, A.P. and Brodersen, R.W. "Predictive System Shutdown and Other Architecture Techniques for Energy Efficient Programmable Computation". *IEEE Transactions on VLSI Systems*, Vol. 1, March 1996, pp. 42-55.
- [Stemm+96] Stemm, M., Gautier, P., Harada, D. and Katz, R.H. "Reducing power consumption of network interfaces in hand-held devices". In *Proceedings of International Workshop on Mobile Multimedia Communications (MoMUC-3)*, Princeton, October 1996.
- [Straub+04] Straub, D., Gefen, D. and Boudreau, M.C. *The ISWorld Quantitative, Positivist Research Methods Website*, 2004. Available at <http://dstraub.cis.gsu.edu:88/quant/>
- [Sybase] Sybase Web site. *SQL Anywhere Studio 9.0.2. Product Manual*. Available at <http://sybooks.sybase.com/nav/detail.do?docset=766>.
- [Sybase04] Sybase. *Introducing SQL Anywhere® Studio*. Part number: DC38120-01-

- 0902-01, Sybase, Inc., October 2004.
- [SyncML] *SyncML Web site*. Open Mobile Alliance. Available at <http://www.syncml.org>
- [Tan04] Tan, K-L. "Energy Efficient Selective Cache Invalidation". In *Mobile Computing Handbook*, I. Mahgoub and M. Ilyas (Eds.), CRC Press, 2004, pp.421-432.
- [Telefonica04] Telefónica I+D. "Trabajar y vivir en la red". Documento online. Febrero 2004. Disponible en http://www.tid.es/documentos/libros_sector_telecomunicaciones/trabajar_vivir_red.pdf
- [Telefonica06] Telefónica I+D. "TIDMOBILE. Manual de usuario del microservidor web local para PDA". Informe técnico. 2006. Disponible en http://www.morfeo-project.org/files/TIDMobile_DeviceManager_UserGuide.pdf
- [Telefonica06a] Fundación Telefónica. "La Sociedad de la Información en España 2005". Documento online. 2006. Disponible en <http://www.telefonica.es/index/informeSI2006.html>
- [Terry+03] Terry, D.B. and Ramasubramanian, V. "Caching XML Web Services for Mobility". *ACM Queue*, Vol. 1, No. 1, March 2003.
- [Tiwari+96] Tiwari, V., Malik, S., Wolfe, A. and Lee, M.T.-C. "Instruction level power analysis and optimization of software". *VLSI Design*. IEEE Computer Society, 1996, pp. 326-328.
- [Trueba+06] Trueba, R., Lago, A.B. y Larizgoitia, I. "Una solución para la movilización de servicios empresariales: plataforma MOSE". *Actas del II Congreso Iberoamericano sobre Computación ubicua (CICU 2006)*, Junio 2006.
- [Truman+98] Truman, T.E., Pering, T., Doering, R., Brodersen, R.W. "The infopad multimedia terminal: a portable device for wireless information access". *IEEE Transactions on Computers*, Vol. 47, No. 10, 1998, pp.1073-1087.
- [UMTSForum02] UMTS Forum. *Key Components for 3G Devices*. Report No. 15 UMTS Forum, 2002.
- [UMTSForum05] UMTS Forum. "3G/UMTS. Towards mobile broadband and personal Internet". October 2005. Available at http://www.umts-forum.org/servlet/dycon/ztumts/umts/Live/en/umts/MultiMedia_PDFs_Papers_Towards-Mobile-Broadband-Oct05.pdf
- [UMTSTDD07] *The Global UMTS TDD Alliance Home Page*. Available at

- <http://www.umtstd.org/>, 2007.
- [Vahdat+00] Vahdat, A., Lebeck, A. and Ellis, C.S. "Every joule is precious: A case for revisiting operating system design for energy efficiency". In *Proceedings of the Ninth ACM SIGOPS European Workshop 2000*, September 2000.
- [Vazquez+05] Vázquez Gómez, I., Lago Vilariño, AB., Larizgoitia Abad, I., Uria, I. y Bañales, R. "WALLIP: Una plataforma de servicios con percepción contextual basada en localización". *Actas de Simposio de Computación Ubicua e Inteligencia Ambiental (UCAmI'2005) dentro del I Congreso Español de Informática*, Septiembre 2005.
- [Valluri+01] Valluri, M. and John, L. "Is Compiling for Performance == Compiling for Power?". In *Proceedings of the 5th Annual Workshop on Interaction between Compilers and Computer Architectures (INTERACT-5)*, Monterrey, Mexico, January 2001.
- [Vijaykrishnan+00] Vijaykrishnan, N., Kandemir, M., Irwin, M.J., Kim, H. and Ye, W. "Energy-driven Integrated Hardware-Software Optimizations using SimplePower". In *Proceedings of the 27th Annual International Symposium on Computer Architecture (ISCA'00)*, June 2000, pp. 95-106.
- [W3C03] World Wide Web Consortium (W3C). *SOAP Version 1.2 Part 1: Messaging Framework*. June 2003. Available at <http://www.w3.org/TR/soap12-part1/>
- [W3C04] World Wide Web Consortium (W3C). *Web Services Architecture*. February 2004. Available at <http://www.w3.org/TR/ws-arch/>
- [W3C04a] World Wide Web Consortium (W3C). *SOAP Message Transmission Optimization Mechanism*. W3C Candidate Recommendation, 26 August 2004. Available at <http://www.w3.org/TR/soap12-mtom/>
- [W3C05] World Wide Web Consortium (W3C). *XML-binary Optimized Packaging*. W3C Recommendation, 25 January 2005. Available at <http://www.w3.org/TR/xop10/>.
- [W3C06] World Wide Web Consortium (W3C). *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*. March 2006. Available at <http://www.w3.org/TR/wsdl20/>.
- [Wadsworth98] Wadsworth, Y. What is participatory Action Research?. Action Research International, 1998. Available at www.scu.edu.au/schools/gcm/ar/ari/p-ywadsworth98.html
- [Weinlein05] Weinlein, T. *Application-Specific Energy Management in Operating Systems*. Diploma Thesis. Operating System Group, University of Erlangen,

- January 2005.
- [Weisser+94] Weiser, M., Welch, B., Demers, A. and Shenker, S. "Scheduling for reduced CPU energy". In *Proceedings of the First USENIX Symposium on Operating Systems Design and Implementation*. November 1994, pp. 13-23.
- [Weissel+02] Weissel, A., Beutel, B. and Bellosa, F. "Cooperative I/O: a novel I/O semantics for energy-aware applications". In *Proceedings of the 5th symposium on Operating systems design and implementation (OSDI'02)*, ACM SIGOPS Operating Systems Review, Vol. 36 , Issue SI (Winter 2002), 2002, pp. 117-129.
- [Weissel+04] Weissel, A., Faerber, M. and Bellosa, F. "Application characterization for wireless network power management". In *Proceedings of the International Conference on Architecture of Computing Systems (ARCS'04)*, January 2004, pp. 231-245.
- [Werner+04] Werner, C., Buschmann, C. and Fischer, S. "Compressing SOAP Messages by using Differential Encoding". In *Proceedings of the IEEE International Conference on Web Services*, 2004.
- [Werner+06] Werner, C., Buschmann, C., Jäcker, T. and Fischer, S. "Bandwidth and Latency Considerations for Efficient SOAP Messaging". In *International Journal of Web Services Research*, Vol. 3, No. 1, 2006.
- [Wiesmann+00] Wiesmann, M., Pedoney, F., Schiper, A., Kemmez, B. and Alonsoz, G. "Database Replication Techniques: a Three Parameter Classification". In *Proceedings of the 19th IEEE Symposium on Reliable Distributed Systems (SRDS2000)*, 2000.
- [Wiesmann+01] Wiesmann, M., Pedoney, F., Schiper, A., Kemmez, B. and Alonsoz, G. "Understanding Replication in Databases and Distributed Systems". In *Proceedings of the 20th International Conference on Distributed Computing Systems (ICDCS'2000)*, 2000.
- [Wiesmann02] Wiesmann, M. *Group Communications and Database Replication: Techniques, Issues and Performance*. PhD. Thesis, École Polytechnique Fédérale de Lausanne, 2002.
- [Yan+05] Yan, L., Zhong, L. and Jha, N.K. "User-perceived latency driven voltage scaling for interactive applications". In *Proceedings of the 42nd Design Automation Conference (DAC'05)*, June 2005.
- [YANKEE04] Yankee group. *Demystifying Next-Generation Broadband Wireless and the Role of WiMAX*, October 2004. Available at http://www.researchandmarkets.com/reportinfo.asp?report_id=

228292&t=d&cat_id=20

- [Yee+05] Yee, W.G. and Frieder, O. "Scalable synchronization of intermittently connected database clients". In *Proceedings of ACM Mobile Data Management (MDM'05)*, 2005.
- [Yuan+03] Yuan, W. and Nahrstedt, K. "Energy-Efficient Soft Real-Time CPU Scheduling for Mobile Multimedia Systems". In *Proceedings of 19th ACM Symposium on Operating Systems Principles (SOSP'03)*, Bolton Landing, NY, October 2003, pp.149-163.
- [Yuan+06] Yuan, W., Nahrstedt, K., Adve, S.V., Jones, D.L. and Kravets, R.H. "GRACE-1: Cross-Layer Adaptation for Multimedia Quality and Battery Energy". *IEEE Transactions on Mobile Computing*, Vol. 5, No. 7, July 2006, pp. 799-815.
- [Zachariadis+02] Zachariadis, S., Capra, L., Mascolo, C. and Emmerich, W. "XMIDDLE: information sharing middleware for a mobile environment". In *Proceedings of the 24th International Conference on Software Engineering*, 2002.
- [Zeng+02] Zeng, H., Ellis, C. S., Lebeck, A. R. and Vahdat, A. "ECOSystem: Managing Energy as a First Class Operating System Resource". In *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (APLOS'02)*, October 2002, pp. 123 - 132.
- [Zeng+03] Zeng, H., Ellis, C. S., Lebeck, A. R. and Vahdat, A. "Currentcy: A Unifying Abstraction for Expressing Energy Management Policies". In *Proceedings of the 2003 USENIX Annual Technical Conference*, June 2003, pp. 43-56.
- [Zeng+05] Zeng, H., Ellis, C.S. and Lebeck, A.R. "Experiences in managing energy with ecosystem". *IEEE Pervasive Computing*, Vol. 4, No.1, January 2005, pp. 62-68.
- [Zheng+05] Zheng, R. and Kravets, R. "On-demand Power Management for Ad Hoc Network". *Elsevier Ad Hoc Networks Journal*, Vol. 3, No. 1, 2005, pp 51-68.
- [Zhong05] Zhong, L. *Energy-Efficient Mobile System Design: The User's Perspective*. PhD. Thesis. Faculty of Princeton University, September 2005.
- [Zhou+04] Zhou, D., Islam, N. and Ismael, A. "Flexible On-Device Service Object Replication with Replets". In *Proceedings International WWW Conference*, New York, USA, May 2004.

Apéndice A

En este apéndice se detalla el cuestionario realizado a empresas dentro del estudio sobre las necesidades de movilidad empresariales.

I-BLOQUE DE INFORMACIÓN SOBRE LA ACTIVIDAD EMPRESARIAL

¿En qué sector de actividad de los siguientes se sitúa su empresa?

- Industrias extractivas
- Industria manufacturera
- Producción y distribución de energía eléctrica, gas y agua
- Construcción
- Comercio
- Reparación de vehículos de motor
- Hostelería
- Transporte y/o almacenamiento
- Comunicaciones
- Intermediación financiera
- Actividades inmobiliarias y de alquiler
- Servicios empresariales
- Administración pública, defensa y Seguridad Social obligatoria
- Educación
- Ingeniería o servicios técnicos
- Actividades sanitarias y veterinarias, servicio social
- Otras actividades sociales y de servicios prestados a la comunidad
- Otros (especificar:)
- NS / NC

¿Cuántos empleados posee su empresa?

- De 0 a 2
- De 3 a 9
- De 10 a 19
- De 20 a 49
- De 50 a 99
- De 100 a 249
- De 250 y más

- NS / NC
- ¿Cuántas oficinas o locales tiene la empresa en total?**
 - Ninguna (la actividad no lo requiere)
 - Trabajo en domicilio particular
 - Una o más (especificar número: _____)
 - NS / NC
- En caso de tener más de un empleado y al menos una oficina o local, ¿Cuántos de estos trabajadores permanecen más de la mitad de su jornada fuera de la oficina o local (p.ej. comerciales)?**

Especificar número: _____

II-BLOQUE SOBRE PROCESOS DE NEGOCIO

- Aproximadamente, ¿cuántos clientes tiene su empresa?**
 - Menos de cinco
 - Entre cinco y veinte
 - Entre veintiuno y cincuenta
 - Entre cincuenta y uno y cien
 - Entre cien y doscientos cincuenta
 - Más de doscientos cincuenta
 - NS / NC
- Aproximadamente, ¿con qué frecuencia se comunica su empresa con cada cliente?**
 - A diario
 - Semanalmente
 - Mensualmente
 - Alguna vez al año
 - NS / NC
- De las vías de comunicación utilizadas con sus clientes, ¿cuál es la más utilizada?**
 - Comunicación directa/ en persona
 - Por teléfono
 - Por fax
 - Por correo postal
 - Por correo electrónico
 - Por extranet
 - A través de una aplicación informática
 - NS / NC
- Aproximadamente, ¿cuántos proveedores tiene su empresa?**
 - Menos de cinco
 - Entre cinco y veinte
 - Entre veintiuno y cincuenta
 - Entre cincuenta y uno y cien
 - Entre cien y doscientos cincuenta
 - Más de doscientos cincuenta
 - NS / NC
- Aproximadamente, ¿con qué frecuencia se comunica su empresa con cada proveedor?**
 - A diario

- Semanalmente
 - Mensualmente
 - Alguna vez al año
 - NS / NC
- De las vías de comunicación utilizadas con sus proveedores, ¿cuál es la más utilizada?**
- Comunicación directa/ en persona
 - Por teléfono
 - Por fax
 - Por correo postal
 - Por correo electrónico
 - Por extranet
 - A través de una aplicación informática
 - NS / NC
- Si la empresa tiene local y al menos un empleado en movilidad, ¿con qué frecuencia necesitan contactar con la oficina o local de trabajo los empleados cuando están fuera?**
- Varias veces al día
 - Al menos una vez al día
 - Alguna vez por semana o menos
 - Nunca
 - NS / NC
- De las vías de comunicación utilizadas con los empleados, ¿cuál es la más utilizada?**
- Comunicación directa/ en persona
 - Por teléfono
 - Por fax
 - Por correo postal
 - Por correo electrónico
 - Por extranet
 - A través de una aplicación informática
 - NS / NC
- ¿Está satisfecho con este método de comunicación?**
- Sí
 - No
 - NS / NC
- En caso de tener varios locales, ¿necesita transmitir información entre ellas?**
- Sí
 - No
 - NS / NC
- ¿Con qué frecuencia necesitan transmitir información?**
- Varias veces al día
 - Al menos una vez al día
 - Alguna vez por semana o menos
 - Nunca
 - NS / NC
- En caso de utilizar Internet para conectar sus procesos con otras oficinas locales de la empresa, ¿qué procesos tiene su empresa conectados por Internet con otras oficinas o locales?**
- Control de inventario

- Control de producción y fabricación
 - Facturación y ventas
 - Transmisión de informes
 - Otro tipo de Integración de procesos (especificar: _____)
 - NS / NC
- En caso de utilizar Internet para conectar sus procesos con clientes o proveedores, de los siguientes procesos, ¿cuáles de ellos tiene su empresa conectador por Internet con clientes o proveedores?**
- Realización y control de pedidos
 - Facturación y ventas
 - Servicio post-venta (dudas, reclamaciones, etc.)
 - Transmisión de informes
 - NS / NC

III - BLOQUE SOBRE EL EQUIPAMIENTO TECNOLÓGICO DE LA EMPRESA Y SU USO

- En caso de tener su empresa más de un ordenador, ¿están sus ordenadores conectados entre sí por una red de área local?**
- Sí
 - No
 - NS / NC
- En caso de tener al menos un ordenador, ¿Principalmente, para qué utilizan habitualmente los ordenadores?**
- Ofimática básica (procesador de texto, hoja de cálculo o base de datos)
 - Facturación
 - Marketing y ventas
 - Nóminas y Recursos Humanos
 - Contabilidad o tesorería
 - Gestión de inventario / almacenes
 - Planificación de producción
 - Otros programas propios de mi sector de actividad
 - Acceder a Internet (buscar Información, e-mall, etc.)
 - Otros (especificar: _____)
 - NS / NC
- ¿Dispone su empresa de PDAs?**
- Sí
 - No
 - NS / NC
- En caso afirmativo, ¿para qué utilizan habitualmente la PDA?**
- Ofimática básica (procesador de texto, hoja de cálculo o base de datos)
 - Facturación
 - Marketing y ventas
 - Nóminas y Recursos Humanos
 - Contabilidad o tesorería
 - Gestión de inventario / almacenes
 - Planificación de producción
 - Otros programas propios de mi sector de actividad

- Acceder a Internet (buscar Información, e-mail, etc.)
 - Otros (especificar: _____)
 - NS/NC
- ¿Disponen de acceso a Internet en su empresa?**
- Sí
 - No
 - NS / NC
- En caso afirmativo, ¿qué tecnología de acceso a Internet utilizan?**
- A través de la línea telefónica (banda estrecha, incluye RDSI)
 - Banda ancha por ADSL
 - Cable
 - Otro tipo de acceso de banda ancha (p.ej. circuito dedicado, LMDS, etc.)
 - Otros (especificar: _____)
 - NS / NC
- Si disponen de acceso, ¿para qué utiliza Internet?**
- Conectar alguno de los procesos con otras oficinas / locales de la empresa
 - Conectar alguno de los procesos con clientes o proveedores
 - Realizar consultas o transacciones bancarias
 - Realizar gestiones con la Administración Pública
 - Formación y aprendizaje
 - Otro uso personal (acceder a juegos, chat, etc.)
 - Para realizar alguna actividad específica de su sector
 - Otro tipo de Integración de procesos (especificar: _____)
 - NS / NC
- ¿Disponen su empresa de Telefonía móvil?**
- Sí
 - No
 - NS / NC
- En caso afirmativo, ¿para qué utiliza la telefonía móvil?**
- Comunicaciones de voz/SMS
 - Comunicaciones de datos
 - Comunicaciones de Voz y Datos
 - Comunicaciones de Datos propios de mi sector
- ¿Disponen su empresa de red WLAN (Wi-Fi)?**
- Sí
 - No
 - NS / NC
- En caso afirmativo, ¿para qué utiliza la red WLAN?**
- Comunicaciones de voz/SMS
 - Comunicaciones de datos
 - Comunicaciones de voz y datos
 - Comunicaciones de Datos propios de mi sector
- ¿Qué factores considera más importante a la hora de adoptar una solución de movilidad?**
- Precio.
 - Conseguir una mayor productividad de los empleados.

- Reducción del tiempo de respuesta.
 - Reducción de costes.
 - Seguridad.
 - Su utilidad.
 - Mayor satisfacción para el cliente.
 - Otros (especificar: _____)
 - NS / NC
- ¿En qué actividades de su negocio aplicaría una solución de movilidad?**
- Tareas administrativas
 - Facturación
 - Marketing y ventas
 - Gestión de la fuerza de trabajo móvil
 - Transporte y Logística
 - Nóminas y Recursos Humanos
 - Contabilidad o tesorería
 - Gestión de inventario / almacenes
 - Planificación de producción
 - Otros procesos propios de mi sector de actividad
 - Acceder a Internet (buscar información, e-mail, etc.)
 - Otros (especificar: _____)
 - NS / NC
- Para los empleados de su empresa que se encuentran desplazados, ¿qué tipo de funcionalidades necesitaría a su alcance desde un dispositivo móvil?**
- Calendario
 - Editor de textos
 - Comunicaciones de voz/ Mensajería SMS ó MMS
 - Gestor de contactos
 - Sincronización de datos con los servidores de la empresa
 - Planificador de proyectos
 - Acceso a la Intranet de la empresa
 - Envío y recepción de e-mails
 - Integración con servicios informáticos de la empresa
 - Acceder a Internet
 - Otro (especificar: _____)
 - NS / NC
- ¿Estaría dispuesto a contratarlo?**
- Sí
 - No
 - NS / NC