

Universidad de Huelva

Departamento de Ingeniería Electrónica, de Sistemas
Informáticos y Automática



Un modelo general de referencia para el acceso remoto a laboratorios docentes y de investigación

Memoria para optar al grado de doctor
presentada por:

Marco Antonio Márquez Sánchez

Fecha de lectura: 23 de octubre de 2015

Bajo la dirección de los doctores:

José Manuel Andújar Márquez

Andrés Mejías Borrero

Huelva, 2015





**Universidad
de Huelva**

Departamento de Ingeniería Electrónica, de Sistemas Informáticos y Automática

Escuela Técnica Superior de Ingeniería

Un Modelo General de Referencia para el Acceso Remoto a Laboratorios Docentes y de Investigación

Tesis Doctoral de

Marco A. Márquez Sánchez

Dirigida por

Prof. Dr. José Manuel Andújar Márquez

y

Prof. Dr. Andrés Mejías Borrero

UNIVERSIDAD DE HUELVA

4 de marzo de 2015

Prof. Dr. José Manuel Andújar Márquez, Catedrático de Universidad de la Escuela Técnica Superior de Ingeniería de la Universidad de Huelva, y Prof. Dr. Andrés Mejías Borrero, Profesor Titular de Universidad de la Escuela Técnica Superior de Ingeniería de la Universidad de Huelva,

CERTIFICAN:

Que D. Marco A. Márquez Sánchez, Ingeniero Industrial e Ingeniero Técnico Industrial (Electrónica, Regulación y Automatismos), ha realizado bajo nuestra dirección y dentro del programa de doctorado Ingeniería de Control, Sistemas Electrónicos e Informática Industrial, el trabajo de investigación correspondiente a su Tesis Doctoral titulado:

Un Modelo General de Referencia para el Acceso Remoto a Laboratorios Docentes y de Investigación

Revisado el presente trabajo, estimamos que puede ser presentado al Tribunal que ha de juzgarlo.

Y para que así conste a efectos de lo establecido en el artículo 8º del Real Decreto 778/1998 y por la normativa Reguladora del título de Doctor de la Universidad de Huelva, autorizamos la presentación de este trabajo en la Universidad de Huelva.

Huelva, a 4 de marzo de 2015

Directores: Prof. Dr. José M. Andújar Márquez

Prof. Dr. Andrés Mejías Borrero

Doctorando: Marco A. Márquez Sánchez

A mi mujer, Pepa,

a mis hijos, Ale y Elena

y a la memoria de mis padres Antonio y Benita.

Agradecimientos...

No sería justo pensar que esta Tesis es sólo el esfuerzo y el trabajo de su autor. Este trabajo ha llegado a su fin gracias a la inestimable ayuda de muchos compañeros y amigos que siempre han estado ahí cuando los he necesitado.

En primer lugar, quiero agradecerle de forma muy especial a mi director de tesis, José Manuel Andújar, su apoyo en todos los momentos difíciles. Su visión científica de los problemas ha sido para mí la inspiración de muchas de las soluciones propuestas en esta Tesis. También quiero resaltar la inestimable ayuda de mi codirector de Tesis, Andrés Mejías. Sin su paciencia, y sus infinitas horas de dedicación y revisión de documentos, yo no habría podido terminar esta Tesis. Y, finalmente, debo agradecerle a la profesora Reyes Sánchez las muchas horas de debate con las que he aprendido a buscar nuevos caminos para abordar la solución a los problemas que me han ido surgiendo. Convencerla a ella de mis soluciones, ha sido siempre para mí una garantía.

También debo agradecerle al profesor Sebastián Dormido el interés que me ha manifestado siempre por la evolución de esta Tesis, y el haberme permitido colaborar con algunos miembros de su grupo. Muy especialmente con el profesor Luis de la Torre: sus ideas y sus desarrollos han hecho posible muchos de los resultados de esta Tesis. Asimismo ha sido muy importante para mí la ayuda prestada por el profesor Francisco Esquembre, quien me ha facilitado todos los recursos de EJS necesarios. Además quiero agradecerle al profesor Enrique Montero las facilidades y el apoyo que siempre me ha dado para la aplicación de esta Tesis a los sistemas de comunicaciones del CIT de la Universidad de Cádiz

Además, quiero agradecerles a mi compañera y directora de Instituto, Eva Fernández, y a mi compañera y subdirectora, Claudia Romero, que siempre me hayan apoyado y me hayan facilitado en todo lo posible la conciliación de mi trabajo con ellas y el desarrollo de esta Tesis.

Finalmente quiero agradecerle a mi mujer la paciencia infinita que ha tenido siempre para compartirme con mis proyectos, mis inquietudes profesionales y mi interés por superarme y aprender algo nuevo cada día.

Índice de contenido

1. Planteamiento General de la Tesis.....	1
1.1. <i>Resumen de la Tesis.....</i>	7
1.2. <i>Innovaciones que aporta la Tesis.....</i>	10
1.3. <i>Rendimiento científico de la Tesis</i>	11
2. Laboratorios Remotos: Evolución Histórica y Análisis de Características.....	15
2.1. <i>Evolución y Análisis de las Clases Prácticas de Laboratorio Remoto (CPLR) según un Enfoque Generacional.....</i>	17
2.1.1. <i>Primera Generación de Laboratorios Remotos.....</i>	18
2.1.1.1. <i>Proyectos Representativos de la Primera Generación.....</i>	18
2.1.1.2. <i>Análisis de la Primera Generación.....</i>	25
2.1.2. <i>Segunda Generación de Laboratorios Remotos.....</i>	28
2.1.2.1. <i>Proyectos Representativos de la Segunda Generación.....</i>	29
2.1.2.2. <i>Análisis de la Segunda Generación.....</i>	33
2.1.3. <i>Tercera Generación de Laboratorios Remotos.....</i>	34
2.1.3.1. <i>Proyectos Representativos de la Tercera Generación.....</i>	35
2.1.3.2. <i>Análisis de la Tercera Generación.....</i>	37
2.2. <i>Evolución y Análisis de las Clases Prácticas de Laboratorio Remoto (CPLR) según el Programa de Control del Experimento y el Subsistema de Interacción.....</i>	39
2.3. <i>Evolución y Análisis de las Clases Prácticas de Laboratorio Remoto (CPLR) según la Tecnología de Implementación del Modelo de Comunicaciones.....</i>	44
2.4. <i>Conclusiones del Análisis de las Distintas Generaciones de Experiencias Remotas de Laboratorio.....</i>	51
3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto.....	55
3.1. <i>Introducción.....</i>	58
3.2. <i>Clase práctica de laboratorio remoto.....</i>	60
3.3. <i>Enfoque de la Ingeniería de Sistemas.....</i>	66
3.4. <i>Proceso de enseñanza-aprendizaje de una clase práctica de laboratorio</i>	70
3.5. <i>Proceso de enseñanza-aprendizaje de una clase práctica de laboratorio remoto.....</i>	78
3.6. <i>Modelo conceptual general de una clase práctica de laboratorio remoto.....</i>	85
3.7. <i>Conclusiones.....</i>	90
4. Modelo General Aplicado de un Laboratorio Remoto.....	93
4.1. <i>Introducción.....</i>	95
4.2. <i>Consideraciones generales sobre la red del sistema de comunicaciones.....</i>	100
4.2.1. <i>Conceptos generales de la arquitectura de una red de paquetes.....</i>	101
4.2.2. <i>Protocolo de comunicaciones.....</i>	105
4.2.3. <i>Arquitectura cliente-servidor.....</i>	108

4.3. Desarrollo experimental a través de una red de paquetes.....	110
4.4. Estructura genérica de un sistema experimental convergente.....	113
4.5. Estructura física y funcional del modelo tecnológico genérico.....	115
4.5.1. Estructura funcional del servidor SARLAB para accesos múltiples.....	118
4.5.2. CONEXLAB.....	120
4.5.3. DIGEXLAB.....	126
4.5.4. Subsistema de interacción.....	133
4.5.5. Subsistema de análisis, diseño y programación.....	134
4.6. Percepción experimental.....	135
4.7. Subsistema de alimentación eléctrica.....	137
4.8. Control topológico.....	138
4.9. Movilidad experimental.....	139
4.10. Conclusiones.....	139
5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible.....	143
5.1. Introducción.....	144
5.2. Implementación de la estructura física de comunicaciones.....	145
5.2.1. Elección de la red pública de datos.....	146
5.2.1.1. Definición de una conexión TCP/IP.....	147
5.2.1.2. Políticas de seguridad en Internet.....	149
5.2.1.2.1. Restricciones en el acceso del usuario a Internet.....	151
5.2.1.2.2. Restricciones en las redes de tránsito.....	151
5.2.1.2.3. Restricciones de acceso en universidades y centros de investigación.....	155
5.2.1.3. Integración de SARLAB en la red de una institución científica.....	157
5.2.1.3.1. Red de campus de una institución.....	157
5.2.2. Características a cumplir por los servidores en la red de comunicaciones elegida. .	161
5.3. Implementación del sistema de convergencia genérico.....	162
5.3.1. Implementación práctica de la capa de adaptación de niveles.....	168
5.3.2. Implementación monolítica del resto de capas para sistemas sencillos.....	170
5.3.3. Implementación modular del resto de capas para sistemas complejos.....	171
5.4. Implementación del subsistema de alimentación.....	176
5.5. Implementación del subsistema de transmisión multimedia.....	180
5.6. Conclusiones.....	183
6. Implementación Funcional del Modelo basado en la Tecnología Disponible.....	185
6.1. Estrategia de desarrollo de las funcionalidades del modelo tecnológico general.....	186
6.1.1. Elección del lenguaje de programación.....	187
6.2. Elección del LMS.....	188
6.2.1. Integración de SARLAB en el LMS.....	191
6.2.1.1. Integración de SARLAB en la tecnología web.....	192
6.2.1.2. Integración completa de SARLAB en Moodle.....	196
6.3. Esquema genérico del lazo de desarrollo experimental controlado por SARLAB.....	201
6.4. Unificación de las comunicaciones.....	203
6.5. Herramienta para el desarrollo rápido de la interfaz de usuario.....	205
6.6. Desarrollo de CONEXLAB.....	212
6.7. Desarrollo de DIGEXLAB.....	219

6.7.1. Funcionalidad de gestión de DIGEXLAB.....	219
6.7.1.1. Configuración inicial de CONEXLAB.....	220
6.7.1.2. Validación de acceso.....	222
6.7.2. Funcionalidad de diseño de DIGEXLAB.....	226
6.7.3. Funcionalidades del sistema de convergencia genérico.....	231
6.7.3.1. Uso de EJS como herramienta de desarrollo rápido de las funcionalidades del sistema de convergencia genérico.....	233
6.7.4. Consideraciones sobre la eficiencia de Internet.....	235
6.8. <i>Control topológico de un sistema experimental</i>	237
6.9. <i>Conclusiones</i>	240
7. Laboratorios Remotos diseñados según el Modelo Propuesto.....	243
7.1. <i>Procedimiento de diseño</i>	244
7.2. <i>Experiencias en explotación usando el modelo</i>	246
7.2.1. UNILabs.....	247
7.2.1.1. Control de un cuatrirrotor de 3 grados de libertad.....	247
7.2.1.2. Control de robots móviles (sistema multi-agentes).....	248
7.2.1.3. Control de posición y supresión de vibraciones de un brazo flexible.....	249
7.2.2. Prácticas ofertadas por otras Universidades y Centros Tecnológicos.....	250
7.2.2.1. Efecto fotoeléctrico.....	251
7.2.2.2. Difracción de la luz.....	251
7.2.2.3. La luz en medios isótropos.....	252
7.2.2.4. Configuración del bucle de abonado para integración de voz y datos.....	253
7.2.3. Práctica ofertadas por la Universidad de Huelva.....	253
7.2.3.1. Automatización con PLCs.....	254
7.2.3.2. Diseño VHDL.....	255
7.2.3.3. Curvas características de placas fotovoltaicas.....	256
7.2.3.4. Modelado de una máquina síncrona.....	257
7.3. <i>Conclusiones</i>	258
8. Conclusiones y Desarrollos Futuros.....	261
8.1. <i>Conclusiones</i>	261
8.2. <i>Desarrollos Futuros</i>	265
Referencias.....	267

Índice de tablas

Tabla 2.1: Requerimientos necesarios a nivel de plugin para la realización de experiencias en los distintos laboratorios ofertados desde el MIT en su proyecto i-Lab.....	42
Tabla 5.1: Servidores SARLAB y LMS en las universidades en las que está en explotación el modelo.....	162

Índice de figuras

Figura 2.1: Interfaz de usuario del proyecto Mercury.....	19
Figura 2.2: Espacio de trabajo (Proyecto Mercury).....	19
Figura 2.3: Esquema del robot (Proyecto Mercury).....	19
Figura 2.4: Proyecto Telerobot.....	20
Figura 2.5: Interfaz de usuario del experimento presentado en [Bohus et al. 1996].....	22
Figura 2.6: Sistema experimental de la experiencia propuesta en [Bohus et al. 1996].....	22
Figura 2.7: Estructura general del experimento de laboratorio propuesto en [Bohus et al. 1996].....	23
Figura 2.8: Sistema experimental de control de niveles en cuatro tanques.....	24
Figura 2.9: Interfaz de usuario desarrollado en Labview del sistema experimental de la figura 2.8.....	24
Figura 2.10: Arquitectura funcional de bloques de un experimento de laboratorio remoto genérico.....	25
Figura 2.11: Detalles de la arquitectura funcional de bloques de un experimento remoto genérico.	27
Figura 2.12: Arquitectura básica i-Lab.....	30
Figura 2.13: Interfaz web de i-Lab.....	31
Figura 2.14: Área de asistencia de la experiencia en i-Lab.....	31
Figura 2.15: Portal del proyecto Telelab (selección de experimentos).....	32
Figura 2.16: Fotograma de un video de demostración en Telelab.....	33
Figura 2.17: Interfaz de usuario del proyecto Automat@Labs.....	36
Figura 2.18: Esquema general de la segunda y tercera generación de sistemas.....	39
Figura 2.19: Flujos de datos genéricos entre el alumno y la planta experimental.....	40
Figura 2.20: Evolución de proyecto WebLab de la Universidad de Deusto.....	52
Figura 3.1: Esquema relacional que representa el proceso enseñanza-aprendizaje en una CPL clásica (presencial). Profesor, alumnos y experimento comparten tiempo y espacio, estableciéndose contacto físico entre ellos.....	59
Figura 3.2: Esquema relacional que representa el proceso enseñanza-aprendizaje de una CPL realizada de forma remota. El profesor y los alumnos están desubicados unos de otros en el espacio, y puede que también en el tiempo. Análogamente el alumno y el experimento de	

laboratorio remoto están desubicados en el espacio.....	61
Figura 3.3: Multiplexación experimental.....	65
Figura 3.4: Diseño y configuración experimental remota.....	66
Figura 3.5: Esquema general de las fases y sistemas intervinientes en una CPL presencial.....	73
Figura 3.6: Modelo relacional del diseño, gestión y desarrollo de una CPL presencial conducida por el profesor y ejecutada por el alumno.....	75
Figura 3.7: Modelo relacional del diseño, gestión y desarrollo de una CPL presencial conducida y ejecutada por el alumno.....	77
Figura 3.8: Modelo relacional general del desarrollo de una CPL presencial.....	78
Figura 3.9: Modelo relacional del diseño, gestión y desarrollo de una CPLR guiada por el profesor y experimentada por el alumno.....	79
Figura 3.10: Modelo relacional del diseño, gestión y desarrollo de una CPLR guiada y experimentada por el alumno.....	79
Figura 3.11: Sistemas participantes en la gestión y desarrollo de una CPLR con convergencia del sistema de comunicaciones.....	82
Figura 3.12: Esquema del modelo conceptual general de una CPLR.....	86
Figura 4.1: Elementos del sistema experimental que permiten el intercambio de señales, a través del sistema de comunicaciones, desde y hacia el sistema enseñanza-aprendizaje que aloja al usuario (ver Figura 3.12).....	98
Figura 4.2: Estructura general de una red de comunicaciones.....	102
Figura 4.3: Ejemplo de transmisión de un paquete de datos por la red.....	104
Figura 4.4: Arquitectura de protocolo genérica.....	108
Figura 4.5: Arquitectura cliente-servidor aplicada al modelo de la Figura 3.12.....	110
Figura 4.6: Lazo de desarrollo experimental remoto a través de una red de paquetes, que incluye el servidor de vídeo y audio.....	111
Figura 4.7: Sistema experimental convergente con varios procesadores/computadores con SADS convergentes.....	113
Figura 4.8: Sistema experimental convergente con procesadores/computadores con SADS convergentes e instrumentos digitales programables.....	115
Figura 4.9: Representación externa del sistema experimental convergente más general.....	115
Figura 4.10: Estructuras funcional y física del modelo tecnológico general.....	117
Figura 4.11: Estructura de un laboratorio de acceso remoto.....	117
Figura 4.12: Estructuras funcional y física del servidor SARLAB.....	119
Figura 4.13: Túnel de comunicaciones a través de una red de paquetes.....	122
Figura 4.14: Estructura funcional de CONEXLAB.....	124
Figura 4.15: Diagrama de interacción de CONEXLAB.....	125
Figura 4.16: Casos de usos de DIGEXLAB.....	126
Figura 4.17: Estructura funcional de DIGEXLAB.....	128

Figura 4.18: Gestión del acceso a un sistema experimental controlado por SARLAB.....	129
Figura 4.19: Diagrama SDL del protocolo genérico de validación de acceso a una experiencia desde DIGEXLAB.....	132
Figura 4.20: Diagrama de interacciones completo de realización de una experiencia remota...	133
Figura 4.21: Modelo tecnológico general de sarlab.....	138
Figura 5.1: Topología internacional del proyecto GÉANT.....	152
Figura 5.2: Topología de red europea.....	153
Figura 5.3: Topología de Rediris.....	154
Figura 5.4: Red RICA.....	155
Figura 5.5: Estructura de seguridad de una institución científica.....	156
Figura 5.6: Estructura de la red de campus.....	159
Figura 5.7: Estructura de una VLAN de SARLAB en una institución genérica.....	160
Figura 5.8: Planta experimental didáctica sensorizada y acondicionada.....	164
Figura 5.9: Estructura funcional del sistema de convergencia genérico.....	166
Figura 5.10: Placa de adaptación de niveles digitales bidireccional 3.3 a 5 V.....	167
Figura 5.11: Mapeo de rangos de sensores a 0/3.3 0 0/5V.....	168
Figura 5.12: Circuito de mapeo a 0/3.3 V o 0/5 V.....	169
Figura 5.13: Circuito de mapeo a rangos 0/10 V -10/10 V o en general -V/+V.....	170
Figura 5.14: Placas microprocesadoras ix86 (32 y 64 bits).....	173
Figura 5.15: Placas basadas en procesadores ARM (PhidgetSBC3 y Raspberry Pi).....	173
Figura 5.16: Esquema de implementación de un sistema experimental convergente SARLAB.....	175
Figura 5.17: Esquema de bloques del subsistema de alimentación.....	176
Figura 5.18: Regleta de alimentación de la firma APC.....	177
Figura 5.19: Subsistema de alimentación con entradas/salida TTL.....	178
Figura 5.20: Módulos de implementación del subsistema de alimentación con Arduino.....	178
Figura 5.21: Módulos de relés amplificados y optoaislados.....	179
Figura 5.22: Diagrama de bloques de una cámara IP.....	180
Figura 5.23: Algunos servidores de vídeo comerciales.....	182
Figura 6.1: Código del applet en el LMS.....	193
Figura 6.2: Interfaz del gestor de reservas.....	194
Figura 6.3: Listado de reservas de un usuario.....	194
Figura 6.4: Página de administración del gestor de reservas.....	195
Figura 6.5: Página de configuración del EJSApp donde se incluyen los parámetros de SARLAB...	198
Figura 6.6: Inclusión de una nueva experiencia remota en Moodle controlada por SARLAB....	199

Figura 6.7: Configuración del servidor SARLAB y el identificador de experiencias.....	200
Figura 6.8: Esquema físico genérico del lazo de desarrollo experimental.....	202
Figura 6.9: Estructura de comunicaciones unificada mediante protocolo ModBus.....	204
Figura 6.10: Interfaz de EJS. A la izquierda se muestra la interfaz de diseño y a la derecha el resultado.....	207
Figura 6.11: Interfaz de desarrollo de procedimientos de EJS.....	208
Figura 6.12: Interfaz de gestión de los elementos en EJS.....	209
Figura 6.13: Elección de los elementos de establecimiento de las comunicaciones en EJS.....	212
Figura 6.14: Modelo tecnológico general que incluye la tecnología de túneles con protocolo SSL/SSH.....	214
Figura 6.15: Diagrama SDL simplificado del protocolo SSH.....	216
Figura 6.16: Diagrama SDL simplificado del proceso de establecimiento de sesión SSH.....	217
Figura 6.17: Diagrama SDL simplificado del gestor de sesiones de SARLAB.....	218
Figura 6.18: Detalle del fichero de experiencias.....	220
Figura 6.19: Fichero XML de configuración de una experiencia.....	221
Figura 6.20: Diagrama de interacción de la validación y el acceso a una experiencia.....	225
Figura 6.21: Estructura del fichero LMSList.....	226
Figura 6.22: Panel principal de la interfaz del cliente DIGEXLAB.....	229
Figura 6.23: Panel de diseño de conexiones de DIGEXLAB.....	230
Figura 6.24: Panel de conexiones de DIGEXLAB.....	230
Figura 6.25: Panel de asignación de salidas de alimentación.....	231
Figura 6.26: Esquema de uso de EJS como herramienta de desarrollo del sistema de convergencia genérico.....	234
Figura 6.27: Esquema básico del control topológico.....	238
Figura 6.28: Representación del control topológico de un sistema experimental.....	240
Figura 7.1: Control de un cuatrirrotor de 3 grados de libertad.....	248
Figura 7.2: Control de robots móviles.....	249
Figura 7.3: Control de posición y supresión de vibraciones de un brazo flexible.....	250
Figura 7.4: Fotografía general del montaje experimental real utilizado en el laboratorio remoto(izquierda). Detalle del disco giratorio motorizado para el intercambio de filtro interferencial a la entrada de la célula fotoeléctrica (derecha).....	251
Figura 7.5: Sistema experimental para estudiar la difracción de la luz.....	252
Figura 7.6: Sistema experimental que estudia la luz en medios isótropos.....	252
Figura 7.7: Sistema experimental de la práctica configuración del bucle de abonado para integración de voz y datos.....	253
Figura 7.8: Sistema experimental de la práctica automatización con PLCs.....	255
Figura 7.9: Sistema experimental de la práctica de diseño VHDL.....	256

Figura 7.10: Sistema experimental de la práctica de placas fotovoltaicas. El alumno percibe los cambios de irradiancia que aplica de forma remota a las placas.....257

Figura 7.11: Sistema experimental de la práctica de modelado de una máquina síncrona.....258

Lista de Acrónimos

<i>Acrónimo</i>	<i>Significado</i>
CPL	Clase práctica de laboratorio.
CPLR	Clase práctica de laboratorio remoto o clase práctica de laboratorio realizada (y puede que también configurada) de forma remota. Clase práctica realizada sobre un laboratorio físico (que puede contener elementos virtuales y/o de realidad aumentada) de forma remota; esto es, a través de la red (ver tabla 3.2) con desubicación física de usuarios (profesor/es y alumno/s) y experimento/s.
ECTS	<i>European Credit Transfer System</i> . Es la unidad de medida común de la duración de los estudios universitarios dentro del EEES. Su objeto no es medir la duración temporal de las clases impartidas por el profesor en cada asignatura, sino medir el tiempo invertido por el alumno para adquirir las competencias del programa de cada asignatura. En este sentido, cada crédito ECTS no solo incluye la asistencia del alumno al aula, sino también su dedicación al estudio o trabajo personal, la realización de seminarios, resolución de ejercicios, actividades académicamente guiadas, tutorías especializadas y colectivas, tareas grupales, etc.
EEES	El Espacio Europeo de Educación Superior (EEES) es un ámbito de organización educativo iniciado en 1999 con la Declaración de Bolonia para armonizar los distintos sistemas educativos de la Unión Europea y proporcionar una forma eficaz de intercambio entre todos los estudiantes. Se integran actualmente en el EEES, aparte de los 27 países de la UE, otros como Rusia o Turquía hasta llegar a la cifra total de 46 países participantes. En la actualidad, todas las enseñanzas universitarias en España están adaptadas al EEES y siguen un proceso homogéneo de diseño e implementación.

Acrónimo	Significado
LMS	Learning Management System. Sistema de gestión del aprendizaje. Herramienta TIC que permite la gestión y presentación de materiales educativos a estudiantes. El objetivo de esta herramienta es permitir el aprendizaje en cualquier parte y en cualquier momento. Es una herramienta Web, es decir, se usa a través de Internet utilizando un navegador Web.
LR	Laboratorio remoto. Laboratorio físico o real con capacidad de poder ser tele-operado a distancia, a través de la red, de modo que se pueda realizar experimentación sobre él con prestaciones análogas a como se haría de forma presencial.
LV	Laboratorio virtual. Laboratorio configurado con elementos (sistemas, instrumentos,...) simulados contenidos en uno o más ordenadores, conectados o no entre sí, con capacidades de aprendizaje y/o gestión de contenido.
RA	Realidad aumentada. Combina contenidos reales (usualmente obtenidos a través de un dispositivo captador de imágenes, tal como una cámara por ejemplo) con contenidos virtuales generados por computador. Esta combinación genera una realidad <i>enriquecida</i> donde el mundo real y virtual están conectados de forma dinámica e interactúan entre ellos.
SADS	Sistema de adquisición y distribución de señales. Sistema compacto o no que realiza las funciones necesarias desde la adaptación de la señal procedente de un sensor hasta la adaptación de la señal para activar un actuador.
SARLAB	Sistema de acceso remoto a laboratorio. Su función es la de resolver los problemas de acceso y gestión de comunicaciones entre redes público/privadas, de modo que el <i>lazo de desarrollo experimental</i> (ver Tabla 3.2) se pueda cerrar de forma remota y transparente para el usuario.
TIC	Tecnologías de la Información y las Comunicaciones. Aquellas que agrupan los elementos y las técnicas usadas en el tratamiento y transmisión de la información, principalmente la informática, electrónica y telecomunicaciones.
TIC/TAC	De las Tecnologías de la Información y la Comunicación a las Tecnologías para el Aprendizaje y el Conocimiento. Aquellas que no solo posibilitan el tratamiento y transmisión de la información, sino que también promueven aprendizaje y generan conocimiento. Para ello se requiere colaboración, indagación, interacción, innovación...

Lista de Conceptos

<i>Concepto</i>	<i>Significado</i>
ancho de banda	En general y referido a una red de datos, el ancho de banda es la máxima velocidad de transferencia de información, la cual suele estar dada en múltiplos de bit por segundo, siendo el bit la unidad mínima de información. Los datos que se transmiten por la red utilizan palabras (denominada byte) formadas por conjuntos de bits ordenados de forma adecuada. Está internacionalmente aceptado que 1 byte está formado por 8 bits, de modo que una palabra de 64 bits contiene 8 bytes.
base de conocimiento	Tipo especial de base de datos para la gestión del conocimiento por parte de personas o máquinas. En el contexto de esta investigación nos referimos al primer caso (artículos, guías, manuales de usuario, formularios, etc.). Hoy en día las bases de conocimiento tienen un fácil acceso informático (las hay públicas y privadas), fundamentalmente a través de Internet.
diseño y configuración experimental remota	Permite configurar de forma remota (sin estar en contacto físico con el recurso) las clases prácticas de laboratorio. Esto posibilita: (1) Adaptar a la modalidad de laboratorio remoto las clases prácticas de laboratorio clásica (presencial), y (2) Conectar recursos desubicados geográficamente (pertenecientes a laboratorios distintos) para configurar un experimento de laboratorio distribuido (ver definición en esta tabla).

<i>Concepto</i>	<i>Significado</i>
dispositivos de red	Conjunto de elementos físicos dotados del software y de los protocolos adecuados para conectar entre sí a los dispositivos de usuario final (computadores, impresoras, escáneres, elementos que conforman el laboratorio remoto y demás elementos que brindan servicios directamente al usuario), posibilitando interacción e intercomunicación entre ellos a través de la red.
estructura de datos	En el lenguaje de la informática, una estructura de datos es una forma de organizar un conjunto de datos elementales con el objetivo de facilitar su manipulación. Un dato elemental es la mínima información que se tiene en un sistema.
experimento de laboratorio	Conjunto de equipos y elementos discretos que adecuadamente conectados, programados y controlados permiten configurar y llevar a cabo una clase (o parte de ella) práctica de laboratorio.
experimento de laboratorio remoto	Experimento de laboratorio desarrollado en un LR.
experimento de laboratorio remoto distribuido	Experimento configurado mediante la conexión remota de recursos desubicados en el espacio (en diferentes LR por ejemplo) y que pueden ser gestionados de forma abstracta como un todo (experimento).
gestor de bases de datos	Es una aplicación informática que permite el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos.
gestor de contenidos	Es una aplicación informática usada para crear, editar, gestionar y publicar contenido digital multimedia en diversos formatos. Permite generar páginas web dinámicas interactuando con el servidor web para generar la página web bajo petición del usuario.

<i>Concepto</i>	<i>Significado</i>
guía docente	Documentación de una asignatura que incluye sus datos generales (denominación, número de créditos ECTS, curso, etc.) y específicos: Descripción de contenidos, situación de la asignatura en la titulación, objetivos, competencias a adquirir por los estudiantes, metodología, técnicas docentes, temario desarrollado, bibliografía, sistemas de evaluación y organización docente.
Ingeniería de Sistemas	Ingeniería interdisciplinar que permite estudiar y comprender la realidad con el propósito de implementar u optimizar sistemas complejos. Puede verse como la aplicación tecnológica del enfoque sistémico o estudio interdisciplinario que trata de encontrar las propiedades comunes a entidades llamadas sistemas, esto es, elementos materiales o conceptuales compuestos cuyos componentes se relacionan con al menos algún otro componente.
instrumento virtual	Sistema modelado con todas sus capacidades de procesado, sensores y controles contenidos en uno o más ordenadores, que permite acceso local a los recursos reales o simulados.
interfaz de red	Dispositivo físico, normalmente configurable por software, que permite la comunicación remota con otros equipos a través de redes privadas o públicas.
latencia	En una red de datos se denomina latencia a la suma de los retardos temporales en la misma. El retardo es producido por la demora en la propagación y transmisión de los paquetes de datos (ver Tabla 3.2) dentro de la red. Este retardo no es determinista sino que tiene carácter aleatorio y por tanto su valor es indeterminado, de difícil acotación y variable. La latencia es la característica que condiciona el uso de las redes en procesos de tiempo real.
lazo de desarrollo experimental	Relación de interacción entre el alumno y el experimento de laboratorio, y viceversa, donde el alumno recibe la respuesta del experimento a las acciones que realiza sobre él.

<i>Concepto</i>	<i>Significado</i>
lazo de desarrollo experimental remoto	Relación de interacción a través de la red entre el alumno y el experimento de LR (ver Tabla 3.1), y viceversa, donde el alumno recibe la respuesta del experimento a las acciones que realiza sobre él.
lazo de gestión experimental	Relación de interacción entre profesor y alumno, y viceversa, donde el alumno recibe la docencia de la clase práctica de laboratorio y el profesor recibe la respuesta del alumno a esa docencia. El profesor gestiona mediante este lazo el proceso enseñanza-aprendizaje.
lazo de gestión experimental remoto	Relación de interacción a través de la red entre el alumno y el/los gestores del aprendizaje, y viceversa, bien de forma directa o a través de un <i>sistema de gestión del aprendizaje remoto</i> (LMS).
modelo OSI	El modelo de interconexión de sistemas abiertos (ISO/IEC 7498-1), también llamado OSI (<i>open system interconnection</i>), fue creado por la Organización Internacional para la Estandarización (ISO) en el año 1984. Es un marco de referencia para la definición de arquitecturas en la interconexión de los sistemas de comunicaciones. Este modelo permite a los fabricantes crear redes que sean compatibles con otras redes, con lo cual, al compartir especificaciones, pueden compartir también información entre ellas.
multimedia	Tecnología que permite manejar y presentar información en formas combinadas y de modo interactivo con el usuario: texto, audio, imágenes fijas, animaciones, vídeo, etc.
multiplexación experimental	Capacidad de asignación remota de un usuario a un experimento (configurar un lazo de desarrollo experimental) entre múltiples usuarios disponibles.

<i>Concepto</i>	<i>Significado</i>
paquetes de datos	Se denomina paquete de datos a cada uno de los bloques en que se divide, en el nivel de red, la información a enviar a través de ésta. En todo sistema de comunicaciones resulta interesante dividir la información a enviar en bloques de un tamaño máximo conocido. Esto simplifica el control de la comunicación, las comprobaciones de errores, la gestión de los equipos de encaminamiento, etc. Un paquete de datos es una unidad fundamental de transporte de información y está generalmente compuesto de tres elementos: una cabecera que contiene usualmente la información necesaria para trasladar el paquete desde el emisor hasta el receptor, el área de datos que contiene los datos que se desean trasladar y la cola que comúnmente incluye código de detección de errores.
pasarela TIC	Concepto que permite unificar el lenguaje de comunicación entre usuarios y experimentos de laboratorios remotos de cualquier naturaleza, lo cual permite desvincular el lazo de desarrollo experimental de la naturaleza del experimento.
plan de estudio	Diseño curricular que se aplica a las enseñanzas impartidas por un Centro de Estudios. En España, los Planes de estudio de enseñanzas universitarias están obligados por Ley a recoger los apartados siguientes: justificación, objetivos, contenidos, recursos, sistema de garantía de calidad, admisión de estudiantes, planificación y resultados previos.
programa de control del experimento	Software que forma parte del Sistema Experimental Remoto y que es el encargado de controlar la planta experimental, a través de las interfaces de actuación y adquisición.
red	Red de computadores o simplemente red es un conjunto de computadores y otros dispositivos interconectados mediante canales de comunicación que permiten compartir recursos e información entre todos los elementos conectados.

<i>Concepto</i>	<i>Significado</i>
sistema de gestión del aprendizaje en red	Subsistema que permite la programación de alto o bajo nivel, generación de estímulos, control, etc., desde un ordenador u otro tipo de terminal informático hacia el <i>sistema experimental</i> , así como la adecuación y procesado de los datos recibidos desde él para su análisis y representación mediante gráficos, modelos, etc.
sistema experimental	Conjunto de elementos hardware y software que, situados normalmente en un laboratorio, permiten al alumno desarrollar un experimento de laboratorio de forma presencial.
sistema experimental remoto	Sistema experimental que se encuentra situado en un laboratorio y que permite al alumno desarrollar un experimento de laboratorio a distancia, desde cualquier ubicación externa al laboratorio con el acceso adecuado a una red de comunicaciones.
subsistema de alimentación eléctrica	Grupo de elementos software y hardware responsable de controlar la alimentación eléctrica de los distintos componentes del sistema experimental.
subsistema de análisis, diseño y programación experimental	Aplicación informática genérica o específica, proporcionada por el/los fabricante/s de los elementos activos que componen el <i>sistema experimental</i> , y que permite realizar sobre ellos las acciones necesarias para completar un experimento: programación, configuración, simulación, etc.
subsistema de convergencia del sistema de comunicaciones	Capa o nivel añadido al sistema de comunicaciones que permite implementar el concepto de <i>pasarela TIC</i> (ver definición en esta tabla).

<i>Concepto</i>	<i>Significado</i>
subsistema de interacción	Subsistema que permite la interacción del usuario con el sistema experimental. En el nivel menos exigente, esta interacción puede reducirse a una simple caja de diálogo o un formulario a rellenar desde la pantalla de un ordenador; sin embargo, el nivel de exigencia de la interacción puede ir en aumento: mediante un manipulador o una completa interfaz gráfica de usuario, que puede incluir la visión física y/o simulada de la evolución del proceso experimental; e incluso en un nivel superior, la sensación de contacto con el sistema experimental remoto puede estar aumentada, mediante el uso de técnicas de realidad aumentada.
subsistema de transmisión multimedia	Conjunto de dispositivos que permiten transmitir por la red el vídeo y el audio que permite seguir la evolución de la experiencia desde la interfaz de usuario.

1. Planteamiento General de la Tesis

Esta Tesis se enmarca dentro de una línea de investigación ya consolidada en el seno del Grupo de Investigación *Control y Robótica* (TEP 192), la cual, desde una perspectiva general de educación en ingeniería, pretende contribuir en los aspectos de diseño, configuración, desarrollo, accesibilidad y uso de los laboratorios docentes (y también de investigación) de forma remota.

El entorno educativo más cercano que soporta los objetivos de esta Tesis está fundamentalmente compuesto por los nuevos planes de estudio recientemente estrenados en España en el ámbito de las ingenierías. Estos planes de estudio están integrados en el *Espacio Europeo de Educación Superior* (*European Higher Education Area*, EHEA), una iniciativa del denominado *proceso de Bolonia* que tiene por objetivo crear un sistema de educación superior en Europa que sea comparable y compatible entre los países firmantes [European Comission, 2013].

Como puede comprobarse a partir de cualquiera de las directivas ministeriales de las distintas ingenierías, los objetivos de las titulaciones están traducidos a competencias que deben adquirir los estudiantes que cursen y superen los estudios. Esto implica que el enfoque didáctico en este nuevo marco de referencia formativo ha de estar necesariamente muy unido a la realidad técnica que le va a tocar vivir al futuro egresado. A partir de aquí es fácil deducir que se hace necesario un desarrollo curricular con una importante

contextualización tecnológica, que aproxime las actividades de aprendizaje de los alumnos a las actividades profesionales en las diferentes ingenierías.

Lo anterior hace más hincapié si cabe en una parte formativa que ha sido tradicionalmente muy importante en los estudios de ingeniería, la cual engloba las denominadas *clases prácticas de laboratorio* (CPL). La CPL es considerada tradicionalmente como una actividad generadora del proceso de enseñanza-aprendizaje a la que, de acuerdo con su concepción intrínseca, le corresponde el papel o función principal del desarrollo de las habilidades experimentales. En el desarrollo de una CPL se sigue la técnica didáctica experimental que tiene, como característica diferencial frente a otras, el poner en contacto físico al alumno con las experiencias experimentales, teniendo como máxima aspiración crearle la sensación de tener el fenómeno experimental entre sus manos, de modo que por un lado refuerce el conocimiento adquirido en clase, y por otro le permita eliminar dudas de conocimiento usando sus propios esquemas mentales mediante la interacción experimental. Los espacios donde se imparte este tipo de docencia se denominan usualmente *laboratorios* y han de estar dotados de los recursos necesarios para la realización de las experiencias prácticas que procedan.

La realización de las CPL constituye en sí misma un proceso de enseñanza/aprendizaje en el que se manifiestan todos los componentes no personales del proceso: problema, objeto, objetivos, forma, estrategias, métodos, contenido, medios y evaluación.

Dando pues por sentado la enorme importancia que tienen en los estudios de ingeniería (también en otros por supuesto) las CPL, es necesario contextualizarlas en el momento actual, donde los condicionantes sociales y económicos por un lado, y las posibilidades tecnológicas por otro, permiten explorar nuevos escenarios que eliminen rigideces temporales y físicas, para permitir unas pautas formativas más al servicio de

los usuarios (profesores y alumnos), y no al revés como hasta ahora. Entre otros, los inconvenientes a los que están sometidas las CPL actuales son los siguientes:

- Los horarios de uso de los laboratorios deben estar contenidos en los horarios docentes de cada Centro, normalmente coincidentes con el de su personal de administración y servicios (PAS) y educativo.
- El tiempo disponible para la realización de una práctica está organizado en un horario dividido en tramos, con lo cual no hay libertad de extensión y cambios sin afectar a otras materias y/o grupos de alumnos.
- Una problemática análoga se da desde la perspectiva del docente, con un encaje horario en su actividad presencial semanal que permite pocos cambios.
- Los recursos de laboratorio son compartidos en la franja horaria por diferentes grupos de alumnos, de modo que la disponibilidad de los recursos con una cierta libertad que permita adecuarse a las necesidades de aprendizaje propias de cada alumno es prácticamente una utopía.
- El tiempo disponible para la realización de una práctica suele ser siempre insuficiente, con lo cual hay que recurrir a soluciones de urgencia que suelen implicar la realización de trabajos previos y posteriores fuera del laboratorio (la mayoría propios de él), con objeto de completar la práctica en tiempo.
- El deterioro del material de laboratorio por una manipulación continua y/o incorrecta es muy acusado. Debido a esto se originan sobrecostes de mantenimiento y/o reposición.

Teniendo presentes estas consideraciones, se pueden plantear las siguientes aspiraciones:

- ¿Por qué no poder utilizar los laboratorios 24 horas al día, los 7 días de la semana?
- ¿Por qué el uso de un laboratorio docente ha de requerir la presencia coincidente en el tiempo y el espacio de un docente y/o un PAS?
- ¿Por qué los alumnos han de tener horarios tan rígidos que pueden condicionar su vida profesional y/o personal?
- ¿Por qué tener que desplazarse a un recinto universitario para realizar una práctica de laboratorio?
- ¿Por qué no poder compaginar horarios de trabajo y de estudios?
- ¿Por qué la interacción entre profesor y alumno en una CPL ha de realizarse en presencia de ambos?

La coherencia de las aspiraciones planteadas, en cuanto a sus posibilidades de llevarse a cabo, se asienta principalmente en el grado de madurez que han alcanzado hoy en día las TIC (Tecnologías de la Información y las Comunicaciones) a nivel de tratamiento de información, comunicaciones y multimedia. Esto ha permitido que en la enseñanza actual sean posibles escenarios de aprendizaje soportados por las TIC. Como una consecuencia directa del uso de estos nuevos escenarios, se están introduciendo nuevos aspectos didácticos como los de gestión y dirección del aprendizaje que no tenían tanta importancia en los modelos de enseñanza anteriores.

Estas nuevas posibilidades que ofrece la incorporación de las TIC como soporte de nuevas formas de enseñanza/aprendizaje, suponen un reto especial en las ingenierías. En efecto, en estas titulaciones las experiencias prácticas (fundamentalmente de laboratorio) constituyen en torno al 50% del contenido formativo total de los planes de estudio. Esto ha originado que en los últimos años

hayan sido presentadas diferentes propuestas en el ámbito de los laboratorios con soporte TIC [Andújar y Mateo, 2010]:

- *instrumento virtual*. Sistema modelado con todas sus capacidades de procesado, sensores y controles contenidos en uno o más ordenadores, que permite acceso local a los recursos reales y/o simulados.
- *Instrumento remoto*. Instrumento virtual o físico con capacidad de comunicación a través de la red, cuya función es permitir la comunicación a distancia de los recursos físicos y/o simulados.
- *Laboratorio virtual (LV)*. Instrumento/s simulado/s contenido/s en uno o más ordenadores, conectados o no entre sí, con capacidades de gestión y/o aprendizaje de contenido.
- *Laboratorio remoto (LR)*. Lugar o entorno cuya función es realizar un control sobre un sistema físico a distancia, con el objetivo de teleoperar un sistema real, realizar experimentos y acceder a los datos a través de la red para obtener medidas. En su significado más amplio, puede tener capacidades de gestión, aprendizaje de contenido y/o reservas de recursos compartidos.

Recientemente en [Andújar y Mateo, 2010] se ha incorporado a la lista anterior el concepto de *Laboratorio Remoto Aumentado* [Andújar, Mejías y Márquez, 2011], un laboratorio remoto en el que se potencian desde el lado del usuario los elementos y capacidades físicas mediante elementos y capacidades virtuales, con la particularidad de que los escenarios físico (real) y virtual están interconectados e interaccionan entre ellos mediante técnicas de realidad aumentada.

Independientemente del contenido tecnológico y académico presentes en propuestas prácticas publicadas en la literatura sobre los tipos de laboratorio anteriores, todas tienen una característica común: *especificidad* y *concreción*. Esto quiere decir que están orientadas a resolver dificultades concretas, y no a tratar de forma general la problemática y reto que supone la enseñanza/aprendizaje de las prácticas de laboratorio de forma remota sin entrar en la naturaleza de las mismas.

Basado en lo anterior, esta Tesis pretende aportar una generalización a la solución del problema de facilitar el proceso de enseñanza/aprendizaje, referido a una *clase práctica de laboratorio remoto* (CPLR). Para ello se proponen aportaciones en tres aspectos fuertemente relacionados unos con otros:

1. Definir un modelo conceptual teórico general, independiente de la tecnología y de la naturaleza de las prácticas, que permite desarrollar y solucionar el problema de manera formal.
2. Diseñar y desarrollar una propuesta tecnológica que pueda dar soporte al modelo conceptual.
3. Realizar una implementación práctica basada en las posibilidades tecnológicas actuales.

El punto de partida del diseño de la investigación que recoge esta Tesis es considerar el acceso a una CPLR como una nueva técnica didáctica, frente al enfoque clásico que considera esta posibilidad de acceso como un recurso más que facilita la realización de prácticas de laboratorio de forma remota. Esto es, desde el enfoque de este trabajo de investigación, el acceso remoto a las prácticas de laboratorio se considera una forma de relación entre los elementos del acto didáctico que supone la realización de una práctica de laboratorio de forma remota. Por este motivo la Tesis abarca diseños,

desarrollos e implementaciones en todos los niveles del acto didáctico, desde la gestión y dirección del aprendizaje hasta la configuración y manejo de experimentos.

Se es consciente de la dificultad que tiene abordar un problema tan complejo como éste desde un enfoque tan general, donde los experimentos, las comunicaciones y el proceso de enseñanza/aprendizaje interrelacionan entre ellos y con individuos (profesores y alumnos) dentro de un todo que es un sistema dinámico complejo. Parece pues que el enfoque debería ser que ese sistema evolucionara hacia un objetivo idóneo, para lo cual todos los elementos deberían trabajar juntos de forma sinérgica.

Del análisis anterior del problema al que se enfrenta este trabajo de investigación surge, casi de forma natural, la idea de abordarlo desde el punto de vista de la Ingeniería de Sistemas. Esto es, desde una perspectiva sistemática y científica que permita aproximar y representar la realidad del problema, así como capturar su fuerte carácter interdisciplinar, partiendo de la base que los elementos integrantes del proceso (experimentos, comunicaciones, enseñanza/aprendizaje e individuos) han de trabajar como un sistema holístico, donde lo importante no son las partes, sino las relaciones y los conjuntos que a partir de ellas emergen.

Con objeto de facilitar la lectura de esta Tesis, se han incluido antes de este primer capítulo una lista de acrónimos y una lista de conceptos (ambas tablas están ordenadas por orden alfabético). Estos términos serán definidos y estudiados a lo largo de la misma, los cuales constituyen los elementos fundamentales del modelo desarrollado. Los conceptos, cada vez que aparecen en el texto, están escritos en cursiva.

1.1. Resumen de la Tesis

Esta memoria está organizada en 6 capítulos de acuerdo a la distribución siguiente: en el capítulo 2, *Laboratorios Remotos: Evolución Histórica y Análisis de Características*, se

realiza un exhaustivo estudio sobre las soluciones propuestas a lo largo del tiempo al problema del acceso remoto a prácticas de laboratorio. Se analizan las herramientas y tecnologías usadas en cada una y se constata que ninguna de estas soluciones es general sino particular, vinculada a la naturaleza experimental del sistema, y con una fuerte especificidad. Por otro lado, estas experiencias se clasifican en el capítulo 2 dependiendo del aspecto del problema más relevante en cada una: el de enseñanza-aprendizaje, el de comunicaciones o el experimental. Así, según esta clasificación, en la primera generación el problema fundamental era el control remoto, en la segunda se empiezan a considerar los aspectos didáctico y de gestión, y en la tercera se actualiza el enfoque didáctico, y se busca facilitar el diseño. La clasificación de las soluciones permitirá obtener un esquema que englobe el funcionamiento de cualquiera de las estudiadas.

El capítulo 3, *Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto*, comienza estableciendo el esquema relacional de la CPL clásica o presencial. A partir de aquí se desarrolla el modelo necesario para llevar este esquema al laboratorio remoto. Así, se propone el esquema relacional que representa el proceso enseñanza-aprendizaje de una CPLR. Aquí se establecen las pautas para desvincular el esquema de la CPLR de la naturaleza del sistema experimental, lo cual permite desarrollar un modelo general. Para el desarrollo del mismo se emplea el enfoque de la Ingeniería de Sistemas. Seguidamente se aplica este enfoque a la CPL clásica o presencial y se extiende a la CPLR. A partir de aquí, se está ya en condiciones de afrontar y desarrollar el objetivo fundamental del capítulo, que es el desarrollo del modelo conceptual de una CPLR.

En el capítulo 4, *Modelo General Aplicado de un Laboratorio Remoto*, el modelo conceptual general desarrollado en el capítulo 3 evoluciona hacia una visión más práctica, hasta llevarlo a un modelo tecnológico genérico más próximo a la implementación final. Esto es, se pretende la evolución del modelo conceptual sin entrar en ninguna tecnología concreta TIC/TAC, que será en cada momento una de las

disponibles. Para ello se irán introduciendo los elementos especializados necesarios que han de permitir la implementación práctica del modelo.

El modelo tecnológico general constituye una idealización tecnológica en el que la red es una red de paquetes genérica (idealizada), y que plantea la estructura más genérica que puede adoptar el sistema experimental con acceso a la red.

En el capítulo 5, *Implementación Física del Modelo Propuesto basado en la Tecnología Disponible*, se presenta el diseño del esquema general de implementación física del modelo (elementos hardware del sistema y aspectos de integración en la infraestructura de red) y en el capítulo 6, *Implementación Funcional del Modelo Propuesto en la Tecnología Disponible*, se aborda el desarrollo de las funcionalidades y las aplicaciones informáticas desarrolladas que dan soporte a esas funcionalidades.

El diseño del esquema general de implementación física del modelo plantea la elección o implementación, en caso necesario, de todos los componentes físicos comprendidos en el modelo tecnológico general, lo cual se realiza en el capítulo 5. Por otro lado, los aspectos relacionados con la elección o el desarrollo de las aplicaciones a ejecutar por los componentes hardware para realizar las funcionalidades asociadas a ellos, se desarrollan en el capítulo 6.

En el capítulo 7, *Ejemplos de Laboratorios Remotos diseñados según el Modelo Propuesto*, se presentan varios casos reales de laboratorios remotos en explotación basados en la implementación del modelo propuesto en esta Tesis. Entre ellos hay sistemas ofertados por Universidades distintas de la de Huelva y otros ofertados por ésta última, en los que la implementación del modelo ha sido completa, desde la concepción de la idea hasta su puesta en explotación.

El capítulo 8, *Conclusiones y Desarrollos Futuros*, presenta las conclusiones más relevantes de esta Tesis y esboza las líneas de investigación que han sido abiertas por la misma.

Las referencias bibliográficas consultadas cierran la memoria de esta Tesis.

1.2. Innovaciones que aporta la Tesis

Esta Tesis aporta un modelo conceptual general para la definición y explotación de prácticas de laboratorio remoto. En contraposición a las soluciones aportadas hasta ahora en la Literatura Técnica, que atienden parcialmente a los distintos aspectos del problema, el modelo propuesto da solución a los tres aspectos fundamentales: el de enseñanza-aprendizaje, el de comunicaciones y el experimental. Asimismo, el modelo propuesto es aplicable a cualquier área de conocimiento. La complejidad del problema ha hecho necesario el enfoque sistémico para abordar como un todo cada una de las partes que lo componen.

El modelo puramente conceptual referido en el párrafo anterior no proporciona una versión implementable del problema. Se podría haber optado por desarrollar directamente la construcción a partir del estado tecnológico actual. Esto sin embargo limitaría el carácter general de la solución, por lo que se ha optado por obtener una visión constructiva, evolucionando el modelo conceptual general a un modelo tecnológico genérico, como segunda aportación de la Tesis, construido a partir de la idealización de los elementos hardware/software que lo conforman. De esta forma, este modelo se puede ir adaptando a las distintas evoluciones que se produzcan en cada uno de sus componentes desde el punto de vista tecnológico.

Otra aportación de esta Tesis es un conjunto de desarrollos hardware y software, plenamente operativos, que realizan la adaptación del modelo tecnológico genérico al

estado actual de la tecnología, y que está en explotación en numerosas Universidades. Este conjunto de elementos aporta una solución general al problema de acceso remoto, y comprende las siguientes aportaciones concretas:

- ✓ *SARLAB (Sistema de Acceso Remoto a Laboratorios)*. Es la implementación sobre la familia de protocolos TCP/IP, que es la utilizada en Internet y en la inmensa mayoría de las redes internas o intranets, del sistema encargado de gestionar las comunicaciones en la realización de una CPLR. Controla el flujo de datos entre el alumno (conectado a Internet) y la experiencia (conectada en la red local de la entidad que la oferta). SARLAB es responsable además de gestionar la concurrencia permitida por cada tipo de experiencia, dotándola además de la capacidad de acceso colaborativo.
- ✓ *Elementos de EJS (Easy Java Simulations) para conseguir la integración de la experiencia en el LMS*. El principal de ellos es el que corresponde a SARLAB, que además aporta la unificación de la validación de acceso entre SARLAB y el LMS. Otros elementos importantes son los que gestionan la unificación de las comunicaciones (máster y esclavo de ModBus).
- ✓ *Elementos de EJS para asistir en el diseño de una CPLR*. Entre ellos se encuentran los correspondientes a plataformas hardware que permiten el acceso a infinidad de sensores y actuadores, como por ejemplo Arduino y Phidgets.
- ✓ Una herramienta completa que asiste el diseño y construcción de la interfaz de usuario de las experiencias de laboratorio con acceso remoto.

1.3. Rendimiento científico de la Tesis

Esta sección está dedicada a enmarcar el entorno científico en el que se ha desarrollado la Tesis y a destacar los resultados de la investigación realizada; para ello

se detallarán las publicaciones con índice de impacto que han surgido de este trabajo, así como las presentaciones en congresos que muestran que la memoria que se presenta constituye la recopilación ordenada e integrada de una producción científica realizada dentro de la línea de investigación dedicada a los laboratorios remotos, en el grupo de investigación *Control y Robótica* (TEP 192).

Artículos en revistas internacionales

Andújar, J.M., Mejías, A. y Márquez, M. **Augmented Reality for the Improvement of Remote Laboratories: An Augmented Remote Laboratory.** *IEEE Transactions on Education.* Vol. 54(3), pp. 492-500, 2011. **Índices de calidad:** Esta revista está incluida en el JCR en la categoría *Education, Scientific Disciplines.* Índice de impacto: 1,165.

Mejías A., Andújar, J. M. y Márquez, M. **Digital Electronics Augmented Remote Laboratory: Dearlab.** *International Joournal of Engineering Education.* Vol. 30(4), pp. 950-963, 2013. **Índices de calidad:** Esta revista está incluida en el JCR en la categoría *Education, Scientific Disciplines.* Índice de impacto: 0,360.

Márquez, M; Sánchez, R.; Tirado, R.; Mejías, A.; Andújar, J. M. **Explore the educational opportunities of a remote laboratory for photovoltaic systems lessons.** *IEEE Transactions on Learning Technologies.* En revision. **Índices de calidad:** Esta revista está incluida en el JCR en la categoría *Computer Science, Interdisciplinary Applications.* Índice de impacto: 1,220.

Márquez, M; Sánchez, R.; Tirado, R.; Mejías, A.; Andújar, J. M. **A Remote Laboratory Devoted to Test Bench for Electric Machines.** *IEEE Transactions on Learning Technologies.* En revision. **Índices de calidad:** Esta revista está incluida en el JCR en la categoría *Computer Science, Interdisciplinary Applications.* Índice de impacto: 1,220.

Andújar, J. M.; Mejías, A.; Márquez, M; Sánchez, R.; Tirado, R. **General Reference Model for Remote Access Systems Laboratories**. *Journal of Engineering Education*. En revision. **Índices de calidad:** Esta revista está incluida en el JCR en la categoría Education, Scientific Disciplines. Índice de impacto: 2,717.

Presentaciones en Congresos

Mejías, A., Márquez, M. Esquembre, F., y Andújar, J.M. **Connecting hardware to Easy Java Simulations: from virtual experiments to remote/local labs**. XVIII International Conference on Multimedia in Physics Teaching and Learning (MPTL' 18). Madrid (España). Actas del Congreso, 11-13 de Septiembre de 2013.

Mejías, A., Esquembre, F., Márquez, M. y Andújar, J.M. **An Integrated Augmented Reality System for Easy Java Simulations**. XVIII International Conference on Multimedia in Physics Teaching and Learning (MPTL' 18). Madrid (España). Actas del Congreso, 11-13 de Septiembre de 2013.

Sánchez, R., Mejías, A., Márquez, M. y Andújar, J.M. **A Remote Laboratory for photovoltaic modules assays using Easy Java Simulations**. XVIII International Conference on Multimedia in Physics Teaching and Learning (MPTL' 18). Madrid (España). Actas del Congreso, 11-13 de Septiembre de 2013.

Mejías, A., Márquez, M., Andújar, J.M. Y Sánchez, R. **A Complete Solution for Developing Remote Labs**. The 10th IFAC Symposium on Advances in Control Education. (ACE 2013). Sheffield (Reino Unido). Actas del Congreso, 28-30 de Agosto de 2013.

Sánchez, R., Andújar, J.M., Mejías, A. y Márquez, M. **Testing Bench for Remote Practical Training in Electric Machines**. The 10th IFAC Symposium on Advances in

Control Education. (ACE 2013). Sheffield (Reino Unido). Actas del Congreso, 28-30 de Agosto de 2013.

Esquembre, F., De la Torre, L., Heradio, R., Mejías, A. y Márquez, M. **Easy Java Simulations meets Moodle, Arduino and (walks towards) the Ipad.** *Winter Meeting American Association of Physics Teachers.* Nueva Orleans (EEUU). Ponencia invitada, 2013.

Esquembre, F., Mejías, A. y Márquez, M. **Connecting Coach Equipment to Easy Java Simulations.** *The World Conference on Physics Education.* Estambul (Turquía). Actas del Congreso, 2012.

Mejías, A., Andújar, J.M. y Márquez, M. **Desarrollo de Herramientas y Técnicas de Realidad Aumentada para la Mejora de Laboratorios Remotos: Laboratorio Remoto Aumentado.** *VI Jornadas CEA de enseñanza a través de INTERNET-WEB de la Ingeniería de y Sistemas y Automática (EIWISA' 2010).* León (España). Actas del Congreso, 2-4 de Junio de 2010.

2. Laboratorios Remotos: Evolución Histórica y Análisis de Características

El objetivo principal de esta Tesis es modelar de una forma genérica y con una visión sistémica el problema de la realización de una CPLR. A partir de este modelo, plasmado en el capítulo siguiente, se desarrollará una solución tecnológica óptima de carácter general, que asistirá al diseñador de experiencias de acceso remoto en la tarea de su implementación física.

El punto de partida para el estudio de la evolución de las tecnologías existentes lo conforma el conjunto de experiencias remotas actualmente en funcionamiento, que presentan una naturaleza experimental propia del área de conocimiento en la que cada una se encuadra. Como se verá en este capítulo, la solución aportada en cada caso es en sí una solución particular vinculada a su naturaleza experimental, con una fuerte especificidad, lo que dificulta su generalización debido precisamente a esta vinculación a un área de conocimiento concreta.

El problema de acceso remoto a experiencias de laboratorio comprende tres aspectos fundamentales: uno de carácter didáctico, otro de comunicaciones y otro asociado al fenómeno experimental. La parte didáctica está resuelta en la experiencia clásica de la que aquella deriva, aunque presenta características similares entre experiencias de

distintas áreas de conocimiento: en todos los casos se pretende que el alumno adquiriera habilidades experimentales, de toma de contacto con elementos reales. Las comunicaciones también presentan características similares en las distintas experiencias remotas existentes ya que cualquiera de ellas, con independencia de su naturaleza experimental, se reduce a la transferencia de información a través de una red de datos. De este modo, es el fenómeno experimental el aspecto diferencial del conjunto de experiencias remotas. Una solución general debería por tanto encontrar un modelo que aporte independencia de la naturaleza experimental.

El proceso enseñanza-aprendizaje debe ser considerado como el fundamental del problema, por lo que este capítulo comienza realizando una clasificación de las soluciones atendiendo a dicho proceso. Para eso, se tienen en cuenta tanto el diseño didáctico de la experiencia como la gestión del aprendizaje a través de la red del conjunto de soluciones publicadas en la Literatura Técnica. Esta clasificación va a permitir, además de estudiar los aspectos relacionados con el diseño y la gestión didáctica en red, analizar los relacionados con la gestión del sistema experimental. Asimismo, la clasificación permitirá obtener un esquema que englobe el funcionamiento de cualquiera de las soluciones estudiadas en el que están presentes:

- ✓ La forma de interacción del usuario con la experiencia.
- ✓ Los componentes necesarios para controlar la experiencia (PC de control de la experiencia, aplicación de control de la experiencia, etc).
- ✓ Los componentes necesarios situados en el laboratorio que permiten realizar el experimento.
- ✓ La arquitectura de comunicaciones que se establece entre todos los componentes que conforman una solución concreta.

Las experiencias han sido clasificadas atendiendo a varios criterios que permiten deducir su evolución en los tres aspectos fundamentales mencionados anteriormente.

Se estudiarán las características aportadas por cada experiencia en los siguientes aspectos:

- ✓ La interacción del alumno con la experiencia mediante una interfaz de usuario.
- ✓ El modelo seguido para resolver las comunicaciones.
- ✓ El lenguaje utilizado para implementar el programa que controla el experimento remoto.
- ✓ Los aspectos relacionados con la gestión experimental y con la gestión del aprendizaje.

2.1. Evolución y Análisis de las Clases Prácticas de Laboratorio Remoto (CPLR) según un Enfoque Generacional

Debido a la inviabilidad de presentar en detalle desde las primeras experiencias hasta las soluciones más actuales, se ha escogido una muestra de ellas que comprende, además de aquellas consideradas pioneras, las que realizan aportaciones innovadoras en cualquiera de los tres aspectos que se han mencionado como fundamentales (didáctico, de comunicaciones y asociado al fenómeno experimental) y las que han tenido un gran impacto en la posterior evolución de las soluciones, o que están en explotación en la actualidad.

Según [Benmohamed et al. 2005], se han planteado tres enfoques generales distintos para realizar una CPLR, lo que nos permite clasificar las soluciones particulares en tres generaciones, atendiendo a esos tres enfoques, que se exponen y se analizan en los siguientes subapartados.

2.1.1. Primera Generación de Laboratorios Remotos

La primera generación toma como base los resultados obtenidos a finales de los 80 en la solución a los problemas de teleoperación sobre líneas de comunicaciones, [Anderson & Spong 1988]. Desde principios de los 90, esta generación se centra en el telecontrol de robots a través de Internet. Estas primeras soluciones estaban más orientadas a la realización de pruebas en entornos con retrasos variables que a las posibilidades de acceso que brindaba la expansión de Internet, [Dalton 2001], pero sirvieron de base para plantear el acceso remoto a los laboratorios de prácticas de las Escuelas Técnicas y los Centros de Investigación.

2.1.1.1. Proyectos Representativos de la Primera Generación

Entre los proyectos pioneros de telerobótica más citados que se desarrollan a mediados de los 90, se encuentran el proyecto Mercury, [Mercury 2015], [Goldberg et al. 1995], [Goldberg e al. 1999], y el proyecto Telerobot, del año 1995, [Dalton 2001], [UWA telerobot 2015], [Taylor & Trevelyan 1995]. El proyecto Mercury combina la robótica y la arqueología en una instalación interactiva a través de Internet. Fue desarrollado en la Universidad de California, Berkeley, y fue el primer sistema que permitía a los usuarios de un navegador web ver y alterar de forma remota un sistema situado en un laboratorio. Los usuarios debían identificar un conjunto de artefactos ocultos bajo un terrario lleno de arena en un laboratorio situado en Los Ángeles (EEUU) con ayuda del robot. El proyecto Mercury estuvo activo desde el 1 de septiembre de 1994 al 31 de marzo de 1995 y en ese tiempo recibió más de 2,5 millones de visitas. En la figura 2.1 se muestra la interfaz del espacio de manipulación de la experiencia (interfaz del usuario), en la 2.2 el espacio de trabajo y en la 2.3 el esquema del robot.

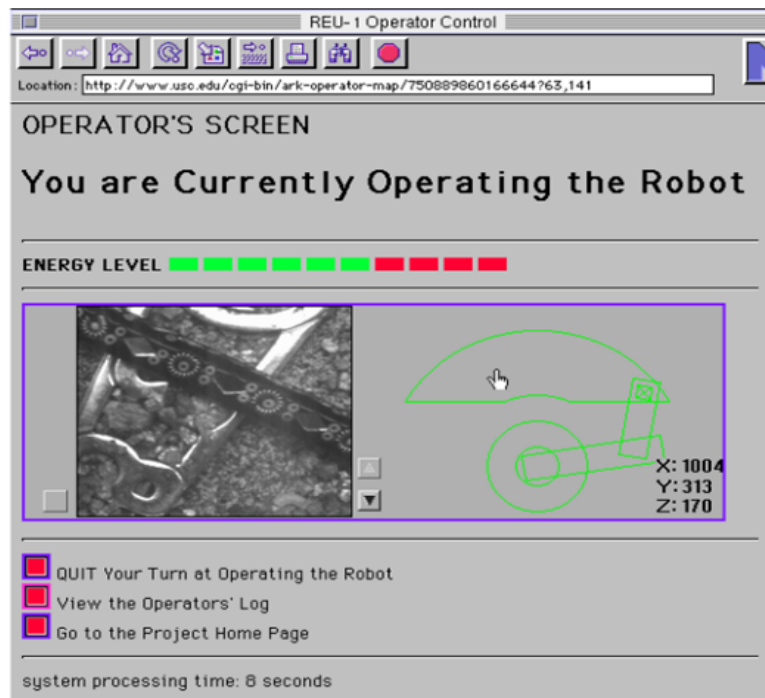


Figura 2.1: Interfaz de usuario del proyecto Mercury.

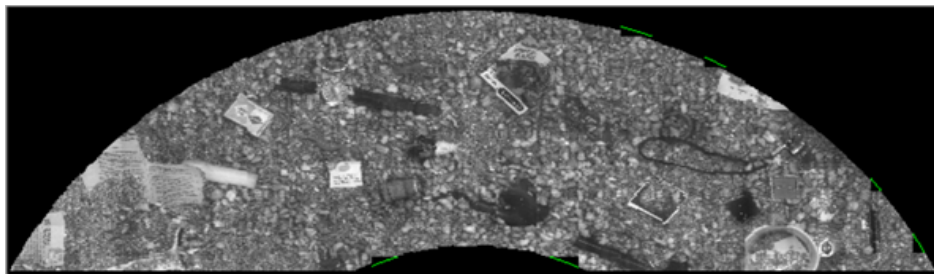


Figura 2.2: Espacio de trabajo (Proyecto Mercury)

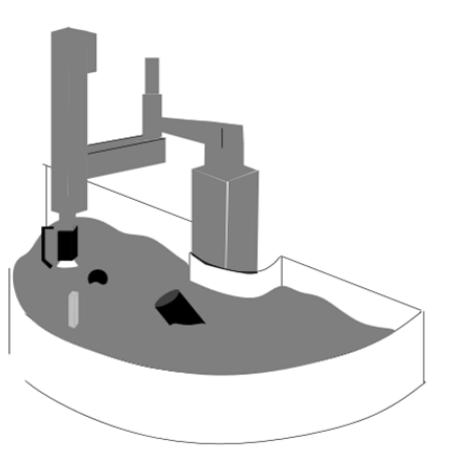


Figura 2.3: Esquema del robot (Proyecto Mercury)

El proyecto Telerobot, [Taylor & Trevelyan 1995], fue desarrollado en la Universidad del Oeste de Australia y posibilitaba el telecontrol de un robot industrial de 6 ejes para manipular bloques de madera utilizando la tecnología web. La interfaz de usuario del alumno incluía las imágenes reales del movimiento del robot. Este proyecto ha estado activo durante 15 años, desde 1994 hasta 2009. Actualmente, está accesible solo para los estudiantes de la UWA y otros grupos bajo petición. El proyecto ha estado en continua actualización desde su implementación. La gestión de la experiencia se realiza a través de una página web desde donde el alumno se identifica y recibe las instrucciones necesarias. La interfaz de usuario está diseñada usando *Labview* (este hecho es importante de cara a evaluar el uso de software de libre distribución como aspecto prioritario en la solución general objeto de esta Tesis). En la figura 2.4 se muestra el material situado en el laboratorio del proyecto Telerobot, así como su interfaz de usuario.

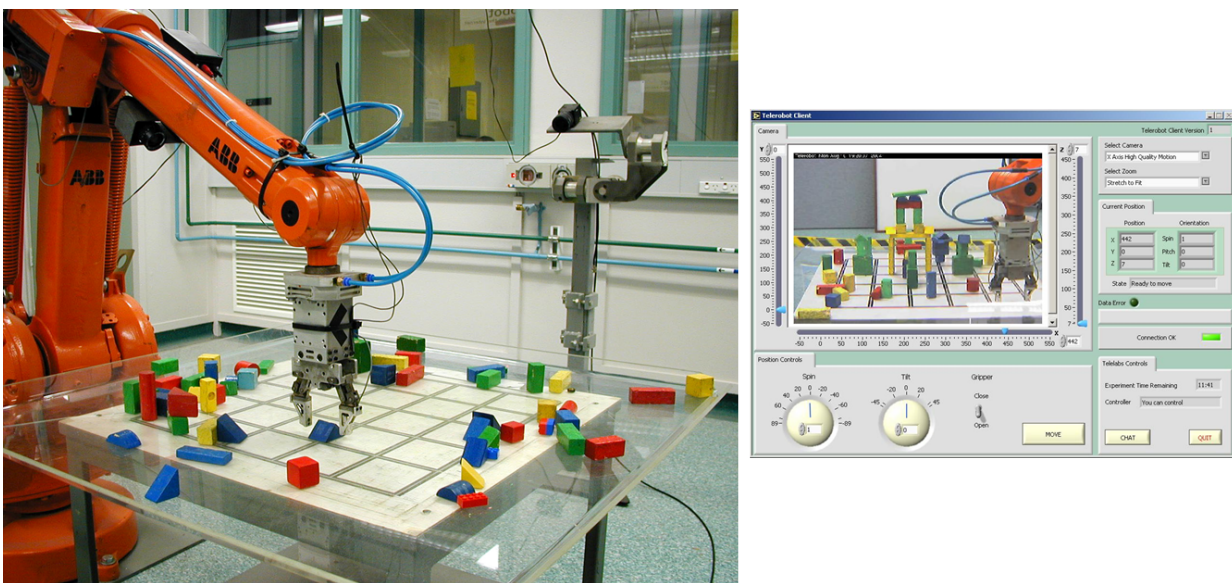


Figura 2.4: Proyecto Telerobot.

A mediados de los 90, todavía en la primera generación, empiezan a aparecer los primeros trabajos que hacen referencia expresa a la terminología propia del concepto LR. Entre los pioneros se puede citar a Molly Shor, de la Universidad Estatal de Oregon,

(EE.UU.) y a Jim Henry, de la Universidad de Tennessee en Chattanooga, (EE.UU.), [García-Zubía & Alves 2011].

En estos primeros trabajos, las soluciones al problema de acceso remoto a una práctica de laboratorio se plantea más en términos de control sobre Internet que en términos relacionados con el aprendizaje a través de la red, [Bohus et al. 1996], [Aktan et al. 1996]. Así, estas soluciones están centradas en resolver los problemas de comunicaciones y los asociados a la monitorización y control de los dispositivos que están dispuestos en el laboratorio para realizar una práctica. En [Bohus et al. 1996] se presentan los cinco paradigmas a resolver desde el punto de vista de la teleoperación en el acceso remoto de experiencias:

- ✓ el experimento,
- ✓ su control,
- ✓ la presencia en el laboratorio o la percepción experimental,
- ✓ las posibilidades de trabajo colaborativo,
- ✓ y la seguridad de los elementos del sistema experimental.

En este trabajo se propone además la arquitectura de comunicaciones cliente servidor como modelo a seguir para resolver el problema de comunicaciones.

Además, en esta práctica remota también se define una interfaz de usuario que puede observarse en la figura 2.5 para la interacción del alumno con el material didáctico del laboratorio, que posibilitaba la modificación de los parámetros necesarios y de la programación para el correcto desarrollo del experimento de laboratorio.

El desarrollo experimental planteaba al alumno el control remoto de un robot, que se muestra en la Figura 2.6. El alumno sigue la evolución experimental a través de una cámara y un micrófono situados en el laboratorio. La arquitectura, figura 2.6, incluye además un conjunto de sensores y actuadores que permiten el correcto posicionamiento

del robot. En la figura 2.7 se muestran también las interfaces para su conexión con el conjunto de computadores comunicados entre sí, que permiten el acceso a toda la información experimental desde red. Además de los sensores y actuadores se incluía un bloque de restricciones que garantizaba la seguridad de todos los componentes.

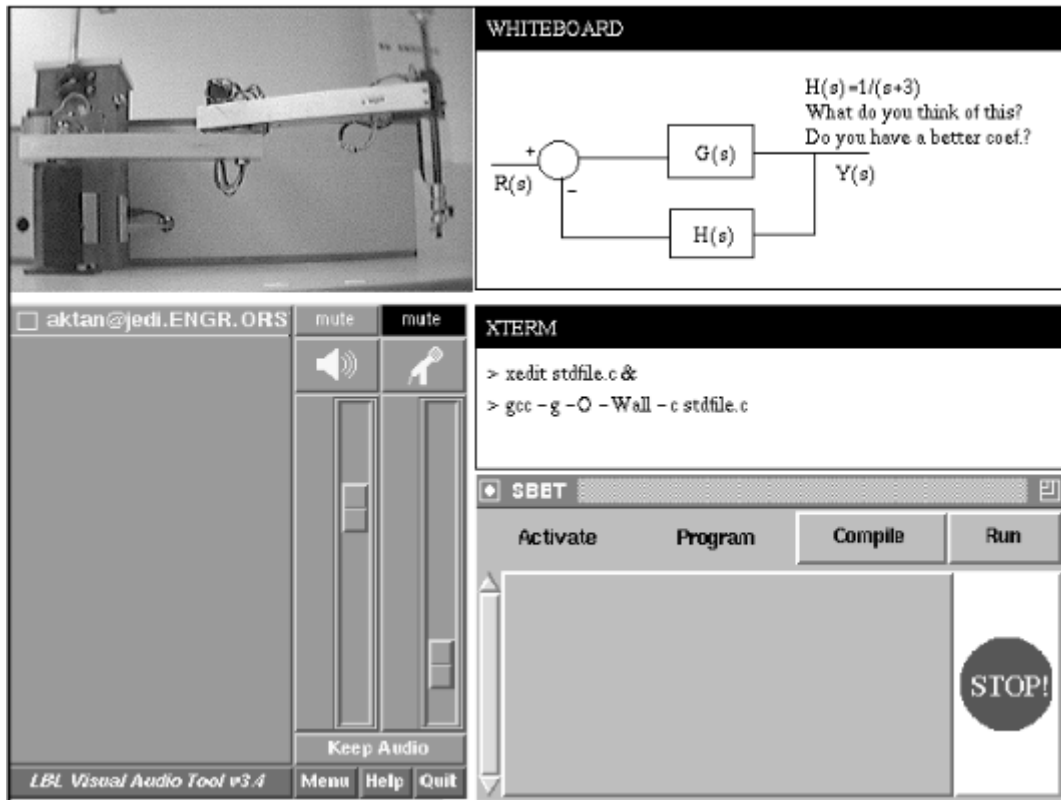


Figura 2.5: Interfaz de usuario del experimento presentado en [Bohus et al. 1996].

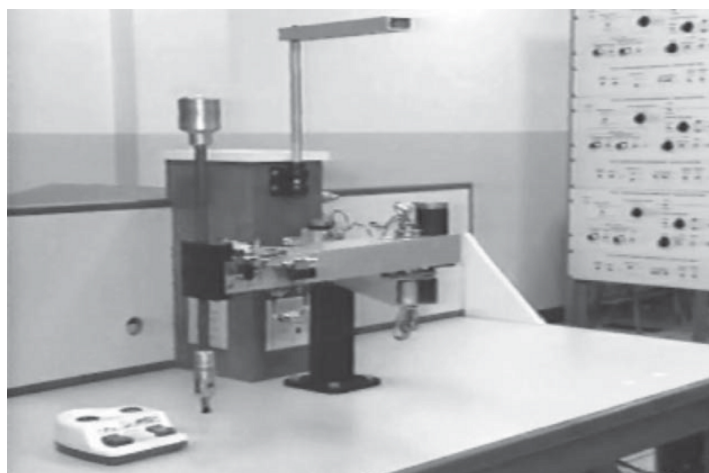


Figura 2.6: Sistema experimental de la experiencia propuesta en [Bohus et al. 1996].

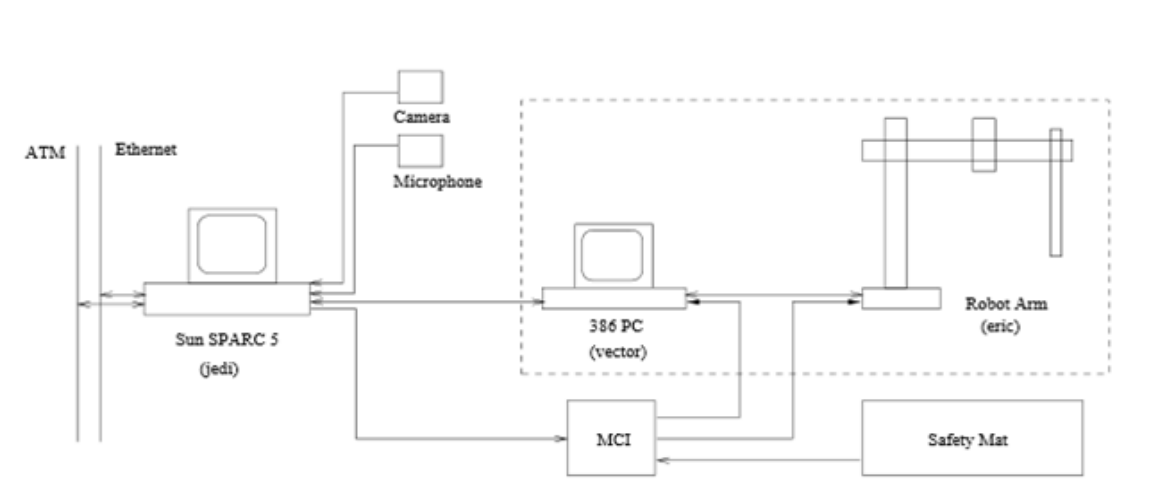


Figura 2.7: Estructura general del experimento de laboratorio propuesto en [Bohus et al. 1996].

Otra de las experiencias pioneras en el acceso remoto a laboratorios es el proyecto desarrollado en la Universidad de Tennessee en Chattanooga, [Henry 1999], [Henry and Knight 2000], que comprende un conjunto de experimentos de laboratorio, publicados en Internet desde el año 1995, sobre control de procesos de forma remota. En el año 1997 se disponía ya de acceso a seis sistemas experimentales, que son los siguientes:

- ✓ Control de presión mediante la variación de la velocidad de una soplante.
- ✓ Control de nivel en un tanque de agua mediante la variación de la velocidad de una bomba de agua.
- ✓ Control de flujo en un bucle cerrado variando la velocidad de una bomba.
- ✓ Control de velocidad de un motor-generator mediante la variación de la tensión de alimentación del motor.
- ✓ Control de la posición en un coche sobre raíl variando el par de un motor.
- ✓ Control de la temperatura en un intercambiador de calor mediante la variación del flujo de agua caliente.

En estos experimentos de laboratorio se aprecia la tendencia a utilizar computadores de tipo personal de Apple o Compatibles PC en la arquitectura hardware para la

implementación del control del experimento de laboratorio. Asimismo, se introduce el uso de *Labview* como herramienta de desarrollo para realizar este control.

Este proyecto se ha seguido desarrollando y estas experiencias están activas y accesibles actualmente [LWB 2015]. Mediante interfaces en *Labview* se propone a los estudiantes la identificación de los sistemas citados arriba y el diseño de controladores para los mismos. En la figura 2.8 se muestra una CPLR para realizar el control de niveles en cuatro tanques, y en la figura 2.9 se muestra el interfaz de usuario diseñado con *Labview*.

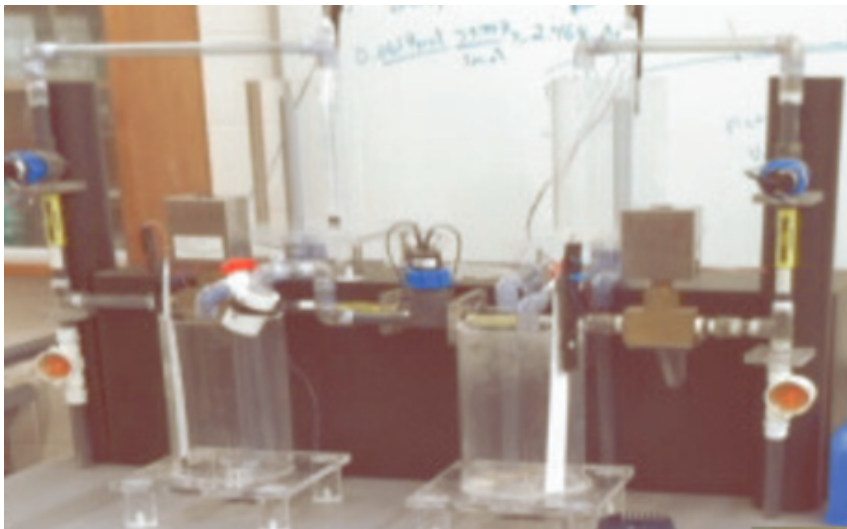


Figura 2.8: Sistema experimental de control de niveles en cuatro tanques.

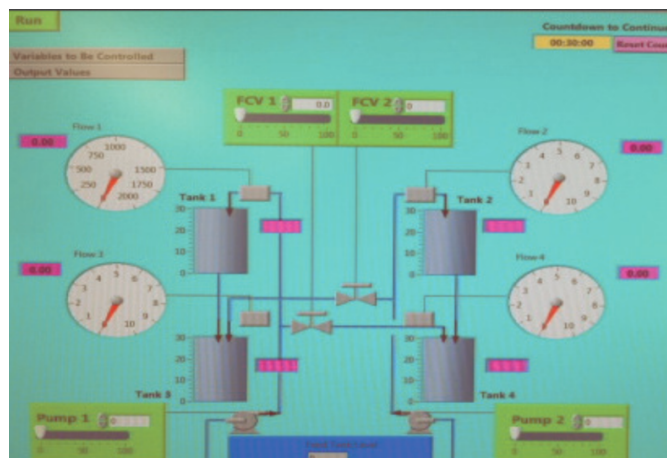


Figura 2.9: Interfaz de usuario desarrollado en Labview del sistema experimental de la figura 2.8.

En la arquitectura anterior los ordenadores están conectados en red mediante una tarjeta Ethernet y tienen una dirección IP válida de Internet. Se ha usado un servidor web para que los usuarios puedan acceder a los experimentos mediante un navegador. Los usuarios podían realizar los experimentos desde los laboratorios de computación del campus o desde sus equipos domésticos a través de los proveedores de Internet.

2.1.1.2. Análisis de la Primera Generación

En esta primera etapa, que se puede denominar experimental o de los pioneros, cualquier concepto sienta las bases de la arquitectura de bloques necesaria para lograr la implementación de una solución al problema del control remoto de un experimento de laboratorio. De hecho, con muy pocas variaciones y matices, la estructura de estas experiencias se puede encontrar en casi todas las implementaciones actuales.

Del análisis de la estructura funcional utilizada en las soluciones de esta generación se puede deducir una arquitectura genérica y abstracta que representa de forma general el esquema de funcionamiento presente en todas las soluciones estudiadas y que se muestra en la figura 2.10.



Figura 2.10: Arquitectura funcional de bloques de un experimento de laboratorio remoto genérico.

En esta estructura funcional se hace referencia a tres componentes que están presentes en todas las soluciones: el sistema de acceso a la experiencia, Internet y el sistema experimental.

El sistema de acceso a la experiencia está constituido por el PC del alumno en el que se ejecuta una aplicación que es la interfaz de usuario con la experiencia y que

proporciona al alumno la capacidad de interactuar con ella y de percibir la sensación de estar realizando una experiencia real.

Internet es la red utilizada para unir el sistema de acceso a la experiencia con el sistema experimental.

El sistema experimental es una abstracción introducida para englobar todas las funcionalidades necesarias para realizar el telecontrol de un experimento de laboratorio de forma remota. Se encuentra situado en el laboratorio y comprende todos los componentes necesarios para la realización remota de una experiencia concreta. Es importante destacar que la naturaleza de estos elementos está fuertemente influenciada por el área de conocimiento al que pertenece la experiencia, como por ejemplo el tipo de elemento de medida utilizado, los niveles de seguridad que se deben establecer, etc. De esta forma, de las soluciones denominadas pioneras, se deduce que el sistema experimental se puede dividir en tres bloques: la planta experimental, el servidor multimedia y el servidor de control experimental.

La **planta experimental** está generalmente constituida por maquetas didácticas, prototipos experimentales, instrumentos de medida, etc. Representa por tanto la abstracción que engloba todas las características asociadas al fenómeno experimental concreto de una actividad práctica, y en su interior se desarrolla un conjunto de procesos experimentales que tienen interés didáctico.

El **servidor multimedia** es el sistema encargado de trasladar al alumno la percepción de la evolución del proceso experimental. Desde la interfaz de usuario se proporciona al usuario el sonido (en caso necesario) y la visión de lo que sucede en el laboratorio, mediante la imagen de una cámara de vídeo y un micrófono. Este sistema puede ser un servidor multimedia, o puede estar implementado sobre un PC al que se conecta una cámara de vídeo y un micrófono, y en el que se ejecuta una aplicación que permite la difusión de audio y vídeo por red.

2. Laboratorios Remotos: Evolución Histórica y Análisis de Características

El **servidor de control experimental** suele ser un PC que tiene acceso a través de alguno de sus buses de expansión a tarjetas de adquisición de datos y a actuadores para leer y actuar, respectivamente, sobre la evolución de los procesos que se producen en la planta experimental. En el servidor de control de la experiencia se ejecuta la aplicación de control experimental que permite a la interfaz de usuario acceder a las tarjetas de adquisición y actuación para controlar la evolución del sistema experimental a través de Internet, convirtiéndolo así en un sistema experimental remoto totalmente operativo.

Las funcionalidades presentes en las soluciones pioneras se resumen en la estructura de bloques mostrada en la figura 2.11, aunque la forma concreta de implementación puede variar de una solución a otra. Por ejemplo, en un gran número de casos el servidor de control experimental está constituido por un PC, pero en otros está formado por un grupo de computadores interconectados entre sí o por un sistema digital basado en microcontroladores, PLCs, controladores digitales, sistemas de domótica, etc.

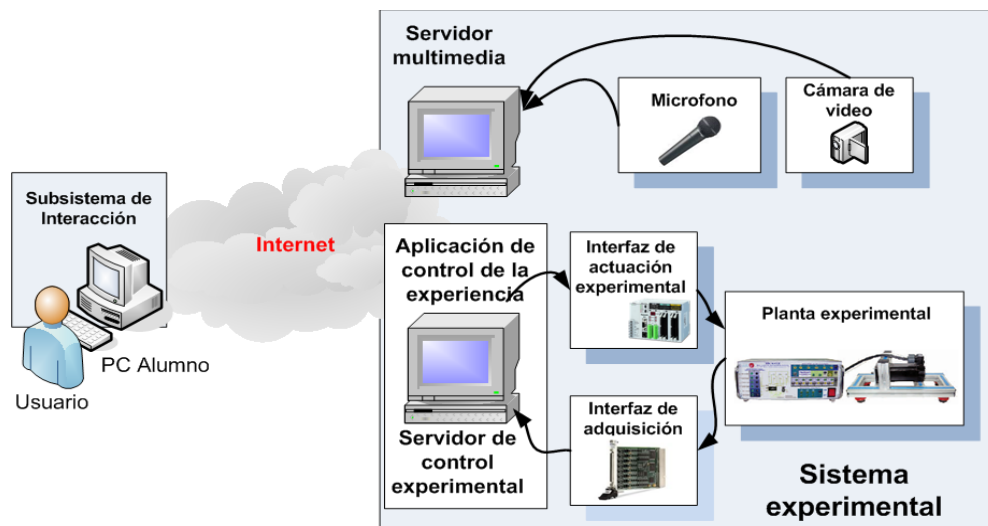


Figura 2.11: Detalles de la arquitectura funcional de bloques de un experimento remoto genérico.

Por tanto, en esta primera generación, además de sentar las bases de la estructura funcional para el control remoto de experimentos de laboratorio, se postula el modelo

cliente-servidor como referencia para resolver el problema de las comunicaciones, aunque la implementación de este modelo queda abierta a un gran número de posibilidades de arquitectura. Asimismo, en esta generación se empieza a hacer uso de *Labview* como lenguaje de desarrollo para la interfaz de usuario, lo que convierte a esta herramienta en una de las soluciones pioneras en este campo.

Teniendo en cuenta lo expuesto anteriormente, esta primera generación de LR se caracteriza por carecer de enfoque pedagógico claro. Su diseño va más orientado a aprovechar la posibilidad de controlar de forma remota un *experimento de laboratorio* a través de Internet, que a la necesidad de desarrollar unas competencias a través de la realización de una actividad práctica. En ese sentido, es significativo que en las publicaciones de la generación pionera no haya referencias al diseño didáctico de las experiencias. Así, en esta primera etapa, el centro del problema es conseguir el control remoto del experimento y que el alumno perciba que está controlando el *experimento de laboratorio* a través de la interfaz de usuario.

Por otra parte, el hecho de no haber tenido en cuenta los aspectos relacionados con el aprendizaje a través de la red, hace que no estén resueltas cuestiones como quién puede hacer una experiencia, cómo hacer llegar al alumno el material necesario para su desarrollo, cómo evaluarla, etc. Todas estas cuestiones, relacionadas con los aspectos de gestión del aprendizaje y de la gestión de los recursos necesarios para la realización del *experimento de laboratorio*, se abordan en la segunda generación.

2.1.2. Segunda Generación de Laboratorios Remotos

En esta segunda generación el uso de la tecnología web se consolida como marco de acceso del alumno al LR y toda la interacción que realiza sobre éste se hace a través de un navegador. Esto motiva a los diseñadores a resolver los problemas de gestión de la primera generación, utilizando directamente las capacidades de difusión de

información que esta tecnología aporta, incluyendo el servidor web en la arquitectura como responsable de la difusión de contenidos. En esta segunda generación se empiezan a introducir también en la arquitectura funcional los componentes destinados a la gestión del aprendizaje y de los recursos experimentales. Además, en las publicaciones empiezan a aparecer detalles relacionados con el diseño didáctico, como por ejemplo objetivos, metodologías de uso, valoraciones del rendimiento de aprendizaje, etc. También se continúa con la expansión en el número de áreas de conocimiento, abarcando experiencias de robótica, automática, electrónica analógica, electrónica digital, microelectrónica, física, química, etc.

2.1.2.1. Proyectos Representativos de la Segunda Generación

Entre los proyectos pioneros que evolucionan hasta esta segunda generación se encuentran los realizados por el Prof. Jesús del Álamo, del Departamento de Ingeniería Eléctrica y Ciencias de la Computación del MIT, que desde 1988 empieza a trabajar en el desarrollo de un laboratorio de acceso remoto de microelectrónica denominado inicialmente WebLab, [WebLab 2014], [Alamo 2000]. La primera versión del sistema de Microelectrónica WebLab se pone en funcionamiento en otoño de 1988 y desde entonces se han utilizado versiones mejoradas en cursos de postgrado, que pueden considerarse integradas en la segunda generación.

El éxito del proyecto de Microelectrónica WebLab ha dado lugar a la iniciativa iLab en la que se ha extendido el concepto WebLab a otras disciplinas de la ingeniería. El proyecto iLab se ha desarrollado en el MIT, [García-Zubía & Alves 2011], y consiste en un framework que permite el desarrollo de experimentos de laboratorio remotos con componentes reales: microelectrónica, amplificadores operacionales, principios de comunicación, servomecanismos, electrónica digital básica y avanzada, robótica, etc. Se trata de un conjunto de laboratorios basados en *Labview* para el estudio de la corriente continua y los principios de las telecomunicaciones. La figura 2.12 muestra la

arquitectura de iLab, donde puede observarse un servidor denominado *Service Broker*, que se encarga de las funciones de gestión del experimento remoto.

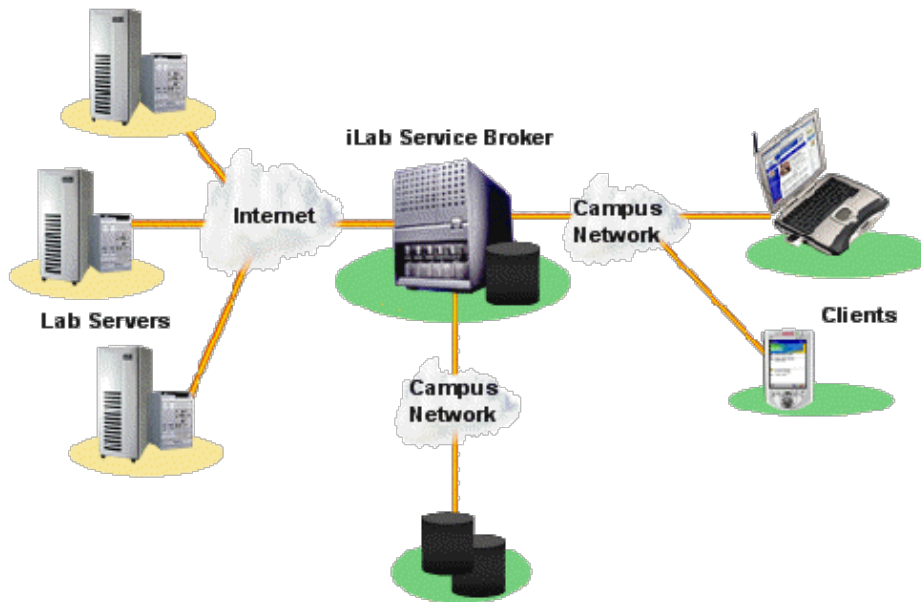


Figura 2.12: Arquitectura básica i-Lab.

En iLab existe una arquitectura distribuida de experiencias remotas en la que cada servidor *Lab Server* se encarga del control de un *sistema experimental* concreto, dejando las labores de gestión al servidor *Service Broker*. El inicio de cualquier experimento en un *Lab Server* pasa porque esté registrado en el servidor *Service Broker* y porque inicie la sesión a través de su interfaz web (figura 2.13). Esta interfaz le facilita al estudiante las siguientes actividades: registra una cuenta de usuario para tener acceso a las experiencias, descarga el cliente para la realización de las experiencias en su sistema, redacta las especificaciones de configuración del laboratorio, carga experimentos anteriores, etc.

Desde esa interfaz se solicita el acceso a las experiencias comentadas anteriormente. Una vez seleccionada una de ellas, se puede acceder al área de asistencia de la experiencia, figura 2.14, donde se pueden consultar manuales, tutoriales, FAQ, etc.

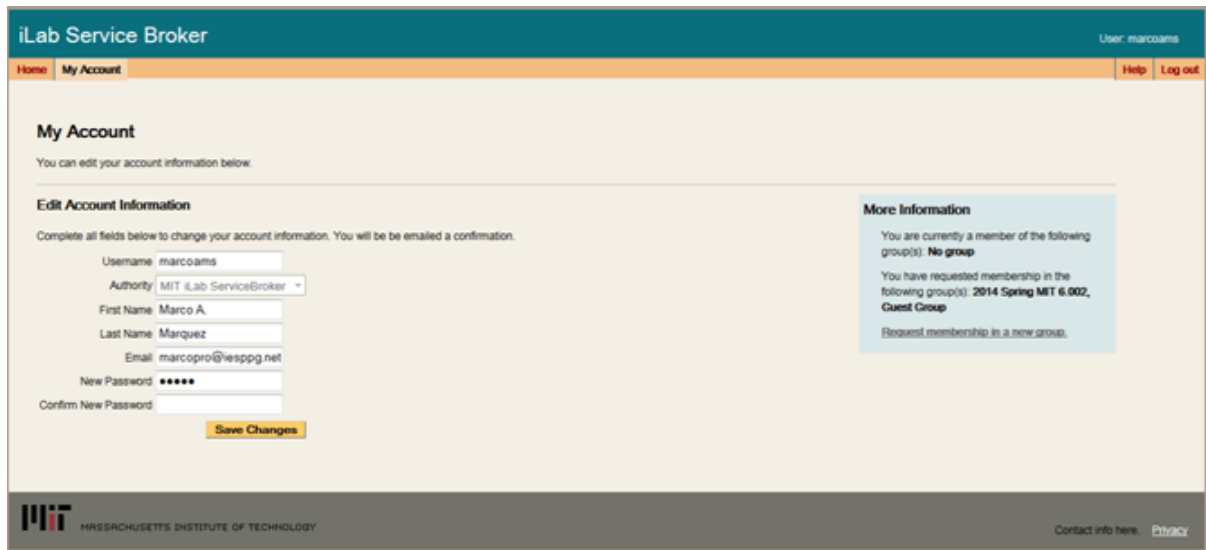


Figura 2.13: Interfaz web de i-Lab.

Microelectronics WebLab Documentation

[Overview](#) | [System Requirements](#) | [Manual](#) | [Tutorials](#) | [FAQs](#) | [For Educators](#)

[A simple experiment \(pn diode\)](#) **Basic Diode Experiment**

[A more detailed experiment \(NMOS\)](#)

First, you will need to create an account at [the WebLab web site](#). Account group permissions are manually approved; as a result, there may be up to a 24 hour delay between when you create the account, and when you can run the WebLab experiments. Once your account has been enabled, log in, and select the group appropriate for your class. Select the **graphical client**, and from there, select **Launch Client**. You may be asked whether you authorize the downloading of the WebLab client software to your computer; approve this. After a short download and startup time, the client should appear in a new window:

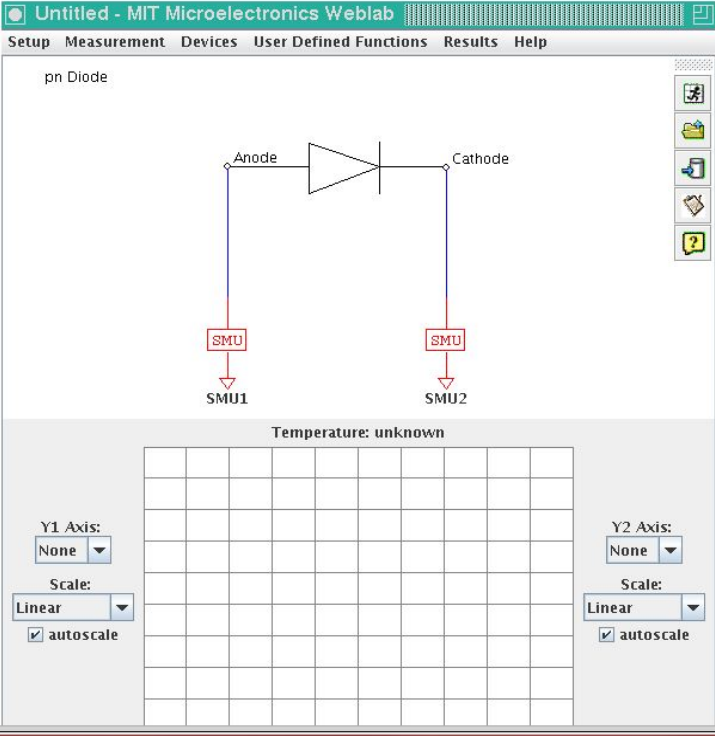


Figura 2.14: Área de asistencia de la experiencia en i-Lab.

Otro proyecto importante que podemos considerar perteneciente a esta segunda generación es el denominado Telelab, [Casini et al. 2003], [Casini et al. 2004], del profesor Marco Casini de la Universidad de Siena. En este proyecto se puede acceder a experiencias de:

- ✓ Control de velocidad de motores DC.
- ✓ Control de posición de motores DC.
- ✓ Control de nivel en un tanque.
- ✓ Control de flujo en un tanque.
- ✓ Control de levitación magnética.
- ✓ Control de un helicóptero.
- ✓ Control de robots.

The image shows the 'Automatic Control Telelab' portal. At the top, the title 'Automatic Control Telelab' is displayed in a stylized font. Below the title, there is a navigation menu on the left with the 'ACT' logo and links for 'Introduction', 'Lab Manual', 'Experiments', 'Competitions', 'People', 'FAQ', 'Reports', 'Links', and 'ACT Home'. The main content area is titled 'Remote Experiments' and contains a grid of experiment cards. Each card includes a title, a small image, and a list of links: 'System Description', 'Video Sample', 'On-line Camera', 'Control Experiment', and 'Process Ready'. The 'Process Ready' status is indicated by a green traffic light icon.

Remote Experiments		
DC Motor Position Control System Description Video Sample On-line Camera Control Experiment  Process Ready	Water Tank Level Control System Description Video Sample On-line Camera Control Experiment  Process Ready	Magnetic Levitation Stable Configuration System Description Video Sample On-line Camera Control Experiment Student Competition  Process Ready
DC Motor Speed Control System Description Video Sample On-line Camera Control Experiment Identification Experiment  Process Ready	Water Tank Flow Control System Description Video Sample On-line Camera Control Experiment  Process Ready	Magnetic Levitation Unstable Configuration System Description Video Sample On-line Camera Control Experiment  Process Ready
Helicopter Simulator 2 DOF Helicopter Control System Description Video Sample On-line Camera Control Experiment Student Competition	Team of Mobile Robots 	

Figura 2.15: Portal del proyecto Telelab (selección de experimentos).

En la figura 2.15 se muestra el menú de experiencias ofertadas en el que se presenta una descripción de cada una de ellas, un vídeo de demostración de su desarrollo, un enlace a la cámara que la visualiza y un enlace de acceso para su realización. Para controlar el acceso, cada experiencia dispone de un semáforo que se muestra en verde cuando la experiencia esta libre. En la figura 2.16, se muestra un fotograma de uno de los experimentos que pueden llevarse a cabo.

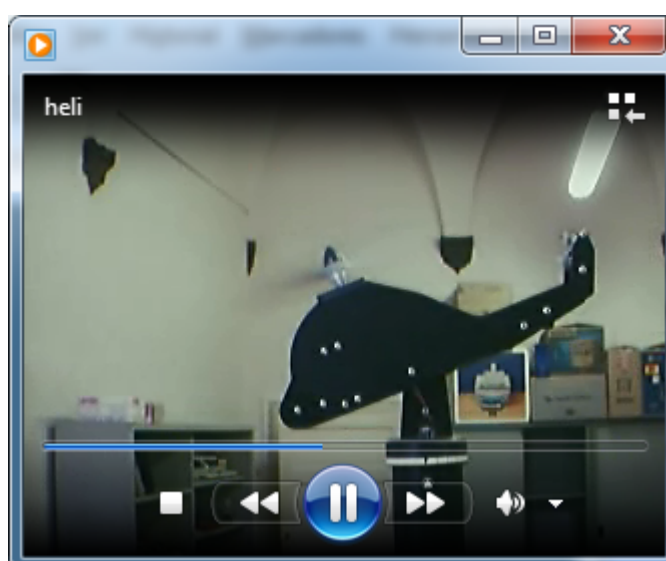


Figura 2.16: Fotograma de un video de demostración en Telelab.

2.1.2.2. Análisis de la Segunda Generación

Esta segunda generación se caracteriza, frente a la anterior, por la existencia de un diseño en el plano didáctico asociado al desarrollo experimental que regula las actividades, y en el que se facilita información a nivel de *base de conocimiento* o de las instrucciones a seguir por parte del alumno para realizar con éxito el experimento. Sin embargo, frente a la siguiente generación, tiene el inconveniente de que las experiencias están implementadas con herramientas especializadas y lenguajes de desarrollo que requieren conocimientos específicos en dichas herramientas, lo que dificulta el diseño

de experimentos de laboratorio remotos a usuarios que no tengan un alto nivel de estos conocimientos.

El diseñador del *experimento de laboratorio remoto* y el usuario disponen de una interfaz de aprendizaje restringida y estática, es decir, sin mecanismos de actualización de contenidos, ya que esos contenidos han sido introducidos en la interfaz por programadores utilizando algún lenguaje de desarrollo de aplicaciones que requiere para su uso conocimientos avanzados. Esto hace que el mantenimiento y desarrollo del experimento exijan un alto nivel científico-técnico en los aspectos experimentales, de comunicaciones y didácticos. Las soluciones de esta generación son, por tanto, desarrollos a medida en las que todas las cuestiones relacionadas con el problema son resueltas por el equipo de diseño, a nivel de programación.

2.1.3. Tercera Generación de Laboratorios Remotos

La tercera generación centra los objetivos de acceso remoto a experimentos de laboratorio en los aspectos didácticos. En este sentido, hasta el año 2001, son muy pocas las publicaciones sobre experimentos remotos que toman en consideración el proceso de enseñanza-aprendizaje como uno de los elementos clave de su existencia, [Tuttas and Wagner 2001]. Las enmarcadas en las generaciones anteriores plantean el problema de acceso remoto a los experimentos en términos de enseñanza asistida por ordenador, siguiendo la línea del modelo estático de la enseñanza clásica. Esta tercera generación viene marcada por la evolución pedagógica del proceso enseñanza-aprendizaje desde el modelo estático puramente receptivo de contenidos del enfoque clásico, al nuevo modelo dinámico basado en el constructivismo. Este modelo dinámico plantea el aprendizaje como un proceso activo en el que la interacción con el entorno es fundamental. Por tanto, en este nuevo enfoque, el aprendizaje experimental a través de Internet se convierte en un medio constructivista, donde el conocimiento llega al estudiante a través de la interacción con la experiencia de manera activa (aprender

haciendo). Se introduce pues la necesidad de un diseño didáctico basado en la actividad del alumno y con metodologías orientadas a la adquisición de capacidades.

2.1.3.1. Proyectos Representativos de la Tercera Generación

Uno de los proyectos que ha aportado desarrollos a esta tercera generación nace en las I Jornadas de Trabajo sobre *Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática* (EIWISA00), celebradas en el año 2000 en la Universidad Politécnica de Valencia. Todos los grupos de trabajo que participaron se plantearon el objetivo a largo plazo de diseñar y construir una red de laboratorios remotos de Automática que permitiera compartir los recursos propios de cada Universidad. Las Universidades que participaban en el proyecto eran las siguientes: Universidad Nacional de Educación a Distancia (UNED), coordinadora del Proyecto, Universidad Politécnica de Barcelona, Universidad Politécnica de Valencia, Universidad Miguel Hernández de Elche, Universidad de León, Universidad de Almería y Universidad de Alicante. El proyecto se denominaba *Automat@Labs*, [Vargas et al. 2011], y aportaba un sistema de reserva de tiempos para la realización de los experimentos y un entorno completo de trabajo con una apariencia común entre todas las experiencias, con el objetivo de facilitar el aprendizaje del alumno. La red de laboratorios remotos en automática *Automat@Labs* es más que la suma de las partes que la constituyen, y fue diseñada para ser percibida por sus usuarios como un único laboratorio con una estructura uniforme independientemente de donde se encontrara la localización física de las plantas.

Para acceder a *Automat@Labs* se necesita únicamente un navegador y estar dado de alta para la realización de las prácticas. Todos los laboratorios comparten un mismo esquema de trabajo y los materiales se proporcionan de forma que las prácticas se pueden realizar de forma autónoma gracias a la utilización de *Easy Java Simulations* (EJS), herramienta usada como base en el diseño de experiencias [EJS 2015]. El proyecto

Automat@Labs está coordinado por el Prof. Dr. Sebastián Dormido Bencomo del Departamento de Informática y Automática de la UNED.

Automat@Labs está accesible en Internet a través del *eMersion* e incorpora tanto elementos para la gestión del experimento como de gestión del aprendizaje. Además, el entorno de realización de experimentos dispone de un gestor de reservas (*BookingServer*), y de un gestor de enseñanza (*eJournal*), que carece de la capacidad de gestión de contenidos, [Andújar y Mateo, 2010]. En la figura 2.17 se muestra la interfaz de usuario. El alumno tiene acceso a los elementos que permiten la realización del experimento y su gestión y a los de asistencia al aprendizaje.



Figura 2.17: Interfaz de usuario del proyecto Automat@Labs.

En la línea de Automat@Labs, la UNED ha ampliado el número de experiencias ofertadas en este proyecto y ha seguido trabajando para conseguir la integración de experiencias en entornos de gestión del aprendizaje. Ha sustituido *eMersion* por *Moodle*

y ha creado módulos que permiten la integración en esta segunda aplicación de experiencias con la interfaz de usuario creada mediante EJS. Además, ha conseguido un mejor soporte de trabajo colaborativo y la integración del gestor de reservas en *Moodle*. La primera implementación de todas estas mejoras ha dado lugar al proyecto UNEDLabs. Posteriormente, el proyecto UNEDLabs ha evolucionado integrando prácticas de laboratorio de distintas universidades españolas, dando lugar a UNILabs [UNILabs 2015].

2.1.3.2. Análisis de la Tercera Generación

En esta tercera generación se funden los enfoques didáctico y tecnológico de la enseñanza en red (*e-learning*). Por tanto, se incorporan los aspectos relacionados con la gestión de la información necesaria para la realización de la experiencia de laboratorio (marco teórico, manuales de usuario, etc.) en un gestor de aprendizaje. De este modo, se introduce en el esquema global del problema el proceso de enseñanza-aprendizaje como un elemento esencial, así como el elemento gestor de dicho proceso, proponiendo así un modelo más activo para las enseñanzas en ingeniería. En este sentido, los comités relacionados con estas enseñanzas empiezan a emitir informes en el año 2004 en los que se hace hincapié en el carácter innovador y dinámico de las enseñanzas superiores, [García-Zubía & Alves 2011], así como la enseñanza de la electrónica en la Unión Europea. Estas directrices empiezan a emerger en todos los desarrollos relacionados con las enseñanzas y las TIC.

De esta forma, lo que caracteriza fundamentalmente a una experiencia de esta tercera generación es la inclusión de sistemas dinámicos que permite a los diseñadores tanto la gestión técnica de la experiencia como la del aprendizaje, posibilitando el desarrollo de metodologías que promueven la actitud activa del alumno. Es decir, las experiencias pueden ser ahora diseñadas de forma dinámica a través de gestores de contenidos por usuarios sin conocimientos de programación avanzados. En este sentido, el uso de un

gestor de aprendizaje (LMS) es la solución más utilizada, aunque no es la única. De hecho, existen otros entornos para el desarrollo de aplicaciones que cumplen con este conjunto de especificaciones [Labshare 2015], [SAHARA 2015]. El uso de un LMS permite integrar la interfaz de usuario con el experimento como un componente más del LMS, o incluir módulos para la gestión de reservas de los sistemas experimentales. La creación de experiencias remotas se hace más fácil en esta generación, lo que permite extender su uso a un gran número de profesores e incluso a los alumnos, lo que va en la línea de convertir al alumno en protagonista de su propio aprendizaje, que realiza de forma activa.

En resumen, la tercera generación de LR se caracteriza por potenciar los aspectos de aprendizaje activo en red y para ello incluye un gestor de aprendizaje, LMS, como componente esencial de la estructura general de bloques. Además, se le añaden las funciones de gestión del experimento y la interfaz para el desarrollo de experiencias. Los LMS facilitan la capacidad de gestionar dinámicamente los contenidos de aprendizaje sin tener que realizar labores de programación. Incluyen además la posibilidad de utilizar distintas estrategias didácticas y un importante número de servicios para el intercambio de información, servicios para el seguimiento de la traza de aprendizaje, pruebas de evaluación, funciones de enseñanza en vivo, etc. Esta nueva estructura supone un avance en la línea de facilitar el diseño de experimentos a profesores y alumnos.

Si se introduce un servidor que se encargue de los aspectos relacionados con la gestión del aprendizaje y del desarrollo del experimento, se obtiene el diagrama genérico que representa todas las experiencias de la segunda generación. Además, si ese servicio es prestado por un LMS en el que se integran todas las funcionalidades, el esquema representa las experiencias de la tercera generación. Por tanto, el esquema mostrado en la figura 2.18 representa, con sus correspondientes matices, cualquiera de las soluciones estudiadas en la segunda y tercera generación. En este nuevo escenario, el

desarrollo propiamente dicho del *experimento de laboratorio remoto* es para el alumno una actividad más en el LMS.

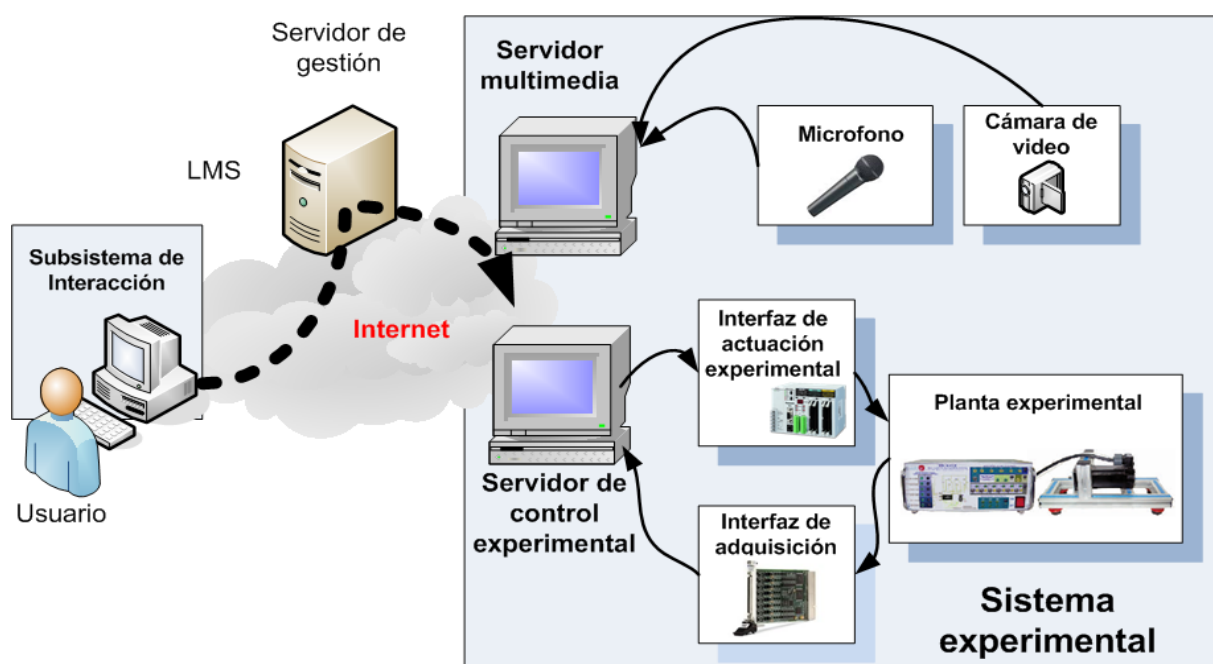


Figura 2.18: Esquema general de la segunda y tercera generación de sistemas.

2.2. Evolución y Análisis de las Clases Prácticas de Laboratorio Remoto (CPLR) según el Programa de Control del Experimento y el Subsistema de Interacción

Del análisis de las soluciones estudiadas se deduce que, en todos los casos, el control de la evolución de la experiencia es consecuencia de las actuaciones del alumno sobre su ordenador, siguiendo un determinado esquema de desarrollo experimental. Estas actuaciones generan un flujo de información a través de Internet entre la interfaz de usuario y el software que controla el experimento, que se está ejecutando en el servidor de control experimental, figura 2.19. Asimismo, ese flujo de datos se traduce en las correspondientes interacciones con la planta experimental, a través de las interfaces de actuación y adquisición. Estas interfaces permiten conectar el servidor de control

experimental con la planta experimental, que se encuentra en el laboratorio. Las comunicaciones entre estas dos aplicaciones se realizan siguiendo el modelo cliente-servidor y estas comunicaciones son generadas, por un lado, como consecuencia de las actuaciones del alumno y por otro debido a la evolución del proceso experimental que se desarrolla en la planta. Por todo ello, las características generales de implementación de estas dos aplicaciones y del modelo cliente-servidor son muy relevantes en el análisis del conjunto de soluciones, desde el punto de vista del telecontrol. En este apartado se estudian las alternativas de implementación del software que permite controlar el experimento.

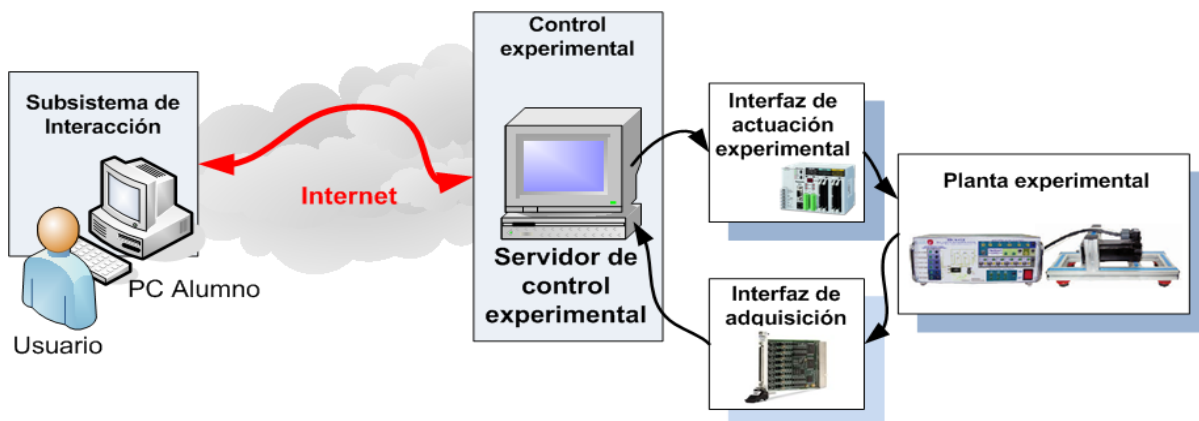


Figura 2.19: Flujos de datos genéricos entre el alumno y la planta experimental

En primer lugar, se abordarán los lenguajes y herramientas de programación utilizados en las distintas soluciones existentes. En general, este programa se puede realizar en cualquier lenguaje de programación, siempre que existan para el mismo, *drivers* para los dispositivos de actuación y adquisición, y siempre que disponga de mecanismos para acceder a red con el modelo cliente-servidor. Este acceso puede realizarse bien a bajo nivel mediante *sockets*, o bien a alto nivel mediante mecanismos de comunicación con un servidor web. De este modo, el lenguaje de implementación del programa que controla el experimento permite clasificar el conjunto de experiencias estudiadas en tres tipos:

1. *Aplicaciones desarrolladas con Labview*. *Labview* es el lenguaje más utilizado en las experiencias publicadas hasta ahora para la realización del programa de control del experimento, [Saad et al. 2001], [Prashant and Yuxiang 2008], [Berntzen et al. 2001], [Chen et al. 1999], [Salzmann et al. 1999], [Gillet et al. 1997], [Golijanek-Jedrzejczyk et al. 2002], [Ko et al. 2002], [Ko et al. 2002a], [Laget et al. 2003]. Eso se debe a que esta herramienta, además de permitir el desarrollo rápido de aplicaciones, es el software más utilizado en los laboratorios para la creación de instrumentos virtuales. Dispone asimismo de una gran cantidad de *drivers* para un gran número de sistemas de actuación y adquisición, adecuados para plantas experimentales, normalmente utilizados en las CPLR. Proporciona además un método de programación gráfico e intuitivo que facilita el desarrollo de este tipo de aplicaciones. Por último, esta plataforma de desarrollo dispone de mecanismos a nivel de *sockets* y de integración en una arquitectura web para poder utilizar los instrumentos virtuales a través de la red.

2. *Aplicaciones desarrolladas con Matlab*. *Matlab* es otra de las herramientas de desarrollo rápido de aplicaciones que aparece en un gran número de publicaciones, [Jiménez et al. 2005], [Casini et al. 2003], [Carusi et al. 2004], [Casini et al. 2004], [Casini et al. 2003a], [Campos et al. 2007], [Dixon et al. 2002], [Junge and Schmid 2000], [Lindsay and Good 2005], para implementar en el computador el programa de control del experimento mediante la utilidad *Matlab Real TimeWorkshop* (RTW). Esta herramienta se usa también ampliamente en los estudios de ciencia y tecnología para la simulación de procesos, para el cálculo matemático a nivel de diseño y para el análisis de problemas de tipo técnico o de ingeniería. Dispone además de una plataforma de diseño gráfico, *Simulink*, que permite el diseño intuitivo, y proporciona mecanismos de comunicaciones a nivel de *sockets* y a través de servicio web. *Matlab* está entre las más utilizadas para la implementación de laboratorios remotos virtuales, [Tilbury and Messner 1998], [Ramos and Ramírez 2005]. En algunos casos suele usarse como motor matemático de la experiencia combinada con *LabView*, [Vargas et al. 2006].

3. *Aplicaciones realizadas con lenguaje de programación de alto nivel.* Además de las dos herramientas comentadas, se han publicado experiencias programadas en lenguajes de alto nivel de propósito general como C, [Fjeldly et al. 1999], C++, [Aliane et al. 2007], Java, [Chang and Hung 2000], [Reguera and Fuertes 2004], [Nakano et al. 2001], Phyton, [Frangu and Chiculita 2010], Visual Basic, [Azorín et al. 2004], y otros.

En el otro extremo de la comunicación, la interfaz de usuario es la encargada de mostrar resultados de una forma interpretable por el alumno, y de permitir sus actuaciones para controlar la evolución de la experiencia. La consolidación del uso de la tecnología web como marco donde presentar las experiencias ha hecho que la interfaz de usuario se deba desarrollar en algunos de los lenguajes que pueden ser ejecutados en un navegador web, añadiendo el correspondiente *plugin*, como es el caso de Java, .NET, JavaScript, Flash, Phyton, etc. A modo de ejemplo, en la tabla 2.1 se han especificado los requerimientos necesarios a nivel de *plugin* para la realización de experiencias en los distintos laboratorios ofertados desde el MIT en su proyecto i-Lab, [García-Zubía & Alves 2011].

S/N	iLab	Software used for Lab Client	Required Plug-in
1	Old Op Amp iLab	C#	.Net 2.0
2	New Op Amp iLab	Adobe Flex and C#	.Net 2.0 and Adobe flash
3	Logic Lab	C#	.Net 2.0
4	Advanced Digital Lab	Python and C#	.Net 2.0
5	Robotics iLab	LabVIEW	LabVIEW runtime engine 8.6
6	Control Engineering iLab	LabVIEW	LabVIEW runtime engine 8.6
7	Strength of Materials iLab	Java	Java Runtime Environment
8	Virtual iLabs for DC Experiments	LabVIEW	LabVIEW runtime engine 8.6
9	Communications iLabs	LabVIEW	LabVIEW runtime engine 8.6
10	Emona DATEX Telecoms iLab	Java	Java Runtime Environment

Tabla 2.1: Requerimientos necesarios a nivel de plugin para la realización de experiencias en los distintos laboratorios ofertados desde el MIT en su proyecto i-Lab.

En cualquier caso, la tecnología Java ha sido hasta ahora la más utilizada, [Chang and Hung 2000], [Fjeldly et al. 1999], [Frangu and Chiculita 2010], [Jochheim and Röhrig 1999], [Hahn 2000], [Hsu et al. 2000], [Azorín et al. 2004], [DeWeerth and Knight 1997], [DeWeerth and Knight 1996], [Ko et al. 2002], [Ko et al. 2002a], [Laget et al. 2003], [Nakano et al. 2001], [Perić and Petrović 2000], [Rohrig and Jochheim 1999], en el desarrollo de la interfaz de usuario, quizás porque los *Applet* fueron una de las primeras opciones que permitían ejecutar código en un navegador web. Este lenguaje de desarrollo tiene como principal atractivo ser de uso abierto y con un gran número de librerías científicas y de comunicaciones que facilitan la implementación de soluciones. Como inconveniente, requiere un alto nivel de conocimientos de programación para el desarrollo de aplicaciones, aunque para tratar de minimizar este inconveniente existe la herramienta, *Easy Java Simulations* (EJS), [Farias et al. 2010], [Dormido et al. 2007], creada por el profesor de Matemáticas de la Universidad de Murcia Francisco Esquembre. Esta herramienta está desarrollada íntegramente en el lenguaje Java y permite crear fácilmente simulaciones matemáticas de fenómenos físicos complejos orientadas al uso en el ámbito científico y académico. Esta herramienta constituye por tanto un recurso de desarrollo rápido de interfaces de usuario. De hecho, ha sido utilizada para ello en el proyecto Autom@Lab, permitiendo su integración con aplicaciones de control de experiencias realizadas en *LabView*, *Matlab*, *C*, *Java* etc., [Tuttas and Wagner 2001]. EJS es una herramienta que permite el desarrollo rápido de aplicaciones de interfaz de usuarios en experiencias remotas, ya que pone a disposición de los diseñadores una interfaz gráfica bastante intuitiva desde la que se pueden construir fácilmente formularios, gráficos en 2D y 3D, visualizadores de vídeo y audio, etc. Asimismo, permite añadir a los flujos de video el uso de realidad aumentada. Además EJS dispone de una tecnología de expansión que permite añadir elementos de actuación y adquisición que el usuario puede utilizar simplemente arrastrando y soltando. Esta utilidad permite que EJS pueda ser utilizado también para la realización del programa que controla el experimento, utilidad que está en continua evolución

añadiendo capacidades de acceso a múltiples dispositivos de laboratorio, tanto por parte de los diseñadores de la aplicación, como por los usuarios de la misma.

Casi en el mismo nivel de utilización se encuentran las soluciones propietarias aportadas por *LabView* y *MatLab*. Éstas son comerciales y aportan soluciones completas al problema: proporcionan tanto las herramientas necesarias para diseñar el software que controla el experimento como los *plugin* necesarios para el diseño de la interfaz de usuario. No obstante sus políticas de licencia obligan a realizar un importante desembolso por estas implementaciones.

La tendencia a integrar la interfaz de usuario en los LMS, unida a la evolución que se ha producido en las tecnologías de implementación web como HTML5, AJAX, etc., así como el fuerte crecimiento en el uso de tabletas, está propiciando las soluciones que utilizan *JavaScript* como lenguaje para el desarrollo de la interfaz de usuario, [García-Zubia et al. 2009].

El resto de las soluciones tiene una presencia muy reducida y sus aportaciones son poco significativas, frente a las comentadas.

2.3. Evolución y Análisis de las Clases Prácticas de Laboratorio Remoto (CPLR) según la Tecnología de Implementación del Modelo de Comunicaciones

Las comunicaciones entre la interfaz de usuario y el software que controla el experimento, necesarias para la evolución remota de una CPLR, se realiza en todos los casos siguiendo el modelo cliente-servidor. Sin embargo, cuando se analiza el conjunto de experiencias, se observa que se han utilizado distintas tecnologías de implementación para el desarrollo de dicho modelo. Todas ellas se pueden encuadrar

en una de las cuatro categorías propuestas en [García Zubía and Sáenz Ruiz de Velasco 2005]:

- ✓ Aplicación a medida cliente-servidor.
- ✓ Uso de un servidor web.
- ✓ Uso del servicio de escritorio remoto.
- ✓ Aplicaciones propietarias.

A éstas se les debe añadir la de uso de técnicas de programación distribuida, aparecida recientemente, [SAHARA 2015],[Labshare 2015].

La primera de estas categorías, aplicación a medida cliente-servidor, es quizás la más utilizada hasta ahora y consiste en desarrollar las dos aplicaciones, la del cliente y la del servidor, en algún lenguaje de programación de alto nivel como *C*, *C++*, *Java*, *Python*, *Visual Basic*, etc. Estas dos aplicaciones son desarrolladas a nivel de *sockets* y *serversockets*, lo cual tiene el inconveniente de exigir al diseñador un amplio conocimiento, no solo en el lenguaje de desarrollo, sino también en el desarrollo de aplicaciones de comunicaciones a bajo nivel. Además, ha de tenerse presente que todas las funcionalidades de la aplicación deben ser abordadas por el diseñador y que el mantenimiento de la aplicación obligará en muchos casos a la modificación, tanto de la aplicación del cliente como de la del servidor. Con esta tecnología de implementación, la aplicación servidora debe ejecutarse en un ordenador que debe estar situado en el laboratorio, y la aplicación cliente en el PC del alumno. En muchos casos, cuando el lenguaje de desarrollo lo permite, esta última se integra en una página web añadiendo en un navegador el correspondiente mecanismo (*plugin*). En otros casos, se debe ejecutar como una aplicación del sistema operativo, [Aliane et al. 2007].

La segunda categoría, servidor web, consiste en aprovechar los mecanismos de transferencia de información y de ejecución remota de aplicaciones que disponen los

servidores web, y utilizarlos como aplicación servidora del modelo. En esta arquitectura, la aplicación cliente es constituida por el mismo navegador y la aplicación servidor por el servidor web. Esta estrategia tiene como beneficio inmediato la garantía de estabilidad de la solución, ya que utiliza un protocolo consolidado a nivel de aplicación. Por tanto, esta estructura garantiza el flujo de información entre el usuario y el servidor. Además, la conexión entre el servidor web y los dispositivos del *sistema experimental remoto* se consigue aprovechando la interfaz CGI (*Common Gateway Interface*) de los servidores Web. Esta interfaz permite a los servidores ejecutar aplicaciones que son invocadas desde el navegador, y enviar los resultados a éstos. [Ko et al. 2002] presenta un ejemplo claro de esta arquitectura y otros ejemplos se pueden consultar en [DeWeerth and Knight 1997] y [DeWeerth and Knight 1996].

La tercera categoría, el uso del servicio de escritorio remoto, consiste en aprovechar los servicios de acceso remoto a escritorios que ofertan prácticamente todos los sistemas operativos, con protocolos como RPC, VNC, XDM, etc. La idea básica de esta estrategia es que el que acceda al escritorio del ordenador donde se encuentra el programa o software de control del experimento, pueda activarlo y controlarlo localmente, como cualquier otra aplicación que contenga el PC. La implementación con esta tecnología descarga al programador de todas las funciones de comunicaciones, siendo solo necesarias las funciones de control de evolución de la experiencia. El inconveniente de esta categoría es que nos obliga a definir perfiles de uso para cada usuario de la experiencia de forma que se garantice la seguridad del equipo y la confidencialidad de los resultados obtenidos por cada usuario. También para esta tecnología existe la posibilidad de integrar la ejecución del cliente en el navegador. Ejemplos de usos de esta tecnologías son [Kahoraho and Larrauri 2002]. [Lasky et al. 2005].

La cuarta categoría, el uso de software propietario, consiste en utilizar algunas de las plataformas de software comercial desarrolladas para dar solución completa al problema, no solo a nivel de comunicaciones, sino a nivel de desarrollo de soluciones de

la interfaz de usuario y del software de control del experimento. Las dos plataformas más representativas de esta categoría son *LabView*, [Chen et al. 1999], con su utilidad *Internet Toolkit*; y *MatLab*, [Dixon et al. 2002], con su *Web Server*, que aportan una solución completa para la implementación de soluciones para múltiples escenarios experimentales. Estas dos plataformas presentan grandes ventajas por su facilidad para desarrollar este software, como se ha comentado en el apartado anterior, pero tienen el fuerte inconveniente de su elevado precio, por lo que no profundizaremos en sus características. Existe una solución intermedia que permite reducir el coste de implementación y que consiste en utilizar *MatLab* o *LabView* para implementar el software de control del experimento, resolviendo la interfaz de usuario mediante la herramienta EJS y las comunicaciones entre ambas utilizando *JIL Server*, [JillServer 2014]. El *JIL Server* ha sido desarrollado por la UNED y permite implementar soluciones completas utilizando *MatLab*, *LabView*, *EJS* y el propio *JIL Server* para diseñar el programa de control del experimento y la interfaz de usuario y las comunicaciones, [Vargas et al. 2006], [Vargas et al. 2009].

En las últimas publicaciones se observa que han aparecido un conjunto de soluciones que derivan de la evolución de las dos primeras categorías. Son consecuencia de utilizar las técnicas de diseño de aplicaciones distribuidas en red para resolver el problema de comunicaciones entre la interfaz de usuario y el software de control del experimento. En estas soluciones se introduce una capa denominada *Middleware*, [Deniz et al. 2003], que proporciona una API (*APplication Interface*) que tiene como misión abstraer la complejidad del proceso que subyace en el problema general. La API le oculta al programador la complejidad de uso del sistema de comunicaciones, el sistema operativo, el lenguaje de programación, etc. La introducción de esta API facilita por tanto el diseño de aplicaciones distribuidas en ambientes heterogéneos, reduciendo el tiempo y el esfuerzo de su desarrollo.

Esta tecnología constituye la quinta categoría, técnicas de programación distribuida, y como ejemplo de la misma, se pueden citar, entre otros, CORBA, DCE, RPC, RMI, Web Services, DCOM, COM+, .NET Remoting, JXTA, MQSeries, etc.

La capa *Middleware* proporciona la conectividad entre la interfaz de usuario y el software de control del experimento, ofreciendo un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa distribuida de abstracción de *software*, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red).

Dentro de esta quinta categoría, una de las tecnologías más utilizada es la *Web Service*, [Buiu and Moanta 2008]. Esta tecnología permite que los servidores Web puedan utilizarse para realizar intercambios de datos con el servidor entre aplicaciones distribuidas en la red, usando un conjunto de protocolos que se transporta únicamente a través del puerto 80. En su esquema básico de funcionamiento en el lado del laboratorio, un servidor publica, mediante el formato XML (*eXtensible Markup Language*) y a través del protocolo WSDL (*Web Services Description Language*), el conjunto de servicios que puede ser utilizado por los clientes con un simple navegador. El cliente intercambia información utilizando protocolos con SOAP (*Simple Object Access Protocol*) o utilizando XML-RPC, que es un protocolo de llamada a procedimiento remoto. Los servicios pueden ser demandados principalmente de dos formas: *Remote Procedure Calls* (RPC) y *Service Oriented Architecture* (SOA).

Los servicios *Web Services* con arquitectura de llamada remota a procedimiento RPC permiten la ejecución remota de aplicaciones, realizando las llamadas a los procedimientos mediante el uso del protocolo HTTP para el transporte. Además se codifica la información necesaria para la ejecución con el protocolo XML. En definitiva, un servicio RPC se traduce a nivel de comunicaciones en una solicitud HTTP-POST en la que el cuerpo de la solicitud está realizado en XML. Esta solicitud provoca la

ejecución de un procedimiento en el servidor que genera un resultado que se envía al cliente, codificado en XML. Los parámetros que se envían codificados en XML a los procedimientos pueden ser de cualquier tipo: datos simples, escalares, estructuras, fechas, etc.

Los servicios *Web Services* también se pueden usar para implementar una arquitectura orientada a servicios SOA. En este caso la unidad básica de comunicación es el mensaje, y el servicio está orientado al intercambio de mensajes entre objetos, en lugar de estar orientado a realizar una determinada operación de forma remota. Los servicios son programas con los que podemos interactuar a través de protocolos estándares de intercambio de mensajes, tales como SOAP, XML, XSD, WSDL y otros. A diferencia de lo que sucede en el caso anterior, la información se puede transportar, además de por el protocolo HTTP, por cualquier tipo de protocolo que permita la interacción entre objetos.

En las experiencias publicadas se utilizan múltiples tecnologías de implementación para desarrollar esta arquitectura. Por ejemplo, el proyecto MoCoLab, [Buiu and Moanta 2008], usa ASP.NET 2.0 de *Microsoft*, o el presentado en [Faltin et al. 2002], que utiliza *Java Server Pages* (JSP). El presentado en [Fjeldly et al. 2002] utiliza COM+ y *LabView*, y el presentado en [Jordán et al. 2002] CORBA y *Tomcat*.

Una característica inherente a esta tecnología es la capacidad de distribuir aplicaciones y, en consecuencia, su uso permite crear estructuras distribuidas de CPLR, como por ejemplo el *Distributed Control Lab* (DCL) de Hasso-Plattner-Institute, University of Potsdam, [HPI 2014], y en el que se pueden realizar los siguientes experimentos:

- ✓ Péndulo de Foucault's.
- ✓ Experiencia de levitación magnética.
- ✓ Robot de Lego

- ✓ Maqueta de control de trenes.
- ✓ Maqueta de una línea de control industrial.

Este proyecto, [Rasche et al. 2004], está implementado en base a una arquitectura orientada a servicios SOAP implementada sobre la plataforma de *Microsoft .NET*.

Otro proyecto importante que utiliza esta tecnología con el objetivo de crear una red distribuida de CPLR que pueda ser compartida entre distintas instituciones de enseñanza, es el proyecto *Labshare*, [Labshare 2015]. En este proyecto participa un número importante de universidades e instituciones científicas australianas entre las que se pueden citar *The University of Technology, Sydney (UTS)*, *the Curtin University*, *el Royal Melbourne Institute of Technology*, *the University of South Australia*, y *el Queensland Institute of Technology*. *Labshare* dispone de las siguientes experiencias de acceso remoto en los laboratorios de la UTS:

- ✓ 11 sistemas de desarrollo con microprocesadores.
- ✓ 5 sistemas de control de nivel de tanques.
- ✓ 5 sistemas de programación FPGA.
- ✓ 1 plano inclinado para estudio de los efectos de la gravedad y la fricción.
- ✓ 1 robot móvil.
- ✓ 10 sistemas para el estudio de las deformaciones de estructuras con carga.
- ✓ 5 sistemas para el control neumático mediante PLC.
- ✓ 1 sistema de generación hidroeléctrica.
- ✓ 1 túnel de viento.

La gestión distribuida de experiencias de *Labshare* recae en una compleja aplicación que se denomina SAHARA, que ha sido desarrollada con Java OSGI (*Open Services*

Gateway Initiative), [SAHARA 2015]. OSGI es un *framework* orientado a la programación modular que proporciona a los desarrolladores un entorno orientado a servicios y basado en componentes. El núcleo de SAHARA tiene una arquitectura de servicio que está basada en las interfaces de SOAP, utilizando XML como medio de comunicación de la información. Para garantizar la portabilidad de SAHARA, todos los componentes y lenguajes (*Java*, *Apache*, *PostgreSQL* y *PHP*) de desarrollo utilizados para la implementación del servidor son multiplataforma.

Finalmente, se puede notar que son muchos los proyectos que han ido evolucionando entre estas categorías, como por ejemplo, el proyecto *WebLab* de La Universidad de Deusto, [WebLab 2014]. Comenzó utilizando aplicaciones propietarias cliente-servidor desarrolladas en *Java* y *C*, y ha ido evolucionado a una arquitectura Web e incorporando las sucesivas innovaciones de esta tecnología, como la incorporación de AJAX y su integración en la plataforma LMS *Moodle*, [García-Zubia et al. 2009], la incorporación de una arquitectura orientada a servicios, SOAP, desarrollada en *Python*, etc. En la figura 2.20 se puede observar la evolución que se ha producido en este proyecto, tanto del lado del cliente como del lado del servidor, así como la evolución producida en la estrategia para la implementación de la solución de comunicaciones.

2.4. Conclusiones del Análisis de las Distintas Generaciones de Experiencias Remotas de Laboratorio

Del análisis de la evolución de las soluciones más representativas a lo largo del tiempo estudiadas anteriormente, se deduce que los diseñadores han ido resolviendo el problema enfocando cada vez uno solo de sus aspectos fundamentales. Así, teniendo en cuenta la primera clasificación de las CPLR presentada en este capítulo, en la primera generación el problema fundamental era el control remoto. En la segunda generación se

empiezan a considerar los aspectos didáctico y de gestión. En la tercera se actualiza el enfoque didáctico, y se busca facilitar el diseño.

WebLab	0.1	1.0	2.0	3.0
Device Server Client Proportion				
Connection with devices	RS-232 	SERVER 	USB 	SERVER Network
Client side technology				
Server side technology				
Protocol	proprietary		SOAP	
Does it use HTTP for transporting everything?	No		Yes	
Data protection	-			

Figura 2.20: Evolución de proyecto WebLab de la Universidad de Deusto.

Sin embargo, en ninguna de las experiencias estudiadas se contempla entre sus objetivos la generalización del proceso de experimentación remota, aspecto que conseguiría romper la dependencia con la naturaleza experimental, y facilitaría la creación de herramientas de diseño aplicables a cualquier área de conocimiento.

2. Laboratorios Remotos: Evolución Histórica y Análisis de Características

Realizar esta generalización es el objetivo de esta Tesis. Por otro lado, el análisis realizado ha permitido obtener una arquitectura funcional genérica en la que encuadrar todas las soluciones, y que servirá de referencia para el diseño de la solución general y de su implementación, que se exponen en los capítulos siguientes.

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

Si bien la implementación y uso de *laboratorios virtuales* (LV) como espacio de enseñanza-aprendizaje, es una línea de investigación y práctica extendida desde finales de los años 90, [Andújar y Mateo, 2010], [Ferrero & Piuri, 1988], [Schmid 2000], y, en cierto sentido, puede ser considerada una tecnología más o menos resuelta (con continuas aportaciones y actualizaciones por supuesto), el diseño y uso de *laboratorios remotos*, [Mejías et al. 2014], [Buiu and Moanta 2008], [Barrios 2013], (LR) entraña mucha más dificultad, hasta el punto que es considerado hoy en día un campo de investigación muy activo y sus posibilidades para la enseñanza y el aprendizaje universitario son innumerables, [Dormido 2004]. No obstante, en este capítulo se parte de la consideración de que el avance a nivel tecnológico no ha sido paralelo ni contextualizado a nivel pedagógico y por consiguiente, se quieren abordar nuevas líneas de estudio y aportar reflexiones que profundicen, más que en la configuración tecnológica del LR, en sus aspectos conceptuales y en el potencial didáctico que conlleva su planificación, integración y experimentación.

Como ha quedado establecido en el capítulo 2, independientemente del contenido tecnológico y académico presentes en propuestas publicadas en la literatura sobre LR, todas tienen una característica común: especificidad y concreción. Esto quiere decir que

están orientadas a resolver casos concretos, y no a tratar de forma general la problemática y reto que supone la enseñanza-aprendizaje de las clases prácticas de laboratorio (CPL) desarrolladas de forma remota (CPLR), sin entrar en la naturaleza de las mismas.

Basado en lo anterior, este capítulo pretende aportar una generalización, en forma de modelo teórico, a la solución del problema de facilitar el proceso de enseñanza-aprendizaje referido a las CPLR. Para ello se propone el diseño y desarrollo de un modelo conceptual teórico general, independiente de la *tecnología de información y comunicaciones* (TIC) disponible y de la naturaleza de la CPLR, que permita desarrollar y solucionar el problema de manera formal. El modelo pretende:

- Aportar un marco general de desarrollo para implementar soluciones independientes de cualquier particularidad tecnológica y didáctica de la CPLR.
- Tener un marco didáctico que permita el desarrollo de la CPLR con objetivos de enseñanza-aprendizaje concretos.
- Asistir en el diseño de la CPLR, ya que al manejar los experimentos de forma abstracta (sin concreción en tecnología ni experimento), se facilita el desarrollo de herramientas de diseño que permitan la creación de CPLR sin necesidad de un gran conocimiento de las TIC. Simplemente hay que tenerlo a nivel de usuario de las mismas.
- Poder manejar cualquier *experimento de laboratorio* de forma abstracta, independientemente de la naturaleza del mismo.
- Permitir que se pueda realizar una implementación concreta de la CPLR con las limitaciones técnicas de la realidad tecnológica que cada momento imponga, facilitándose que en sucesivas implementaciones en

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

el tiempo se puedan incorporar los nuevos avances que se produzcan en el ámbito de las TIC.

- Aportar referencias constructivas tanto a nivel de estructura física de la CPLR como funcional y de desarrollo didáctico.

El modelo que se presenta se estructura en base a 5 elementos fundamentales de naturaleza muy diversa y que son los protagonistas en una CPLR: *experimento de laboratorio*, comunicaciones, enseñanza-aprendizaje, gestión del aprendizaje y usuarios (profesores y alumnos), de modo que teniendo en cuenta la interrelación entre estos elementos y el objetivo de que han de funcionar como un sistema holístico (integrador, nunca por partes), surge casi de forma natural la idea de abordar el problema desde el punto de vista de la *Ingeniería de Sistemas* .

Este capítulo se estructura del modo siguiente: en la introducción se establece el esquema relacional de la CPL clásica o presencial, esto es, la habitual en las Escuelas de Ingeniería (y también en otros campos del saber), donde alumnos y profesores comparten espacio físico (laboratorio) para la realización de la clase. A partir de aquí el capítulo desarrolla el modelo necesario para llevar este esquema al laboratorio remoto. Así, en la sección segunda se propone el esquema relacional que representa el proceso enseñanza-aprendizaje de una CPLR. Aquí se establecen las pautas para desvincular el esquema de la CPLR de la naturaleza del sistema experimental, lo cual permite desarrollar un modelo general. Para el desarrollo del mismo se emplea el enfoque de la *Ingeniería de Sistemas* , lo cual se justifica en la sección tercera. En la sección siguiente, la cuarta, se aplica este enfoque a la CPL clásica o presencial y en la sección quinta se extiende a la CPLR. A partir de aquí se está ya en condiciones de afrontar y desarrollar, en la sección sexta, el objetivo fundamental del capítulo, que es el desarrollo del modelo conceptual de una CPLR. Por último, en la sección séptima se aportan las conclusiones principales del capítulo.

3.1. Introducción

Independientemente del planteamiento psicopedagógico que sustente el acto educativo, un buen uso didáctico de las TIC siempre enriquecerá los procesos de enseñanza-aprendizaje, [Nickerson et al, 2007]. En la actualidad, la sociedad de la información exige el abandono de prácticas memorísticas y reproductoras en favor de las metodologías constructivistas centradas en los alumnos y en el aprendizaje autónomo y colaborativo, [European Comission, 2013]. Esto implica un paso más de la mera aplicación de las TIC al proceso educativo, de modo que el interés no puede estar centrado sólo en dotar a los alumnos y profesores de formación para dominar estas tecnologías.

Es en el entorno educativo donde aparece el concepto TIC/TAC (tecnologías del aprendizaje y del conocimiento). Tratan de orientar las tecnologías de la información y la comunicación hacia unos usos más formativos, tanto para el estudiante como para el profesor, con el objetivo de aprender más y mejor. Se trata de incidir especialmente en la metodología y no únicamente en asegurar el dominio de una serie de herramientas tecnológicas. Se trata en definitiva de conocer y de explorar los posibles usos didácticos que las TIC tienen para el aprendizaje y la docencia. Es decir, las TIC/TAC van más allá de aprender meramente a usar las TIC y apuestan por explorar estas herramientas tecnológicas al servicio del aprendizaje y de la adquisición de conocimiento, [Abdulwahed and Nagy, 2011].

Centrando el estudio en las CPL, éstas proporcionan un proceso de enseñanza-aprendizaje específico, complementario a otros tipos de clases (magistrales, de problemas, seminarios, etc.) y probablemente más integrales, al menos en el ámbito de la ingeniería, en el sentido de que permiten acercar el desarrollo y objetivos del aprendizaje lo más posible al entorno técnico/tecnológico de los diferentes campos profesionales de la ingeniería, [Ernest and Kopplint, 1962], [Nikolic et al., 2014].

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

Realmente, junto con las prácticas en empresas, representan el entorno más cercano del alumno a su futuro profesional.

En el desarrollo de las CPL clásicas (presenciales), esto es, con el profesor, el alumno y el experimento coincidentes en el tiempo y en el espacio en el laboratorio físico, la evolución del proceso enseñanza-aprendizaje se produce como consecuencia de la interacción física entre alumno-profesor y alumno-experimento, [Corter et al., 2004. La Figura 3.1 esquematiza las relaciones del proceso de enseñanza-aprendizaje de una CPL clásica en ingeniería (también en otros campos del saber). Nótese las realimentaciones que se producen en el proceso de enseñanza-aprendizaje.

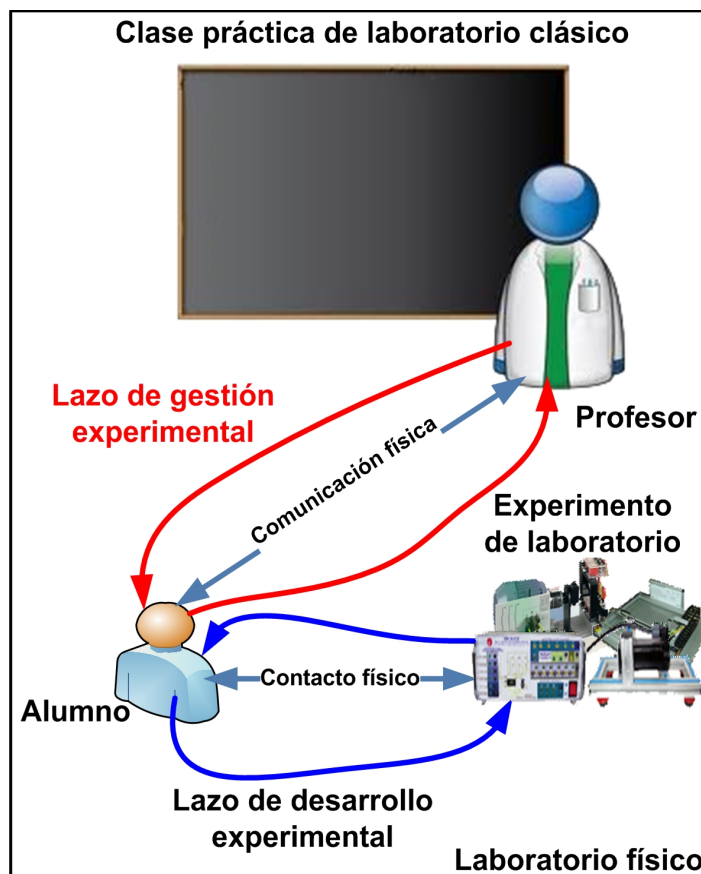


Figura 3.1: Esquema relacional que representa el proceso enseñanza-aprendizaje en una CPL clásica (presencial). Profesor, alumnos y experimento comparten tiempo y espacio, estableciéndose contacto físico entre ellos.

En la Figura 3.1 se aprecian dos lazos de realimentación claramente definidos, el *lazo de gestión experimental* que se establece entre alumno y profesor por comunicación física y el *lazo de desarrollo experimental* que se establece entre el alumno y el *experimento de laboratorio* por acciones/reacciones *in situ*. Nótese que los lazos se representan con línea continua; con ello se quiere enfatizar el hecho de que los lazos se cierran con elementos ubicados en el laboratorio. El de gestión experimental con lenguaje aprendido (hablado, escrito y gestual) entre el profesor y el alumno, y el de desarrollo experimental con acciones físicas sobre el experimento (manuales o no pero *in situ*) y medidas y/o reacciones de éste.

El objetivo fundamental de este capítulo es desarrollar un modelo conceptual general que permita configurar el esquema de la Figura 3.1 para ser impartido de forma remota, configurando lo que hemos dado en denominar CPLR.

3.2. Clase práctica de laboratorio remoto

Parece evidente que el esquema relacional enseñanza-aprendizaje ligado a la realización de una CPL de forma remota no seguiría el esquema clásico de la Figura 3.1. En efecto, ahora hay una desubicación física (y puede que también temporal) del profesor, los alumnos y el/los experimentos, de modo que a los problemas técnicos inherentes a esta forma de realizar las clases prácticas de laboratorio (fundamentalmente en el *lazo de gestión experimental*), hay que añadir los referidos al *lazo de desarrollo experimental*. En la realización de clases prácticas de forma remota (ver Figura 3.2) los lazos de gestión experimental y de desarrollo experimental se cierran a través de un *ente* que permite la desubicación física y temporal del profesor, los alumnos y el/los experimentos, al cual se le denomina *red* en su significado más general en el ámbito de las comunicaciones, [Alexander and Smelser, 2003]. Nótese que a diferencia de la Figura 3.1, los lazos se representan ahora con línea discontinua y se les

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

añade el sobrenombre de *remoto*; con ello se quiere enfatizar el hecho que los lazos han de cerrarse con elementos no ubicados en el laboratorio, con lo cual las comunicaciones y entendimiento entre el profesor, alumno y *experimento de laboratorio*, ahora *remoto*, no es algo trivial ni resuelto *a priori*.



Figura 3.2: Esquema relacional que representa el proceso enseñanza-aprendizaje de una CPL realizada de forma remota. El profesor y los alumnos están desubicados unos de otros en el espacio, y puede que también en el tiempo. Análogamente el alumno y el experimento de laboratorio remoto están desubicados en el espacio.

La Figura 3.2 muestra claramente que el escenario de la CPL ha cambiado de forma sustancial, sin embargo el objetivo debería ser que, aprovechando las ventajas de que el proceso enseñanza-aprendizaje pueda realizarse de forma no presencial, pudieran mantenerse las innatas debidas a la presencia física de alumno, profesor y experimento de las CPL clásico. Si esto se consigue, es decir si el alumno tiene en todo momento la sensación de tener la experiencia en sus manos y, además, percibe la gestión del

aprendizaje por parte del profesor, no sólo no se verá resentido el proceso enseñanza-aprendizaje por realizarse de forma remota, sino que saldrá reforzado.

Como puede apreciarse en la figura 3.2 , el profesor y el alumno están desubicados en el espacio y puede que también en el tiempo, de modo que su nexo de unión es el ente que hemos denominado *red*. Ahora, la interacción del profesor y el alumno a través de la *red* para la gestión del proceso enseñanza-aprendizaje, o *lazo de gestión experimental*, puede ser de forma directa (forma que podríamos denominar *a mano*, donde el profesor es el gestor directo del aprendizaje) y/o de forma indirecta, desde un nivel más elevado, a través de un elemento (que es denominado *sistema de gestión del aprendizaje en red*) que realiza la función de interfaz de gestión de enseñanza-aprendizaje entre profesor y alumno de forma desubicada (espacio y tiempo); administrando, distribuyendo y controlando bajo demanda las distintas relaciones, fases y momentos formativos, [Kent and Cardt, 1961], [ERNST, 1983]. Por supuesto, los *lazos de gestión experimental remoto directo e indirecto* pueden convivir y actuar de forma conjunta durante el desarrollo de la CPLR. En cualquier caso y para ambos lazos el gestor último del aprendizaje es el profesor.

Teniendo en cuenta que el objetivo del capítulo es diseñar un modelo teórico conceptual que recoja el proceso enseñanza-aprendizaje en el escenario de las CPLR, y que se pretende que este modelo sea general y válido para todo el ámbito de la ingeniería (y por extensión también en otros campos del saber), es fundamental hacerlo independiente de la naturaleza del experimento de laboratorio. Así, con respeto del *lazo de desarrollo experimental remoto* de la Figura 3.2, si se consigue desvincular éste de la naturaleza del *experimento de laboratorio remoto*, se dispondrá de un modelo general para la configuración y realización de CPLR, aplicable en cualquier campo del conocimiento (mecánica, automática, física, etc.), lo cual daría carácter general al modelo. Para conseguir esto, las actuaciones sobre el *experimento de laboratorio remoto* y la medida de

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

sus reacciones (*lazo de desarrollo experimental remoto*) deberían compartir un formato común sea cual fuere la naturaleza del experimento, Esto es, las señales de actuación utilizadas por el usuario para el desarrollo del experimento y las señales recibidas por el usuario para verificar la evolución del mismo, no han de depender de la naturaleza del experimento de laboratorio. Esto posibilitaría manejar las particularidades de cada experimento abstraídos de su naturaleza, lo cual permitiría establecer un esquema general de relaciones en las clases prácticas de laboratorio.

La solución que se propone a nivel conceptual y teórico para llegar a un modelo general válido para un *experimento de laboratorio remoto* de cualquier naturaleza, es utilizar las posibilidades de gestión virtual y remota que proporcionan las TIC, de modo que cualquier acción, reacción u otra característica física que sea relevante para la experimentación de cualquier naturaleza, pueda ser procesada a distancia, de forma abstracta desde un recurso computacional. Esto permitiría que profesor, alumnos y experimentos pudieran interactuar usando un mismo lenguaje, con pasarelas e interfaces entre ellos completamente transparentes para todos, de modo que lo verdaderamente importante sea el proceso enseñanza-aprendizaje de la CPL, no los recursos tecnológicos que permiten que esto pueda hacerse de forma remota.

Para entender lo que persigue el modelo a desarrollar, piénsese en los ejemplos siguientes de *lazo de desarrollo experimental* de una CPL presencial de cualquier ámbito de la ingeniería u otro campo del saber (ver Figura 3.1): pulsar un botón y observar una respuesta, regular una válvula y obtener una medida, regar una planta y medir la humedad en el suelo, actuar sobre una probeta y medir su deformación, etc. Si se consigue desvincular el *lazo de desarrollo experimental* de la naturaleza del experimento, de modo que pueda ser generado y cerrado a través de la *red* mediante periféricos de un computador (teclado, monitor, ratón, manipuladores 3D, puertos de E/S, etc.) en el lado del usuario y en el del experimento, podríamos llegar a la situación de la Figura 3.2.

Nótese que la desvinculación del *lazo de desarrollo experimental remoto* de la naturaleza del experimento se realizará, en la práctica, mediante las posibilidades tecnológicas que aporten las TIC en cada momento (esto se concretará en los capítulos siguientes). Esto es, las TIC deben permitir conectar, a través de la *red*, al usuario remoto con el mundo físico del *experimento de laboratorio remoto*, de modo que ambos puedan interactuar en un lenguaje común para que el lazo se cierre. Esto quiere decir que para dar carácter general a la solución del problema, se propone que el modelo no esté asociado a ninguna tecnología de implementación concreta, y en este sentido se ha de entender que lo que se va a denominar a partir de ahora *pasarela TIC* es un concepto teórico, esto es, no supone asumir ninguna tecnología concreta, sino simplemente que en cada momento se aplicará/n la/s disponible/s.

En virtud de lo anterior, es conveniente remarcar que en esta investigación:

El concepto de *pasarela TIC* circunscribe, en líneas generales, el problema de acceso a laboratorios remotos a un problema de comunicaciones, de modo que una vez resuelto, usuarios (alumnos y profesores) y *experimento de laboratorio remoto* (y sus elementos individualizados) pueden compatibilizar caminos, en el sentido de que todas las señales que se emiten o reciben son interpretables por todos.

Entre las muchas aportaciones que puede proporcionar el concepto de *pasarela TIC* a la realización de CPLR, podrían resaltarse dos a nivel de la explotación didáctica:

- ✓ *multiplexación experimental*. Ante múltiples usuarios disponibles para realizar una CPLR, la *pasarela TIC* permite que tanto profesores como alumnos puedan gestionar la CPLR y establecer el *lazo de desarrollo experimental remoto* a voluntad, de entre todos los posibles usuarios del grupo de conexiones intervinientes. Véase Figura 3.3.

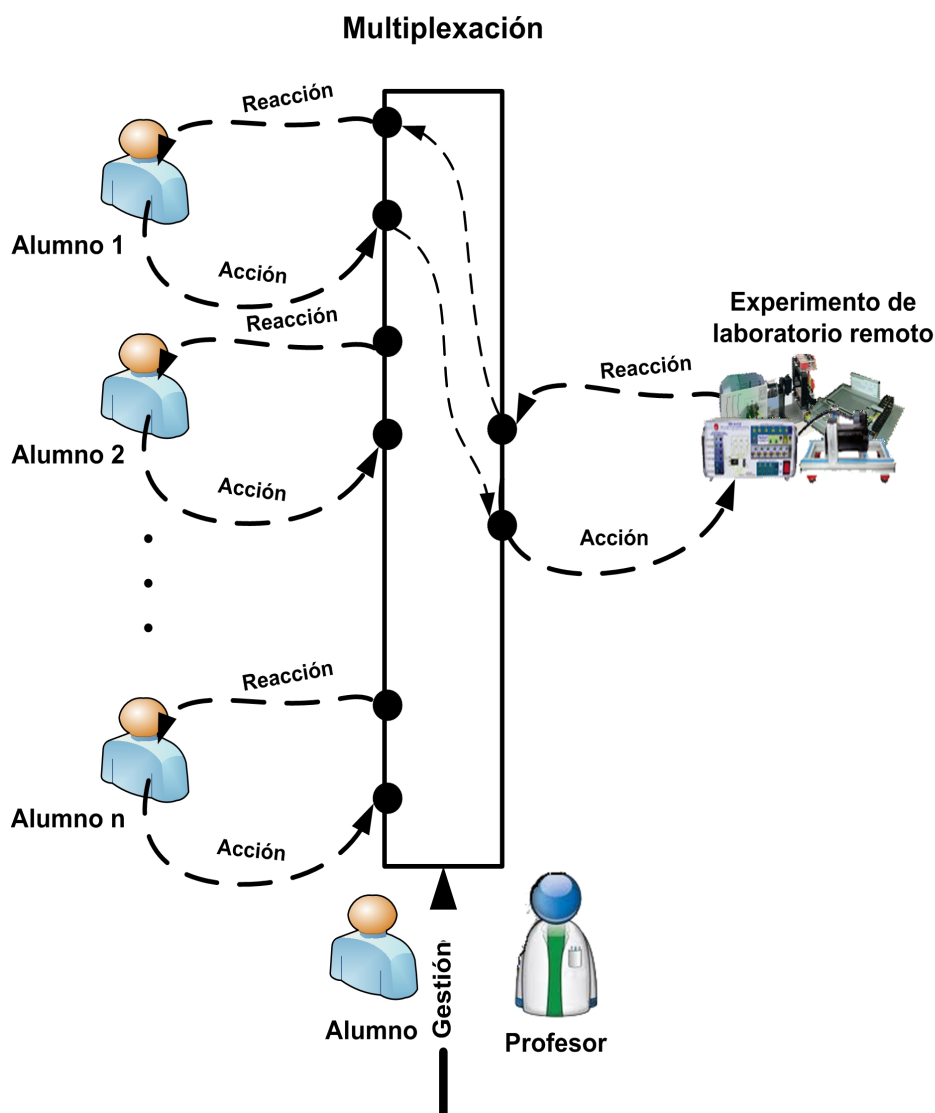


Figura 3.3: Multiplexación experimental

- ✓ *diseño y configuración experimental remota*. Los usuarios (profesores y alumnos) pueden diseñar y configurar de forma remota las CPL. Es más, este concepto permite incluso configurar *experimentos de laboratorio remoto* mediante la conexión en red de recursos desubicados en el espacio y que puedan ser gestionados de forma abstracta como un todo (experimento), lo cual se muestra en la Figura 3.4. Este concepto, que se denomina *experimento de laboratorio remoto distribuido*, está basado en la compatibilidad que la *pasarela TIC* proporciona a cada elemento utilizado para configurar el

experimento de laboratorio remoto distribuido.

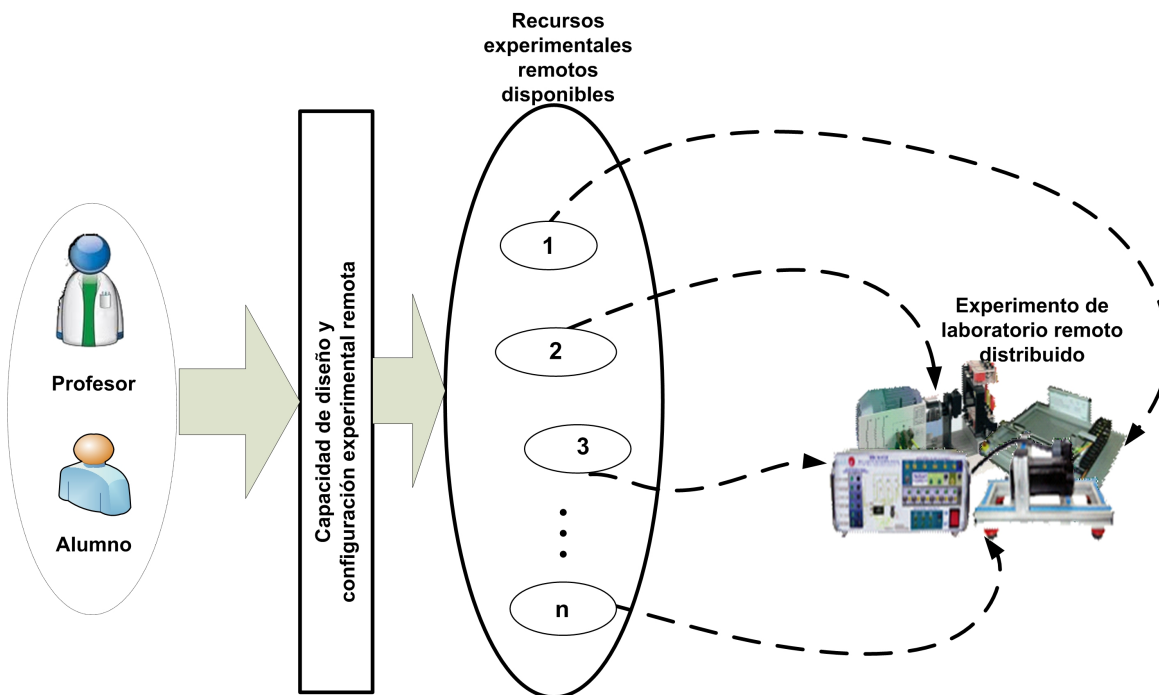


Figura 3.4: Diseño y configuración experimental remota

3.3. Enfoque de la Ingeniería de Sistemas

Como se ha comentado al inicio del capítulo, los todavía escasos ejemplos de laboratorios remotos operativos y visibles a nivel internacional, han sido diseñados e implementados desde el punto de vista de una experimentación de naturaleza concreta: laboratorio remoto de manipuladores robóticos, laboratorio remoto de control de procesos, etc., [Buitrago et al., 2011], [Guimarães et al., 2003], [Jara et al., 2011], [Fabregas et al., 2011], [Calvo et al., 2009], [Andria et al., 2007], [Indrusiak et al., 2007], [Vargas et al., 2011]. Estos laboratorios suelen apoyarse en las facilidades de

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

comunicación de las herramientas software de análisis y diseño habituales en su ámbito del conocimiento (Matlab, LabView, C++, Java, etc.), para resolver de forma integral toda la problemática a nivel didáctico, de acceso y experimental. El resultado final obtenido es el de laboratorios muy específicos, fuertemente influenciados por sus definiciones experimentales, y por tanto difícilmente generalizables ni aplicables a cualquier otra área experimental diferente de aquella para la que fueron concebidos. Esto trae como consecuencia la dificultad para crear herramientas de diseño genéricas que permitan asistir a profesores no interesados en la tecnología que soporta la arquitectura e implementación física del laboratorio remoto, sino en su explotación docente, de forma que puedan llevar a cabo CPLR sin necesidad de poseer un alto conocimiento en la/s tecnología/s que soporta/n el laboratorio remoto: comunicaciones, electrónica de adquisición y procesamiento de señales, programación, etc.

Podría decirse que el problema de diseño e implementación de un laboratorio remoto se ha abordado hasta ahora en la literatura como un problema *total*, donde para llegar a actuar sobre un elemento del experimento y medir su reacción, ha habido que resolver previamente una cantidad enorme de problemas técnicos a todos los niveles. Cada experiencia de laboratorio remoto presente en la literatura ha sido realizada prácticamente *ad hoc*, con un enfoque en la naturaleza del experimento, y por tanto difícilmente generalizable. Lógicamente, si un laboratorio remoto ha de servir para más que presentar una realización concreta, esto es, para tener un uso cotidiano por profesores y alumnos con un objetivo docente, éstos han de ser abstraídos de la problemática técnica que subyace en el funcionamiento del laboratorio. La razón es obvia, ya que los intereses de la mayoría de los usuarios (profesores y alumnos) estarán centrados en la realización de la CPL desde el punto de vista del proceso enseñanza-aprendizaje de la materia objeto de la misma, no en los medios tecnológicos que soportan que ésta pueda hacerse de forma remota. Esto es especialmente importante en el caso del profesor, ya que si se ve obligado a realizar esfuerzos alejados de la materia

sobre la que trate la CPL y de su propia formación científico/técnica (piénsese por ejemplo en un experimento de ensayo de materiales donde al profesor, para poder implementar el laboratorio remoto, se le exijan desarrollos de electrónica, instrumentación, programación, control, comunicaciones, etc.), se sentirá disuadido, probablemente, del uso del laboratorio remoto.

No obstante lo anterior, si el modelo para la CPLR que desarrolla esta investigación se centrara únicamente en intentar desvincular la solución de comunicaciones (a todos los niveles) de la naturaleza del experimento (es decir, que bajo el mismo modelo se pueda abordar cualquier experiencia concreta, usada de una forma didáctica concreta y que permita un nivel de interacción con el usuario concreto), se estaría adoptando una visión parcial del problema. Esto es, aunque el modelo podría permitir realizar clases prácticas de áreas y temáticas muy diferentes bajo el mismo concepto de laboratorio remoto, al no tener en cuenta en la creación del modelo los condicionantes del binomio enseñanza-aprendizaje, su utilización quedaría limitada a aspectos sobre todo tecnológicos, pero quedarían muy restringidas sus posibilidades docentes. En consecuencia, para que el modelo sea realmente general y efectivo para el proceso enseñanza-aprendizaje, éste debe contemplar como un todo los elementos intervinientes en el proceso y sus relaciones. Éstos son fundamentalmente cinco y de naturaleza muy diferente: experimentos de laboratorio, comunicaciones, enseñanza-aprendizaje, gestión del aprendizaje y usuarios (profesores y alumnos), de modo que teniendo en cuenta la correspondencia entre estos elementos y el objetivo de que han de funcionar como un sistema integral con una interrelación completa entre ellos (en el momento en que se rompe el todo se pierde alguna de sus propiedades fundamentales), se propone abordar el problema desde el punto de vista de la *Ingeniería de Sistemas* [Smartta and Ferreira, 2013], [Gilberta et al., 2014]. Esto es, el modelo se generaliza a partir de aglutinar sistemas, sus componentes y la relación entre ellos (señales que los conectan). Las características que pueden ser aplicables a cualquier sistema son:

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

- ✓ La interrelación de sus componentes (relación entre las partes y el todo).
- ✓ Los sistemas están ordenados en una jerarquía.
- ✓ Las partes de un sistema no son iguales al todo.
- ✓ Los límites de los sistemas son artificiales.
- ✓ Los sistemas pueden ser abiertos o cerrados – según la influencia con su entorno.
- ✓ Cada sistema tiene entradas, procesos, salidas y lazos de realimentación.
- ✓ Las fuerzas dentro de un sistema tienden a ser contrarias entre ellas para mantener el equilibrio.
- ✓ La Entropía como medida de la cantidad de información y ruido.

En virtud de lo anterior, resulta obvio que los modelos relacionales de las Figuras 3.1 y 3.2 pueden ser considerados relaciones entre sistemas, aunque de naturaleza muy diversa. En las secciones que siguen se afronta el diseño del esquema relacional de la Figura 3.2 bajo el prisma de la *Ingeniería de Sistemas* : el ámbito didáctico del problema, su solución y posterior aplicabilidad general. Se vuelve a hacer hincapié en el ámbito didáctico, ya que la solución al diseño del modelo objeto de este capítulo, ha de estar basada en el binomio enseñanza-aprendizaje de los procesos involucrados en el modelo, no en el problema de automática, química, física, ..., del experimento; o en el problema de realización del *experimento de laboratorio* de forma remota. Consecuentemente, la solución al problema planteado debe contemplar un enfoque necesariamente didáctico.

Por último es conveniente remarcar que en esta investigación:

El concepto *sistema enseñanza-aprendizaje* se entiende como una globalidad o conjunto de todos los elementos físicos o no involucrados en él. El *sistema enseñanza-aprendizaje* es el medio, no necesariamente físico, en el que se produce el *proceso enseñanza-aprendizaje*, cuya dinámica está dada por los estados (pasos) por los que discurre.

3.4. Proceso de enseñanza-aprendizaje de una clase práctica de laboratorio

En este apartado pretendemos modelar con un enfoque de la *Ingeniería de Sistemas*, el desarrollo del proceso de enseñanza-aprendizaje ligado a una CPL clásica (presencial) con el objetivo de poder conseguir, en un paso posterior, su reproducción remota. En la introducción de este capítulo (ver Figura 3.1), se planteó que la evolución de la CPL se produce como consecuencia de la interacción de tres entidades (profesor, alumno y experimento de laboratorio) a través de dos lazos de realimentación, *gestión experimental* (relación profesor-alumno) y *desarrollo experimental* (relación alumno-experimento de laboratorio), [Corter et al., 2004].

Desde el punto de vista de la *Ingeniería de Sistemas* la evolución del proceso enseñanza-aprendizaje de una CPL se plantea como la interacción de 3 sistemas (profesor, alumno y experimento de laboratorio) mediante lazos de realimentación, donde el objetivo de control por parte del profesor, como gestor del aprendizaje, es que la salida del proceso (grado de aprendizaje del alumno) alcance un nivel de referencia (estipulado en el programa de la asignatura), el cual puede ser evaluado por el profesor durante el desarrollo de la clase (en línea) o a posteriori (fuera de línea) mediante una prueba.

Siguiendo con el esquema de la Figura 3.1, la interacción que se produce a través de los dos lazos de realimentación modificará, respecto del proceso enseñanza-aprendizaje, el estado de los tres sistemas, de modo que hay que controlar la evolución de dicho proceso para que el alumno pueda alcanzar el aprendizaje de referencia como consecuencia del aumento de sus habilidades, competencias y conocimientos. El modelo a desarrollar debería contemplar tanto la evolución de los estados de los sistemas como la naturaleza y el funcionamiento de los lazos de realimentación del proceso. Nótese que la problemática que se pretende resolver es mucho más amplia que la mera

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

realización de la CPL; la solución de este problema es sólo una parte (aunque necesaria) del proceso. Abordar este enfoque es fundamental para entender que lo que persigue el modelo a desarrollar no es simplemente realizar una CPL de forma remota (resolver un problema tecnológico), sino proponer una nueva forma de enseñanza-aprendizaje basada en la posibilidad, cierta hoy en día, de realizar CPLR.

Se comenzará por analizar el desarrollo de una CPL presencial genérica, tal como se imparte hoy en día en cualquier Escuela de Ingeniería o Facultad de Ciencias. Esto permitirá establecer los cambios de estado en los sistemas intervinientes respecto de la salida de interés (aprendizaje de los alumnos), y el funcionamiento de las relaciones entre los sistemas mediante los lazos que los interconectan, así como la naturaleza del intercambio de acciones e información que se producen durante el desarrollo del proceso (enseñanza-aprendizaje).

El modelo parte de dos premisas fundamentales:

- 1 No se pretende modelar cuantitativamente la dinámica del proceso, sino representar la dinámica de la CPL y sus implicaciones de carácter didáctico, con el objetivo de abordar, en un paso posterior, su reproducción de forma remota. Es decir, se busca un modelo cualitativo que refleje y facilite el entendimiento del proceso, esto es, del binomio enseñanza-aprendizaje asociado a una CPL.
- 2 Se tomará como escenario de partida una CPL genérica, cuyo diseño y correcta realización garanticen el cumplimiento de los objetivos de aprendizaje tanto a nivel de desarrollo de habilidades experimentales, como de competencias y conocimiento propuestos en el diseño curricular de los estudios en los que esté encuadrada.

El modelo perseguido será construido de forma gráfica, incluyendo en su estadio final todos los sistemas y elementos intervinientes en el proceso. Se comenzará por una formulación sencilla para ir ampliando paso a paso conforme se vaya profundizando

en el modelo para conseguir su función: modelar con el grado de detalle necesario el proceso enseñanza-aprendizaje de una CPL.

Una CPL presencial (clásica) requerirá, en general, de una fase previa de diseño y preparación, para luego ser organizada e impartida, también de forma general, en tres partes: introducción, desarrollo y conclusiones, [Sell and Rüütman, 2014]. Ha de tenerse en cuenta que en este tipo de clases se requiere del alumno una actividad y participación especialmente intensas. Los 3 sistemas que más interrelacionan en el proceso de enseñanza-aprendizaje durante la CPL son el profesor, el alumno y el experimento de laboratorio. La Figura 3.5 esquematiza lo anterior como una primera aproximación, sin entrar aún en un modelo de relación del tipo de la Figura 3.1.

Atendiendo a la Figura 3.5, el profesor, en la fase de diseño y preparación, configura el experimento sobre el cual va a realizar el alumno la CPL (dependiendo del curso y lo avanzado de la asignatura en la carrera, el alumno puede participar también en la configuración del experimento como una actividad más de la CPL). Para ello se basa en los requisitos docentes de la asignatura a la que corresponde la CPL. Para aclarar ideas, en el modelo actual en Europa de asignaturas adaptadas al *Espacio Europeo de Educación Superior* (EEES), donde la unidad de medida es el *crédito ECTS*, cada asignatura dispone de una *guía docente* donde se especifican sus datos generales y específicos, [European Commission, 2013]. Como se puede apreciar en la Figura 3.5 el profesor crea también, en la fase de diseño y preparación, la *base de conocimiento* necesaria tanto para la realización adecuada de la CPL (guía) como para que ésta cumpla los objetivos docentes requeridos (en el modelo educativo del EEES serían los reflejados en la *guía docente*). Como se verá en esta sección, dependiendo del grado de formación del alumno, éste puede contribuir a la creación y enriquecimiento de la *base de conocimiento* de la CPL.

En la fase de introducción se inicia propiamente la realización de la CPL. El profesor contextualiza (prerrequisitos, requisitos, recomendaciones, etc.) al alumno para la

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

realización de la CPL, así como le proporciona cuanta información sea precisa para la correcta realización de la misma (una parte de la *base de conocimiento*). Esta información ha sido generada por el profesor en la *fase de diseño y preparación* de la CPL. En la *fase de introducción* de la CPL el alumno es sobretodo un sujeto pasivo que recibe información.

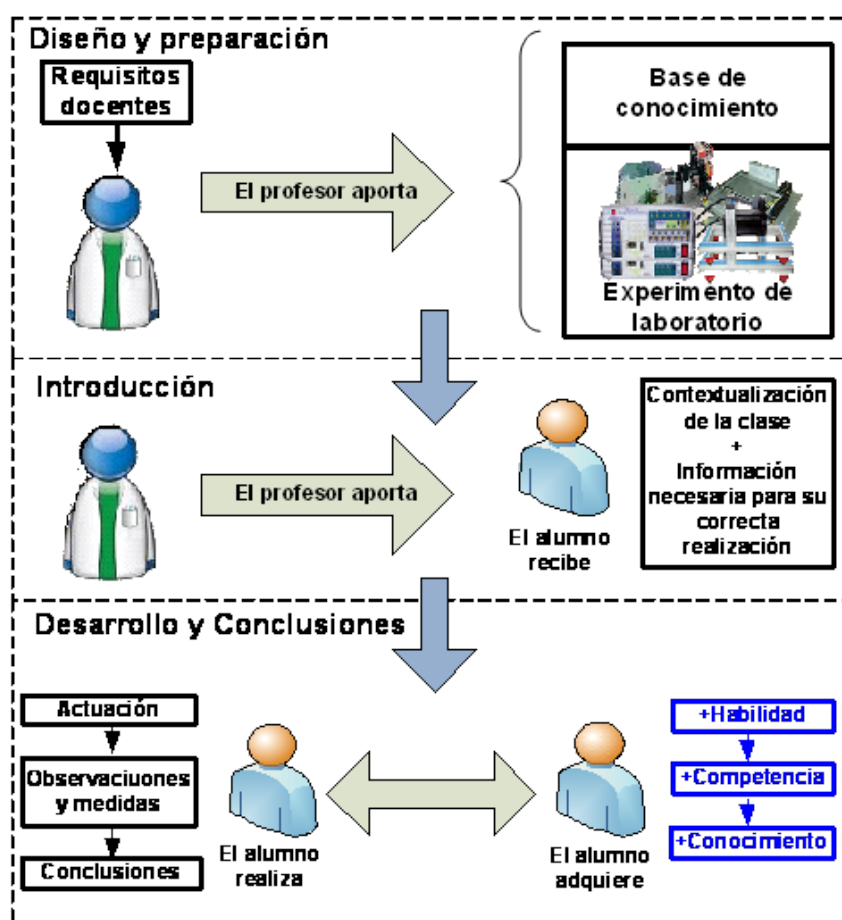


Figura 3.5: Esquema general de las fases y sistemas intervinientes en una CPL presencial.

La *fase de desarrollo y conclusiones* se presenta de forma conjunta (ver Figura 3.5). La razón es que la casuística de las CPL es muy diversa, de modo que la fase de conclusiones después de un análisis de los resultados no siempre va al final de la clase; a menudo es preciso tener que sacar conclusiones sobre resultados parciales que van siendo obtenidos durante la realización de la CPL, de modo que el sentido de esas

conclusiones puede variar su discurrir previsto. Durante esta fase el alumno es el verdadero protagonista de la CPL, comportándose como un sujeto activo, quedando el profesor en un plano de supervisión o incluso, dependiendo de la tipología de la clase, desapareciendo prácticamente del escenario, [European Comission, 2013].

En la *fase de desarrollo y conclusiones*, el alumno actúa sobre el experimento a diferentes niveles (operándolo, reconfigurándolo, etc.), realiza observaciones, obtiene medidas (datos) y analiza los resultados (de índole muy diversa dependiendo de la tipología de la CPL) concluyendo sobre ellos.

Si la CPL ha sido diseñada y configurada de forma adecuada por el profesor, la correcta realización de ésta por parte del alumno le debe proporcionar las habilidades, competencias y conocimiento previstos en la *guía docente* de la asignatura correspondiente.

En lo que sigue vamos a implementar el esquema de la Figura 3.5 en forma de estructuras de relación, siguiendo la representación de la Figura 3.1 bajo el enfoque de la *Ingeniería de Sistemas*. El objetivo es encontrar un modelo dinámico que represente cualitativamente el proceso enseñanza-aprendizaje en una CPL clásica (presencial).

La metodología clásica considera las CPL como un proceso de enseñanza-aprendizaje diseñado y administrado por el profesor (él es el único gestor del aprendizaje) y desarrollado de manera activa por los alumnos. El profesor, generador de la *base de conocimiento* necesaria para la realización de la CPL (ver Figura 3.5), suministra al alumno la parte conveniente de ésta, tanto para la realización adecuada de la clase (guía) como para que ésta cumpla los objetivos docentes requeridos: test, baterías de preguntas, modelo de informe o memoria; en general un mecanismo de control y prueba que permita evaluar el cumplimiento de los objetivos de aprendizaje de la CPL. El modelo relacional del diseño, gestión y desarrollo de la CPL presencial se muestra en la Figura 3.6. Nótese que la acción del profesor, a través del *lazo de gestión experimental*

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

directo, equilibra la posible carencia de conocimientos del alumno, así como las dificultades que pudiera tener en la ejecución de la CPL. En cuanto al *lazo de desarrollo experimental* el alumno va adquiriendo más habilidad, competencia y conocimiento (a partir de una condición inicial al comienzo de la CPL) conforme va obteniendo medidas del *experimento de laboratorio* (acorde a sus actuaciones sobre él) y va interaccionando con la *base de conocimiento* a través del *lazo de gestión experimental indirecto*. En base a lo expresado en la sección 3.3. queda claro pues el enfoque de la *Ingeniería de Sistemas* que hemos utilizado a la hora de representar y entender el diseño, gestión y desarrollo de la CPL presencial.

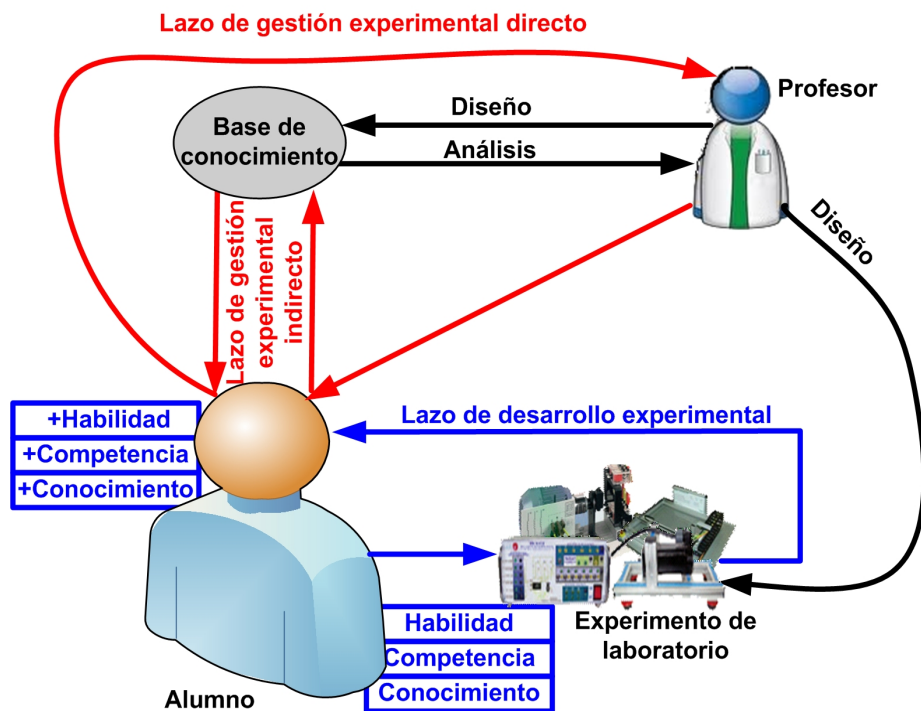


Figura 3.6: Modelo relacional del diseño, gestión y desarrollo de una CPL presencial conducida por el profesor y ejecutada por el alumno.

Desde el punto de vista del alumno, el modelo relacional mostrado en la Figura 3.6 puede ser en algunos casos muy rígido, en el sentido de que la CPL está guiada en todo momento por el profesor, dejando muy poco margen de maniobra al alumno. Este modelo parece apropiado en los primeros cursos o en las primeras CPL de una

asignatura, cuando el alumno no tiene aún los suficientes conocimientos previos y/o su familiarización con el laboratorio es escasa, [Magin and Kanapathipillai, 2000]. Sin embargo, cuando debido a su formación el alumno alcanza una cierta madurez en sus conocimientos, parece más apropiado utilizar esquemas para el desarrollo de la CPL que impliquen más al alumno, esto es, que éste disponga de mayor autonomía en la clase hasta ser el verdadero gestor de la misma, quedando el profesor en un segundo plano, ejerciendo una labor más de orientación, [Magin and Kanapathipillai, 2000]. Abundando en este asunto, el concepto en el *EEES* de crédito *ECTS* trata de buscar una participación lo más activa posible del alumno en su propio proceso de aprendizaje, cambiando la cultura de estudiantes *receptores de información* a la de estudiantes *inquisidores del saber y del conocimiento*, [European Comission, 2013].

Bajo este enfoque, el *lazo de gestión experimental directo* de la Figura 3.6 dejará de tener sentido, puesto que la clase ya no estará conducida por el profesor, es decir, él no será el gestor del aprendizaje, o al menos el gestor principal, quedando su labor más a nivel orientativo. Además, el alumno (o grupo de ellos), basándose en sus habilidades, competencias y conocimientos previos, así como en las orientaciones del profesor, será también el diseñador de la *base de conocimiento* necesaria para la realización de la CPL, así como del propio experimento de laboratorio. Este modelo relacional de CPL diseñada, gestionada y desarrollada por el alumno se muestra en la Figura 3.7.

Llegados a este punto es importante hacer hincapié en que este trabajo de investigación no pretende realizar un estudio de las diferentes metodologías de diseño e impartición de CPL presenciales buscando la más óptima, sino analizar de forma general las más usuales para poder llegar a implementarlas en una CPLR. En cualquier caso, el modelo de este tipo de clases que desarrolla esta investigación pretende ser lo suficientemente general como para dar cabida a las diferentes casuísticas.

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

En relación con lo anterior, si se obvia la gestión del aprendizaje y el diseño de la CPL, y se atiende sólo a su realización por parte del alumno, los modelos relacionales de las Figuras 3.6 y 3.7 son, como cabía esperar, análogos al representado en la Figura 3.8, la cuál podría ser considerada como el esquema general del desarrollo de una CPL presencial.

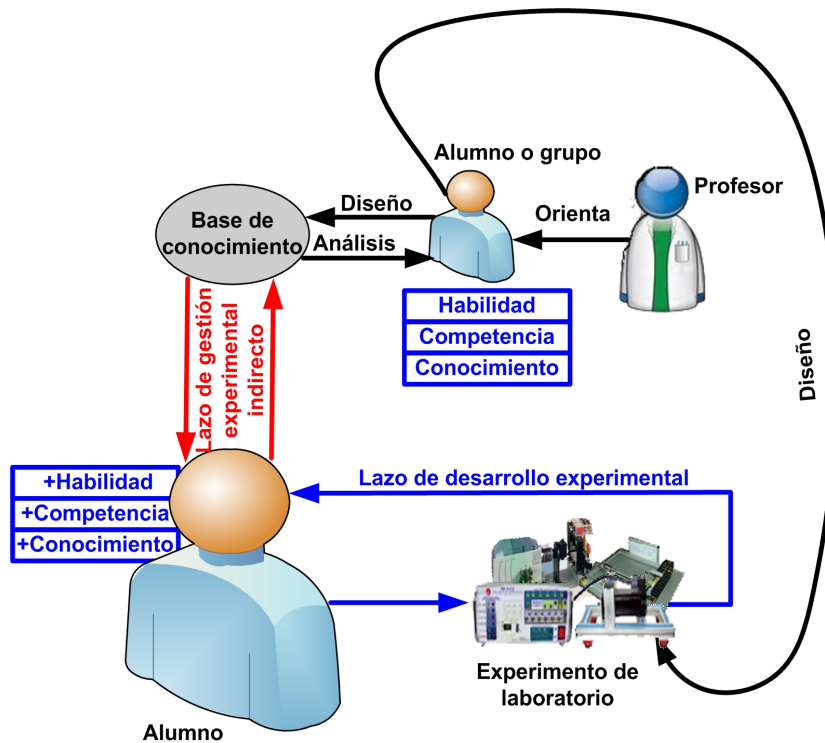


Figura 3.7: Modelo relacional del diseño, gestión y desarrollo de una CPL presencial conducida y ejecutada por el alumno.

En esta sección, el análisis realizado desde el punto de vista de la *Ingeniería de Sistemas* del proceso enseñanza-aprendizaje ligado a la realización de una CPL presencial, ha permitido encontrar los modelos relacionales (Figuras 3.6, 3.7 y 3.8) que reflejan la dinámica del proceso. Esto ha posibilitado identificar los sistemas intervinientes en el proceso así como las variables de interés y su naturaleza.

En lo que sigue en el capítulo se tratará de extrapolar los modelos de la CPL presencial para su realización como CPLR.

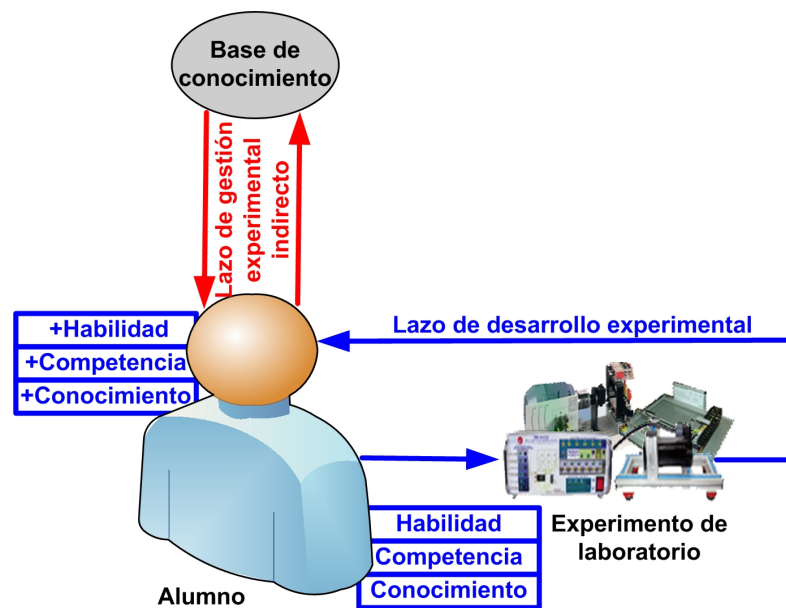


Figura 3.8: Modelo relacional general del desarrollo de una CPL presencial.

3.5. Proceso de enseñanza-aprendizaje de una clase práctica de laboratorio remoto

Ahora se pretende realizar los esquemas de las Figuras 3.6 y 3.7 según el modelo de la Figura 3.2, esto es, para el caso de que la CPL y el proceso enseñanza-aprendizaje ligado a ella sean realizados de forma remota.

En virtud de lo estudiado en el capítulo, la *red* por un lado y el concepto de *pasarela TIC* por el otro, permiten soslayar las limitaciones de comunicación y contacto físico de los lazos representados en la Figura 3.6 para ser implementados de forma remota, lo cual se muestra en la Figura 3.9. Si el esquema a trasladar para su implementación remota es el de la Figura 3.7, la realización sería la de la Figura 3.10. Nótese en esta Figura como el *gestor del aprendizaje* es el propio alumno. El escenario de *red*, y de comunicaciones en ella, más general posible para el diseño, gestión y desarrollo de una CPLR es el propio de una *red* pública de área geográfica lo más extensa posible, lo que

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

en la actualidad equivale a Internet. La red pública, soporte de la CPLR, debe permitir la conexión bidireccional entre dos nodos cualesquiera de dicha red en formato punto a punto y punto a multipunto. Esta característica es tan general que podemos afirmar que la cumplen todas las redes actuales y también que será cumplida por las futuras que tengan la denominación de públicas.

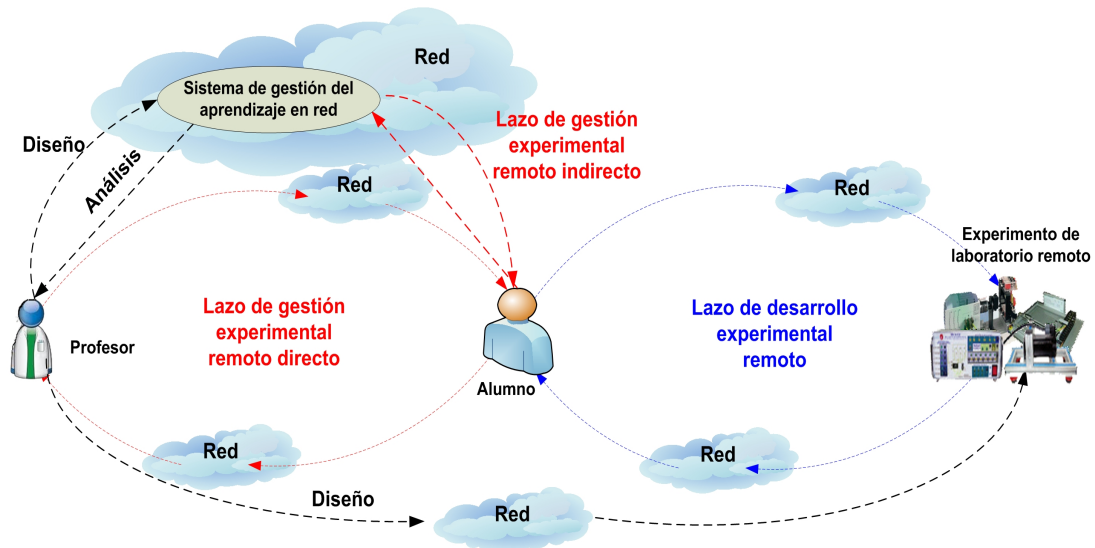


Figura 3.9: Modelo relacional del diseño, gestión y desarrollo de una CPLR guiada por el profesor y experimentada por el alumno.

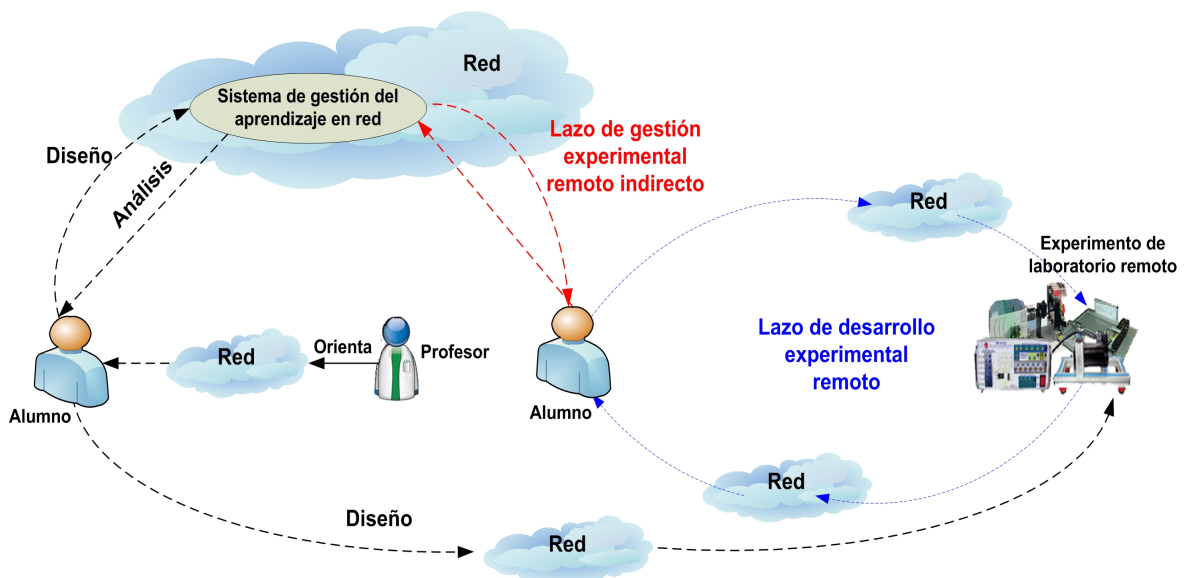


Figura 3.10: Modelo relacional del diseño, gestión y desarrollo de una CPLR guiada y experimentada por el alumno.

Los modelos de las Figuras 3.9 y 3.10 requieren de un *sistema de comunicaciones* que permita cerrar los diferentes lazos a través del ente que hemos denominado *red*. Este *sistema de comunicaciones* debe tener un comportamiento *transparente* para el usuario (profesor y alumno), de modo que el concepto de *pasarela TIC* en el *lazo de desarrollo experimental remoto* permita trasladar acciones y reacciones a través del *sistema enseñanza-aprendizaje* (implícito en los modelos relacionales de las Figuras 3.9 y 3.10) de la CPLR, independientemente de su naturaleza y de los elementos que conformen el *experimento de laboratorio remoto*. Sin embargo, debemos tener presente que la *red* sólo puede transportar estructuras de datos (*paquetes de datos*) de extremo a extremo, [Tanenbaum and Wetherall, 2010], con lo cual el concepto de *pasarela TIC* implicará, a bajo nivel, el disponer de una interfaz que traduzca las acciones a través de la *red* sobre los experimentos y las reacciones de éstos en *paquetes de datos*. A esta interfaz, que en realidad es un subsistema dentro del sistema de comunicaciones, la hemos denominado en nuestro modelo *subsistema de convergencia del sistema de comunicaciones*, y permite resolver el *lazo de desarrollo experimental remoto* (ver Figura 3.11). Desde el punto de vista operativo, esta convergencia implica una capa o nivel más a añadir a los inherentes al sistema de comunicaciones. En la situación tecnológica actual, el sistema de comunicaciones estaría regido por la norma al uso de interconexión de sistemas abiertos (ISO/IEC 7498-1), también llamado *modelo OSI (open system interconnection)*, [Tanenbaum and Wetherall, 2010].

En virtud de lo anterior y a nivel de sistemas, el esquema de la CPLR de las Figuras 3.9 y 3.10, teniendo en cuenta sólo su gestión y desarrollo (no la fase de diseño), podría quedar representado como en la Figura 3.11. Nótese que el *experimento de laboratorio remoto* ha sido considerado como un sistema más (*sistema experimental remoto*). Nótese también que ahora no es necesario hacer distinción en si el gestor del aprendizaje es el profesor, el alumno o ambos y, en consecuencia, tampoco si el *lazo de gestión experimental remoto* es directo o indirecto. En la Figura 3.11 se representan todos los usuarios

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

participantes en la CPLR integrados en el *sistema enseñanza-aprendizaje*, entendiéndose éste como el sistema que involucra a las personas, tanto a nivel de gestores del aprendizaje como de receptores del mismo. Respecto al *lazo de gestión experimental remoto*, éste queda resuelto mediante el *sistema de comunicaciones*. Ello se debe a que se asume que el *sistema de comunicaciones* dispone de las funcionalidades necesarias para hacer transparente a los usuarios (alumnos, profesores y gestores del aprendizaje) el *lazo de gestión experimental remoto* (que realiza el proceso de gestión del aprendizaje asociado a la CPLR en desarrollo) mediante el *sistema de gestión del aprendizaje en red*. Con la tecnología actual, éste es un problema ya resuelto a través de los LMS, de uso común en los entornos educativos [WebCT, 2014], [Blackboard, 2014], [Brightspace, 2014], [Exact-learning, 2014], [Moodle, 2014], [Dokeos, 2014], [Claroline, 2014], [Sakaiproject, 2014]. Son herramientas TIC desarrolladas como consecuencia de la evolución natural de los *gestores de contenidos* en el campo de la educación; están implementadas sobre un *gestor de bases de datos* que facilita el tratamiento de cualquier *estructura de datos*, incluso si es *multimedia*. Las capacidades básicas aportadas por los LMS actuales son:

- 1 *Gestión de acceso*: controlar quién puede acceder a los datos y qué puede hacer con ellos.
- 2 *Gestión de contenidos*: base de conocimiento y cualquier otra información adicional que sea necesaria (en nuestro caso para la realización, control y evaluación de la CPLR).
- 3 *Presentación de la información*: control de la oferta de ésta a los usuarios según criterios (didácticos por ejemplo) definidos por el gestor.
- 4 *Comunicación grupal*: e-mail, chats, foros, conexiones *on line*, etc.
- 5 *Intercambio de información*: todos los elementos del grupo pueden intercambiar cualquier tipo de información *multimedia*.

No obstante lo anterior, la estructura y relaciones del modelo que desarrolla esta investigación no depende de las funcionalidades del *sistema de gestión del aprendizaje*; en consecuencia, la evolución tecnológica de éste no hará sino permitir implementaciones prácticas del modelo con más prestaciones concretas, aunque bajo el mismo marco conceptual.

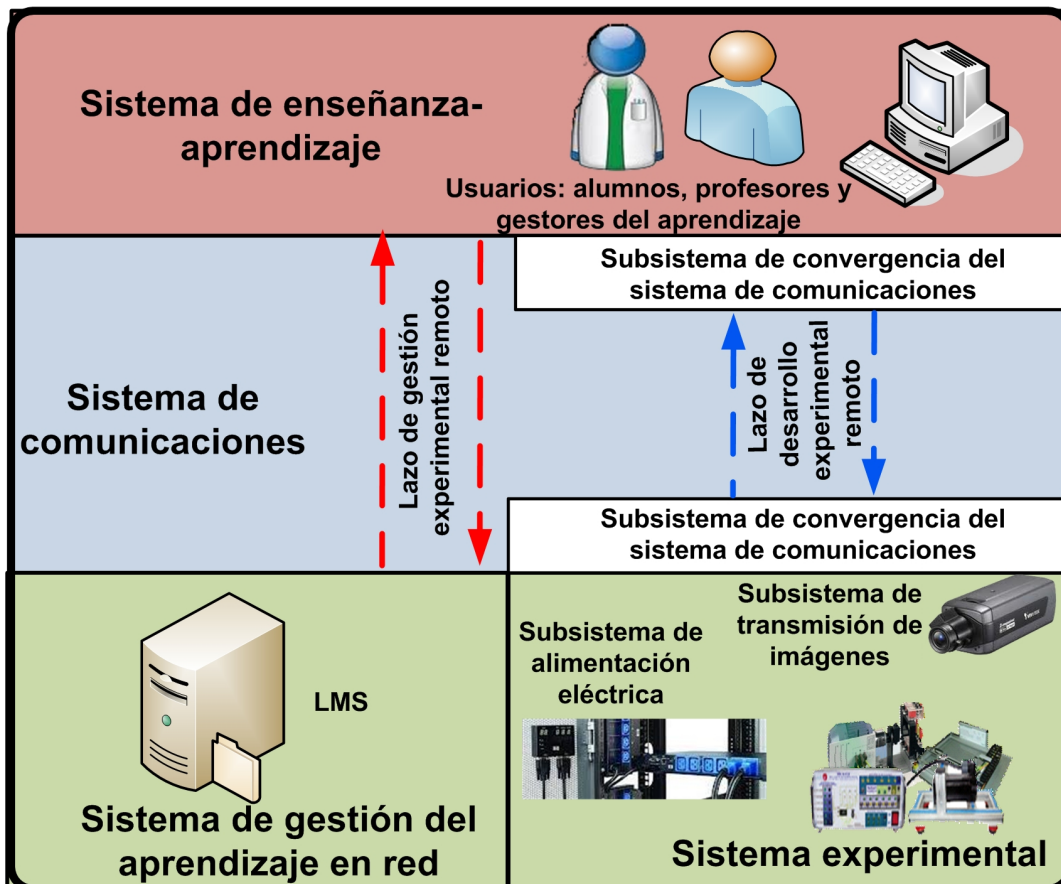


Figura 3.11: Sistemas participantes en la gestión y desarrollo de una CPLR con convergencia del sistema de comunicaciones.

Todos los usuarios (profesores, alumnos y gestores del aprendizaje) acceden a los sistemas participantes en la Figura 3.11 mediante un computador u otra interfaz dotada de las aplicaciones adecuadas. Así por ejemplo, en el computador del usuario que está realizando la CPLR debe haber varias aplicaciones o una integral que proporcione:

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

- 1 Conectividad e interacción con el *sistema de gestión del aprendizaje en red* y, a través de éste o no, conectividad e interacción con el/los gestor/es del aprendizaje, lo cual permitirá cerrar de forma directa y/o indirecta el *lazo de gestión experimental remoto* (ver Figuras 3.9 y 3.10).
- 2 Conectividad e interacción con el *sistema experimental remoto*, de modo que el usuario pueda tener un control total (gracias a la *pasarela TIC* proporcionada por el *subsistema de convergencia del sistema de comunicaciones*) del *sistema experimental remoto*. Esto permite cerrar el *lazo de desarrollo experimental remoto*. En la frontera con el *sistema de enseñanza-aprendizaje*, el *subsistema de convergencia del sistema de comunicaciones* traducirá las acciones producidas en uno de los dispositivos de interfaz disponibles para el usuario (teclado, pantallas táctiles, ratón, señaladores 3D, etc.), a una *estructura de datos* de red (paquetes), que se enviarán, normalmente utilizando un ordenador, al *sistema experimental remoto* a través del *sistema de comunicaciones*. Asimismo, las reacciones del *sistema experimental remoto* se traducirán, también mediante el *subsistema de convergencia del sistema de comunicaciones* situado en esa frontera, en *estructura de datos* que representen las reacciones del *sistema experimental remoto*. Estas reacciones serán mostradas al usuario por dispositivos de representación de la información.

El esquema de la Figura 3.11 y las consideraciones anteriores implican que todos los sistemas participantes en la gestión y desarrollo de la CPLR hablan el mismo lenguaje y se entienden entre ellos, esto es, desde o hacia todos los sistemas la información que circula es en forma de *paquetes de datos*. De este modo y desde un punto de vista tecnológico, la gestión y desarrollo de una CPLR se circunscribe a un problema en el ámbito de las comunicaciones. Esto, al menos en parte, es también cierto para el proceso enseñanza-aprendizaje ligado a la CPLR.

En el esquema de la Figura 3.11 se asume (no están representados de forma explícita) que en un nivel más bajo a los computadores de acceso, gestión y desarrollo de la CPLR se encuentran los *dispositivos de red*. Éstos son necesarios para gestionar por la *red* los *paquetes de datos* que se generen y permitir la interconexión de todos los sistemas participantes.

Como complemento de refuerzo en la percepción y desarrollo de la CPLR, el *sistema experimental* de la Figura 3.11 ha sido completado con un *subsistema de transmisión multimedia*, con objeto de conseguir una mayor sensación de realidad física del alumno en la realización de la CPLR, ya que en todo momento le permite disponer de imagen y sonido en tiempo real del experimento y sus evoluciones.

La configuración final del *sistema experimental* se complementa con el *subsistema de alimentación eléctrica*. Su razón de ser es la siguiente: a efectos operativos, el concepto de laboratorio remoto implica su disponibilidad continua (24 h al día durante 7 días a la semana) para los usuarios sin necesidad de estar atendido físicamente por personal de forma continua. Esto implica que o bien se dejan permanentemente alimentados todos los componentes del *sistema experimental* (incluida la iluminación necesaria para que el usuario pueda recibir imágenes en tiempo real del experimento y sus evoluciones), lo cual genera un gasto energético y desgaste de componentes innecesario cuando no hay uso, o se conecta la alimentación precisa sólo cuando se vaya a realizar un experimento. Éste es el cometido del *subsistema de alimentación eléctrica*, el de suministrar energía eléctrica a todos los elementos participantes del *sistema experimental* sólo cuando se vaya a realizar el experimento, de modo que cuando éste acabe debe cortar de forma automática el suministro eléctrico al *sistema experimental*.

Para finalizar esta sección es preciso tener en cuenta que si bien la *red* pública resuelve el problema de comunicaciones en el nivel y extensión más elevado (esquema de la Figura 3.11), queda aún por resolver un problema más cercano al *sistema*

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

experimental remoto y a sus usuarios. Desde la *red* pública se producirán accesos múltiples a los sistemas experimentales que permitan configurar las diferentes CPLR. Sin embargo y de forma general, los *sistemas experimentales remotos* se encontrarán en una o varias redes locales de acceso controlado (centros educativos, empresas, etc.), con lo cual será necesario disponer de un sistema que enlace la *red* pública y las diferentes redes locales, realizando una función de gestión de accesos que contemple aspectos como quién usa el recurso, cómo lo usa, con qué finalidad, etc. Esto, junto a la necesidad de disponer de subsistemas específicos para el desarrollo de la CPLR, así como de interacción con ella, será tratado en la sección siguiente y servirá para completar el modelo buscado.

3.6. Modelo conceptual general de una clase práctica de laboratorio remoto

Como se ha comentado en la sección anterior, el *sistema de comunicaciones* de la Figura 3.11 ha de resolver el problema del enlace entre la *red* pública y las redes privadas involucradas en la CPLR. Es decir, es preciso que el *sistema de comunicaciones* tenga la capacidad de gestionar dos o más redes que a su vez estén enlazadas entre sí: una que siga siendo de acceso público y otra/s que tenga/n acceso controlado y aloje/n los sistemas experimentales o parte de éstos. Al sistema que ha de resolver esta necesidad de comunicaciones lo hemos denominado SARLAB (sistema de acceso remoto a laboratorio) y será desarrollado en los capítulos que siguen de esta memoria.

En nuestro modelo, SARLAB es el *túnel* de comunicaciones que debe implementar todas las funcionalidades necesarias para que el *lazo de desarrollo experimental remoto* se pueda cerrar de forma transparente para el usuario, en cualquier escenario de redes públicas/privadas e independientemente de los controles de acceso y gestión que éstas impongan. La ubicación de SARLAB en el modelo conceptual general se muestra en la

Figura 3.12. Nótese que SARLAB conecta el escenario experimental (donde se ubica el *sistema experimental remoto*) con el *sistema de enseñanza-aprendizaje* (donde se ubica el usuario) y, a través de éste, con el *sistema de gestión del aprendizaje en red*.



Figura 3.12: Esquema del modelo conceptual general de una CPLR.

SARLAB ha de permitir multisesiones, ya que debe facilitar el acceso a todos los sistemas participantes en la CPLR, cada uno pudiendo estar alojado en su propia *red* local, con lo cual, teniendo en cuenta la generalidad del modelo buscado, éste debe propiciar, mediante SARLAB, que cada CPLR pueda tener su correspondiente estructura de configuración independiente.

Llegados a este punto, es importante remarcar que (ver Figura 3.12):

SARLAB, junto con las dos *capas de convergencia del sistema de comunicaciones*, permite manejar de manera transparente los datos en los escenarios de enseñanza-aprendizaje y del laboratorio remoto, los cuales están desubicados en el espacio y puede que también en el tiempo. Esto permite que el manejo *del sistema experimental remoto* por parte del usuario, tanto en la preparación como en el desarrollo de la CPLR, quede definido por un conjunto de conexiones programables por el usuario y la gestión de los datos generados, independientemente de la naturaleza del *sistema experimental* y de los datos transportados por la *red*, lo cual hace que el modelo desarrollado sea general y de aplicación a una CPL de cualquier rama de la ingeniería y también de otros campos del saber. En el lado del usuario (*sistema enseñanza-aprendizaje*) SARLAB realiza las labores de gestión del desarrollo de la CPLR, a partir de la definición que se ha realizado de ésta y, a través del usuario, en coordinación con el *sistema de gestión del aprendizaje en red*.

En las CPL presenciales de hoy en día, es de uso común la programación de alto o bajo nivel, generación de estímulos, control, etc., desde un ordenador u otro tipo de terminal informático hacia el *sistema experimental*, así como la adecuación y procesado de los datos recibidos desde él para su análisis y representación mediante gráficos, modelos, etc. En definitiva, aunque la CPL sea presencial, es usual que gran parte de ella la realice el usuario desde un ordenador sobre el que controla el *sistema experimental*. El traslado de estas capacidades al ámbito de la CPLR, se representa en nuestro modelo con un subsistema en el lado del usuario, el cual hemos dado en denominar *subsistema de análisis, diseño y programación experimental* (Figura 3.12). Nótese que en general este subsistema será una aplicación informática genérica (como MATLAB por ejemplo) o específica, proporcionada por el/los fabricante/s de los elementos que componen el *sistema experimental* y que puede estar dotada de algún

hardware en la ubicación del *sistema experimental*. Por lo común, el *subsistema de análisis, diseño y programación experimental* será específico para cada asignatura e incluso para cada CPLR, y deberá proporcionar, con carácter general, las funcionalidades siguientes:

- ✓ La generación de estímulos desde el usuario y hacia el *sistema experimental remoto*, tanto desde modelos del mismo como a partir de *instrumentos virtuales* u otras aplicaciones.
- ✓ La interpretación, en el lado del usuario, de las respuestas del *sistema experimental remoto* mediante su representación adecuada.
- ✓ La capacidad de programación desde el lado del usuario de los elementos que configuran el *sistema experimental remoto*.

Es importante destacar que estas tres características y otras pueden residir en una única aplicación informática o en varias, y que en un experimento concreto puede que sólo sea/n necesaria/s alguna/s de estas características.

Respecto de la sensación de control del experimento por parte del usuario, y de su percepción de estar tratando con un sistema real, aunque sea *a distancia*, las CPLR necesitan una capacidad de interacción con el *sistema experimental remoto* mucho más exigente que la CPL presencial, donde el contacto físico del usuario con el *sistema experimental* aporta un valor intrínseco. Así, en el nivel menos exigente, esta interacción puede reducirse a una simple caja de diálogo o un formulario a rellenar desde la pantalla de un ordenador; sin embargo, el nivel de exigencia de la interacción puede ir en aumento: mediante un manipulador o una completa interfaz gráfica de usuario, que puede incluir sonido real o simulado, y la visión física y/o simulada de la evolución del proceso experimental, e incluso en un nivel superior la sensación de contacto con el *sistema experimental remoto* puede estar *aumentada*, mediante el uso de técnicas orientadas a enriquecer la percepción humana, como es el caso de la *realidad aumentada*

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

(RA). Hoy en día existen ya en la literatura diferentes ejemplos del uso de técnicas de RA para enriquecer el sistema de interacción en las CPLR, [Andújar, Mejías y Márquez, 2011], [Mejías et al. 2014], [Mejías y Andújar, 2011], [Henderson and Feiner 2011], [Marner et al. 2014].

En virtud de lo anterior, la especificidad y nivel de exigencia de interacción de la CPLR, necesita estar soportada por un sistema específico. Éste queda reflejado en nuestro modelo con la denominación de *subsistema de interacción* (Figura 3.12), el cual, situado en el lado del usuario, que es quién lo maneja, es indispensable en las CPLR, ya que puede considerarse la ventana de contacto entre el usuario y el *sistema experimental remoto*.

Ahora, la Figura 3.12 representa el cumplimiento del objetivo fundamental de este capítulo, esto es, la traslación al ámbito remoto del modelo de la Figura 3.1. Ésta representaba el modelo relacional del proceso enseñanza-aprendizaje en una CPL clásica (presencial), donde profesor (como gestor del aprendizaje), alumnos y experimento compartían tiempo y espacio, estableciéndose contacto físico entre ellos. En la CPLR (Figura 3.12), los *lazos de gestión y desarrollo experimental* se implementan ahora de forma remota, y el/los gestor/es del aprendizaje, trabajan desubicados espacial y temporalmente del *sistema experimental* y del alumnado que realiza la CPLR, a través de un *sistema de gestión del aprendizaje en red* (LMS).

Como resultado natural de la solución del desarrollo de la CPLR, podemos concluir que ésta se llevará a cabo en el *laboratorio remoto* (LR), entendido éste como el laboratorio físico o real con capacidad de poder ser teleoperado a distancia, de modo que se pueda realizar experimentación sobre él con prestaciones análogas a como se haría de forma presencial.

A modo de conclusión de esta sección remarcamos que:

El modelo conceptual general desarrollado considera que la evolución del proceso de enseñanza-aprendizaje en la CPLR se genera como consecuencia del tráfico de información a través del *sistema de comunicaciones*, asumiendo que el flujo controlado y adecuadamente gestionado de información a través de la *red* es el elemento clave para el desarrollo de la CPLR y la consecución de sus objetivos docentes.

3.7. Conclusiones

En síntesis, podemos concluir que la CPLR desarrollada en este capítulo, implementada con las prestaciones del concepto genérico de LR, es una capacidad educativa que crece paralela e íntimamente enlazada con el propio desarrollo de la sociedad del conocimiento. El desarrollo de las tecnologías TIC/TAC propicia la posibilidad de afrontar un salto cuantitativo y cualitativo de magnitud aún difícil de calibrar en nuestras formas de enseñar y aprender. Ésta, podríamos decir, revolución educativa aún en sus inicios, pivota o gira en torno a los ejes o principios básicos siguientes:

- ✓ La autonomía y protagonismo del estudiante sobre su proceso formativo.
- ✓ Las capacidades actuales y futuras de las tecnologías TIC/TAC.
- ✓ El aprendizaje experiencial haciendo uso remoto de los recursos y herramientas disponibles.
- ✓ La comunicación e interacción que se desarrolla en el acto educativo.
- ✓ La combinación de la educación formal y del aprendizaje informal o incluso *a la carta*.

3. Modelo Conceptual General de una Clase Práctica de Laboratorio Remoto

No se trata simplemente de añadir tecnología como escenario o espacio a los procesos de enseñanza-aprendizaje tradicionales. El modelo que propone esta investigación, basado en las capacidades tecnológicas presentes y futuras, supone, a nuestro juicio, un cambio educativo mucho más profundo, ya que propone reinventar los fenómenos y procesos educativos convirtiéndolos en una nueva forma de acceder a la formación basada en un mayor protagonismo del sujeto que aprende (personalización). Ahora éste tiene que cumplimentar tareas o acciones de búsqueda, análisis y elaboración de información, e incluso puede llegar a convertirse en gestor de su propio aprendizaje.

Con las posibilidades que ofrecen los LR, los estudiantes pueden trabajar desde cualquier sitio, no existen barreras espaciales ni temporales. Además, el bagaje de experiencia que se produce a través de las CPLR otorga a los alumnos una serie de competencias que les servirán no solo en su formación inicial si no también en su futuro académico y profesional.

Ha de quedar claro que el concepto de CPLR que se ha modelado en este capítulo pretende satisfacer sus requerimientos técnicos y también los educativos, de modo que desde las soluciones tecnológicas se debe poder responder a interrogantes tales como: ¿por qué y cómo se justifica un determinado proceso de enseñanza-aprendizaje?, ¿qué variables didácticas y organizativas intervienen?, ¿qué objetivos se pretenden conseguir?, ¿quiénes son los destinatarios?, ¿cómo serán las estrategias que se activarán para alcanzar los objetivos propuestos? y ¿qué secuencias temporales se determinarán para la asimilación de los aprendizajes? Consideramos que estas cuestiones son básicas y nucleares para afrontar cualquier propuesta formativa de un *plan de estudio* utilizando el LR como un elemento vehicular.

La Universidad ha de insertarse en la sociedad tecnológica que nos invade, teniendo en cuenta que gracias al avance de nuevas tendencias didácticas y de las posibilidades

tecnológicas disponibles, estamos en disposición de desarrollar un sistema de formación remota que servirá para mejorar el proceso de aprendizaje de aquellas personas que decidan aprender a cualquier hora, en cualquier momento y en cualquier lugar, sin las rigideces y condicionantes (muchas veces disuasorios para el estudiante) de los sistemas presenciales tradicionales. Los recursos remotos constituyen una nueva forma de enseñar y aprender, y abren posibilidades didácticas insospechadas, más aún en el ámbito de las CPL, las cuales han sido habitualmente un escollo para el desarrollo de la formación reglada remota en el ámbito de la ingeniería (y también en otros campos del saber), ya que la simple virtualización de la CPL ni resuelve su desarrollo, ni menos aún aprovecha todas sus posibilidades formativas.

Ha de remarcarse por último que el modelo de CPLR que se ha desarrollado en este capítulo es completamente general, independiente de la tecnología TIC/TAC disponible (el avance de estas tecnologías no hará sino mejorar las prestaciones y capacidades del modelo) y de la naturaleza de la CPLR. En virtud de ello, nos estamos enfrentando a una nueva forma de enseñar y aprender bajo la cual hay que reconsiderar gran parte de los conceptos didácticos, organizativos, metodológicos y evaluativos que estructuran la comunicación didáctica presencial actual. Y esto significa, a nuestro modo de ver, un avance cualitativo de extraordinarias consecuencias para la formación universitaria y que en estos momentos sólo alcanzamos a vislumbrar y predecir.

4. Modelo General Aplicado de un Laboratorio Remoto

En el capítulo anterior se llevó a cabo una generalización, en forma de modelo teórico, de la solución del problema de facilitar el proceso de enseñanza-aprendizaje referido a las clases prácticas de laboratorio realizadas de forma remota (CPLR). Para ello se propuso un modelo conceptual teórico general, independiente de las tecnologías disponibles de información y comunicaciones así como de aprendizaje y conocimiento (TIC/TAC), y de la naturaleza misma de la CPLR. El modelo presentado permite desarrollar y solucionar el problema de manera formal. El enfoque sistémico propuesto en ese modelo obliga a manejar el problema como un todo indivisible, y por tanto a tener presente a lo largo de su desarrollo todos los procesos representados en la figura 3.12. Esto es, el proceso de enseñanza aprendizaje, el de comunicaciones, el experimental y el de gestión del aprendizaje. Además, hay que tener en cuenta que todos ellos evolucionan a través de las interacciones que se producen mediante los dos lazos representados en la figura 3.12, el de gestión experimental remota y el de desarrollo experimental remoto.

Es importante remarcar que este modelo es completo, en el sentido de que no solo pretende satisfacer los requerimientos técnicos de la CPLR, sino también los educativos. En dicho modelo la gestión del aprendizaje en red se encomienda a un sistema ya resuelto hoy en día: el LMS (su evolución tecnológica natural con el tiempo no hará sino

aportar más prestaciones), mientras que la gestión del experimento requirió el desarrollo de diferentes subsistemas, la mayoría de ellos conectados al sistema de comunicaciones, los cuales hacen de interfaz entre los usuarios y el sistema experimental. La introducción del LMS como sistema que permite implementar el *lazo de gestión experimental remoto* delega en él todas las funciones derivadas del aprendizaje a través de la red. Esto obliga a integrar como módulos del LMS todas las funcionalidades necesarias para gestionar las interacciones que se producen a través del lazo de desarrollo experimental. Así, las CPLR podrán realizarse como una actividad más a gestionar remotamente desde este sistema.

De lo dicho en el párrafo anterior, el problema se circunscribe a resolver el *lazo de desarrollo experimental remoto*. Esto incluye sus interfaces, tanto con el sistema de enseñanza-aprendizaje como con el sistema experimental remoto.

Por otro lado, el concepto clave para el desarrollo realizado ha sido el de pasarela TIC implementada en el *subsistema de convergencia del sistema de comunicaciones* del modelo. Este concepto permite manejar cualquier sistema experimental remoto de forma abstracta, independientemente de la naturaleza del mismo, ya que unifica el lenguaje de comunicación entre usuarios y sistemas experimentales de laboratorios remotos genéricos, lo cual permite desvincular el lazo de desarrollo experimental remoto de la configuración y objeto del sistema experimental. Esto se lleva en el modelo hasta su funcionalidad total, haciendo que el usuario se pueda desenvolver desde el sistema de enseñanza-aprendizaje hasta el sistema experimental y viceversa de forma transparente para él, en el sentido de no tener que reparar en el problema de entendimiento y gestión de señales de cada experimento de laboratorio particular, ya que ello lo resuelve el sistema de comunicaciones implementado en el modelo.

En este capítulo se va a evolucionar hacia una visión más práctica del modelo conceptual teórico del capítulo 3 (ver Figura 3.12) hasta llevarlo a un modelo

tecnológico genérico más próximo a la implementación final que será abordada en los siguientes capítulos. Esto es, se pretende la evolución del modelo conceptual sin entrar en ninguna tecnología concreta TIC/TAC, que será en cada momento una de las disponibles. Para ello se irán introduciendo los elementos especializados necesarios que han de permitir la implementación práctica del modelo.

4.1. Introducción

La estructura y configuración del sistema de comunicaciones que permite conectar de forma transparente para el usuario el sistema de enseñanza-aprendizaje y el *sistema experimental*, independientemente de la naturaleza de éste, ha de ser tal que proporcione al usuario el control completo del *sistema experimental* desde su interfaz (ver Figura 3.12). Para ello son necesarias diferentes tareas que se irán desarrollando en este capítulo de forma secuencial.

Del análisis de las soluciones realizado en el capítulo 2 desde el punto de vista constructivo, se deduce que la primera tarea a abordar es la medida desde el sistema de enseñanza-aprendizaje de las variables de interés del sistema experimental, lo cual implica, en el nivel más bajo, la utilización de los sensores necesarios en éste. Esto es, cualquier variable de interés ha de ser convertida, mediante el sensor apropiado, en una magnitud eléctrica. También, para poder controlar el sistema experimental actuando sobre él, es preciso que cualquier señal de actuación que se genere en el sistema enseñanza-aprendizaje se convierta en una acción concreta sobre el sistema experimental. Esto ha de realizarlo un actuador que pueda ser gobernado de forma remota desde el sistema enseñanza-aprendizaje. Así pues, en el nivel más bajo (unido a los elementos que componen el sistema experimental) y con carácter general, se han de conectar al experimento de laboratorio tantos sensores y actuadores como señales haya que medir y acciones haya que realizar.

En la mayoría de los casos, los sensores proporcionarán a su salida una señal analógica, esto es, de valor continuo, la cual no puede ser procesada directamente por la computadora del usuario, que precisa una señal digital codificada en lógica binaria. Así pues, la salida analógica del sensor ha de ser convertida a una señal digital que represente el valor medido. El dispositivo electrónico que realiza esta operación se denomina convertidor analógico/digital o abreviadamente convertidor A/D.

Desde el lado del sistema de enseñanza-aprendizaje hacia el sistema experimental (ver figura 3.12) las señales están codificadas en lógica binaria. Con carácter general, estas señales no son aptas para gobernar los actuadores, los cuales suelen necesitar magnitudes eléctricas (corriente o tensión) de valor analógico. Por tanto, en el lado del sistema experimental, la señal digital codificada en lógica binaria procedente de la computadora del usuario ha de ser convertida en una señal analógica. Esta operación es realizada por un dispositivo electrónico denominado convertidor digital/analógico o abreviadamente convertidor D/A.

La sola conversión A/D o D/A no resuelve toda la casuística presente en la traducción correcta de las señales digitales procedentes del sistema de enseñanza-aprendizaje hacia los actuadores del sistema experimental, así como las analógicas procedentes de sus sensores (en algunos casos los sensores pueden generar de forma directa señales digitales, bien porque la señal a medir es digital de por sí –interruptor abierto o cerrado por ejemplo–, o porque el sensor es parte indivisible de un circuito integrado que procesa las señales y las suministra ya en formato digital) hacia el sistema de enseñanza-aprendizaje. En general serán necesarios acondicionamientos de las señales, tales como ajustes de rangos (cambios de escala de tensiones o corrientes, para ajustar por ejemplo el rango de señal de salida de un sensor al rango de señal de entrada de un convertidor A/D) e incluso, dentro de las magnitudes eléctricas, cambios entre ellas (por ejemplo conversión de tensión a corriente o viceversa). Por último, a la hora de llevar a cabo una implementación práctica sobre un sistema real (esto se verá con detalle en los

capítulos siguientes), es preciso tener en cuenta las especificaciones técnicas de los dispositivos reales a usar, tales como potencia, precisión, rango de entrada, rango de salida, velocidad, alimentación eléctrica, etc.

En la Figura 4.1 se muestra de forma esquematizada el proceso de entrada de señales al sistema experimental procedentes del sistema de enseñanza-aprendizaje y el camino contrario. Todos los procesos y dispositivos de acondicionamiento de señales de entrada y salida, hacia y desde el experimento de laboratorio ubicado en el sistema experimental, se engloban en lo que se denomina sistema de adquisición y distribución de señales (SADS). La tecnología actual permite disponer de SADS compactos (dispositivo único con un número determinado de entradas y salidas) programables.

Por último la Figura 4.1 incluye dos elementos más: el procesador/computador digital y la interfaz de red. El procesador/computador digital gobierna el funcionamiento del SADS y sirve de vehículo de conexión, a través de la interfaz de red, con el sistema de comunicaciones. Realmente la tecnología actual permite multitud de configuraciones para el esquema de la Figura 4.1. Así por ejemplo, el procesador/computador digital puede ser un PC con el SADS integrado (tarjeta alojada en uno de sus puertos de expansión interiores o exteriores – USB, PCI...-) y la *interfaz de red* también (tarjeta de red conectada a uno de sus puertos o integrada en la propia placa base). Más aún, hoy en día hay disponibles plataformas autónomas de muy pequeño tamaño, basadas en una placa con un microcontrolador y un entorno de desarrollo integrado que resuelven, salvo la etapa de acondicionadores de señal, el esquema de la Figura 4.1 desde el sistema de comunicaciones hasta los sensores y actuadores, [Maiti et al. 2015]. No obstante la implementación tecnológica concreta, como ya se ha dicho, será objeto del capítulo siguiente.

Respecto de la propia realización del experimento de laboratorio remoto, se ha de reparar en el modo de interacción entre el usuario ubicado en el sistema de enseñanza

aprendizaje y el experimento de laboratorio ubicado en el sistema experimental (ver Figura 3.1). Así, la interacción puede ser en línea si el usuario está interactuando con el experimento mientras éste se realiza, o fuera de línea si el usuario envía los datos/parámetros necesarios para la realización del experimento al procesador/computador digital ubicado en el sistema experimental, de modo que éste toma el control del experimento y, cuando éste se ejecuta, envía los resultados al usuario. Tanto en un caso como en otro, es preciso implementar los mecanismos de sincronización necesarios en el sistema de comunicaciones para evitar errores y/o pérdida de información. Esto se verá con detalle en el capítulo siguiente.

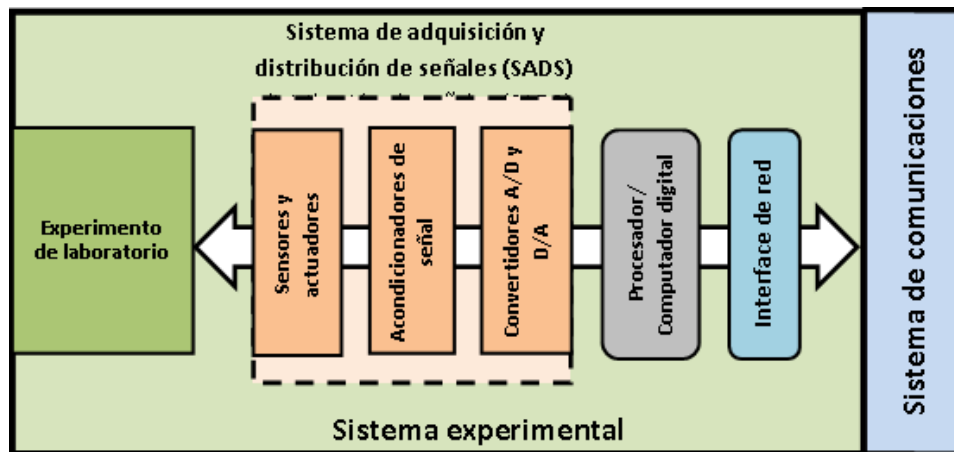


Figura 4.1: Elementos del sistema experimental que permiten el intercambio de señales, a través del sistema de comunicaciones, desde y hacia el sistema enseñanza-aprendizaje que aloja al usuario (ver Figura 3.12).

Cuando la interacción usuario-experimento se hace en línea es preciso tener muy en cuenta la *latencia* de la red de comunicaciones, [Szymaiak et al. 2008]. Ésta adquiere aún más importancia si el control del experimento (procesado de señales y acciones de control sobre el mismo) se realiza en el sistema de enseñanza aprendizaje (ver Figura 3.2), esto es, en la computadora del usuario. El cierre del lazo de desarrollo experimental remoto de esta forma, si bien deja el control del experimento completamente en manos del usuario, puede traer como desventaja la generación de

inestabilidades y errores debidos a la *latencia*, hasta el punto que pueden hacer irrealizable el experimento. Una forma de solucionar este problema es dejar en manos del procesador/computador digital ubicado en el sistema experimental (ver Figura 4.1) las labores de control sobre el experimento (o al menos las más críticas), de modo que se generen lazos de control locales (experimento de laboratorio – procesador/computador digital) sin salir del sistema experimental, quedando el lazo de desarrollo experimental remoto para tareas menos críticas, como por ejemplo intercambio de información y valores de parámetros/consignas sobre el experimento. El valor de la latencia depende de la tecnología de red utilizada y constituye uno de los condicionantes de comunicación más complejo de resolver.

Todo lo dicho permite entrever que aún dentro del concepto de laboratorio remoto, el diseño y preparación de los experimentos de laboratorio requiere de un análisis minucioso, ya que dependiendo de la naturaleza de éstos, la configuración de funcionamiento del laboratorio puede ser diferente. Lo que sí es muy importante es que esta configuración pueda realizarse desde alto nivel, con objeto de que el usuario no avezado pueda obviar los condicionantes tecnológicos implicados.

Para completar esta sección introductoria se ha de mencionar a dos subsistemas que, aunque no participan de forma explícita en la realización de la clase práctica de laboratorio remoto (CPLR), son elementos muy importantes para su mejor y eficiente implementación; se trata, ver figura 3.12, de los subsistemas de alimentación eléctrica y de transmisión multimedia, los cuales serán diseñados de forma conceptual en este capítulo e implementados de forma práctica en el siguiente.

Con objeto de ir desarrollando todos los elementos del lazo de desarrollo experimental remoto, vamos a analizar a continuación los condicionantes que impone la red para la implementación del sistema de comunicaciones. En efecto, para evolucionar el modelo conceptual teórico desarrollado en el capítulo 3 hacia el modelo tecnológico

genérico objeto del presente capítulo, es necesaria una mayor concreción del sistema de comunicaciones, así como una mayor especificación de la estructura del sistema de control de las comunicaciones, que en nuestro modelo hemos denominado SARLAB. El primer paso en la concreción del modelo será considerar la utilización de una red genérica de paquetes en la implementación de la red de comunicaciones del modelo conceptual como la arquitectura de red posible, [Tanenbaum and Wetherall 2014]. Por tanto, se comienza con la presentación de las consideraciones generales sobre la red del sistema de comunicaciones. Una vez introducida la tecnología de paquetes en el sistema de comunicaciones, se plantea cómo se puede establecer el lazo de desarrollo experimental remoto a través de esta tecnología desde la perspectiva del telecontrol y la telepresencia.

A continuación se estudian los problemas fundamentales de telecontrol y presencia experimental que se establecen en el lazo de desarrollo experimental para la realización de una experiencia a través de una red de paquetes, desde el punto de vista de una arquitectura funcional y física. Finalmente se presentan el *subsistema de interacción* y el de análisis, diseño y programación experimental, así como los de alimentación eléctrica y transmisión multimedia.

4.2. Consideraciones generales sobre la red del sistema de comunicaciones

El concepto de acceso a la red de comunicaciones que permite implementar el modelo de laboratorio remoto que desarrolla esta investigación pretende ser lo más general y abierto posible, entendiéndose con ello que el usuario puede acceder al laboratorio remoto desde una red pública abierta y, desde ella, con los lógicos permisos, interaccionar con el experimento de laboratorio que puede estar ubicado en una o varias redes locales propiedad de diferentes organismos y/o empresas. Estas redes públicas

abiertas de carácter universal son hoy día una realidad con cada vez más cobertura, tanto desde ubicaciones fijas como a través de dispositivos móviles, [Odlyzko 1999], [Raychaudhuri and Mandayam 2012].

El análisis que sigue de la red de comunicaciones, en su intención de ser general, de modo que pueda dar cabida a las tecnologías actuales y las venideras, pretende alcanzar un modelo físico y funcional que refleje los aspectos relevantes a tener en cuenta cuando se plantea el control de un proceso (la realización de una clase práctica de laboratorio remoto –CPLR– en nuestro caso) a través de una red de datos. Se analizarán por tanto las características de la red que afectan a la gestión de la información de la CPLR desde el sistema de enseñanza-aprendizaje hasta el *sistema experimental* y viceversa. El análisis se realizará sin asociarlo a ninguna tecnología de implementación de red particular, con objeto de conservar su generalidad, dejando para los capítulos siguientes los condicionantes tecnológicos propios de la implementación práctica que se realice. Así pues, el estudio se va a centrar en los aspectos lógicos y de tratamiento de la información de una arquitectura de red de comunicaciones lo más general posible.

4.2.1. Conceptos generales de la arquitectura de una red de paquetes

Este apartado está dedicado a conocer los principios de transmisión de información a través de una red de paquetes (o de conmutación de paquetes como también se denomina), la cual transmite datos encapsulados con la información suficiente para que, independientemente del camino intermedio que recorran, lleguen a su destino en forma correcta. Este tipo de red se considera genérica, en el entendimiento de que el avance tecnológico dotará a la misma de cada vez más funcionalidades y capacidades, pero asumiendo que la filosofía de trabajo en la red permanecerá en el tiempo, al menos en los próximos años. De hecho, la arquitectura más general que se puede plantear para una red de comunicaciones estaría formada por un conjunto de elementos

interconectados entre sí por algún medio físico que facilitará el intercambio de información entre todos los elementos de la red (ver Figura 4.2). Cada uno de los elementos que constituye la red se denominará nodo, y se denominará enlace a la transferencia de información entre los nodos. Por supuesto, a la hora de implementar de forma práctica la red, los nodos han de ser compatibles con las tecnologías de transmisión de los enlaces. La información en la red va de un nodo a otro (nodos extremos) atravesando nodos intermedios. Éstos vertebran la red, de modo que cuando dos nodos extremos desean generar y recibir comunicaciones, los nodos intermedios encaminan (proporcionan el enlace) el tránsito de los paquetes para que la comunicación entre los nodos extremos funcione de forma correcta. De forma general, cualquier par de nodos de la red puede ser en un momento dado nodos extremos y, también, cualquier nodo de la red puede ser en un momento dado intermedio. Esto es, todos los nodos pueden comunicarse entre sí, propiedad que se conoce como conectividad global.

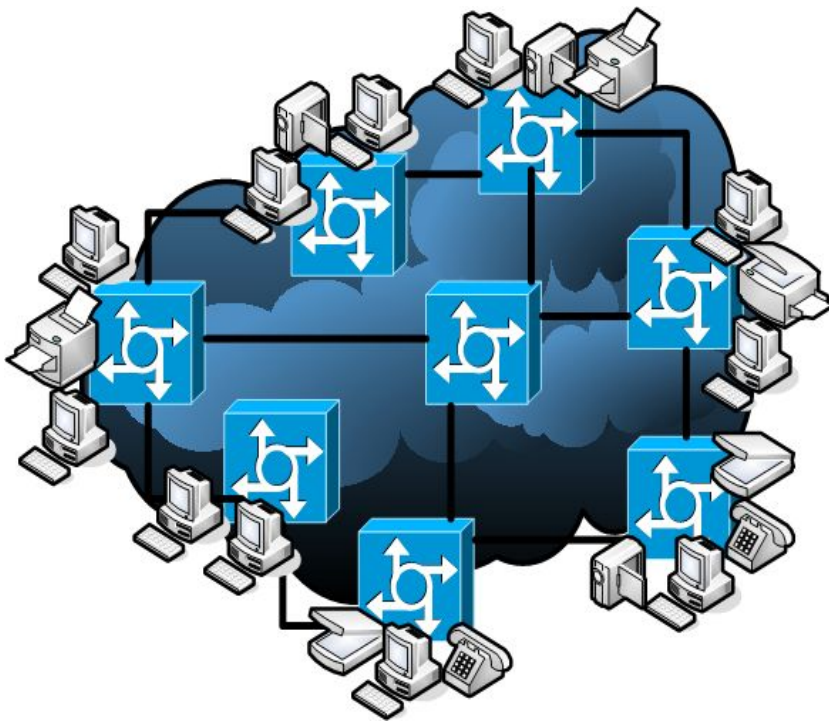


Figura 4.2: Estructura general de una red de comunicaciones.

El modelo planteado en este punto es tan genérico que permite referenciar cualquier arquitectura que pueda llegar a tener una red de comunicaciones independientemente de la tecnología de implementación.

La red, planteada de una forma tan general, es un sistema que debe contar con los elementos adecuados para que el hecho de compartir el espacio físico (cable o espectro radioeléctrico) por parte de los usuarios sea ordenado y no se pierda información por el camino. Esto es, el espacio físico no es de uso exclusivo para ninguna comunicación¹, con lo cual la información de múltiples comunicaciones va mezclada por el mismo espacio físico (los paquetes van mezclados). Por tanto, para que un nodo intermedio pueda distinguir en cada paquete que ha de enlazar de qué nodo extremo proviene y a qué nodo extremo va (dirección de los nodos extremos), el paquete ha de llevar dicha información. Así, el nodo extremo que envía el paquete encapsula la información que quiere enviar añadiendo una cabecera que contiene la dirección del nodo de destino y de origen. Los nodos intermedios utilizan esta información para encaminar desde origen a destino la información como se puede ver en la Figura 4.3. En ella se muestra un ejemplo de comunicación donde el paquete que se transmite sólo lleva información de dirección y datos, aunque también podría llevarla (en la cola) de detección de errores. Los nodos intermedios utilizan la información de cabecera para encaminar los datos desde origen a destino.

Abundando en el funcionamiento de una red de paquetes, ha de tenerse en cuenta que incluso no es posible garantizar que todos los paquetes de una comunicación sigan el mismo camino, con lo cual, al tener los caminos un retardo (latencia) variable, no es posible garantizar que la información llegue ordenada al nodo destino. Es más, si los

¹ Pueden haber redes, normalmente de uso restringido, donde esto no sea así, esto es, que el camino establecido (circuito) entre nodos extremos sea de uso exclusivo de éstos, bien porque sólo hay dos nodos o porque los nodos intermedios reservan el espacio físico para el uso exclusivo de los nodos extremos. El circuito establecido puede permitir el tráfico de información en los dos sentidos o puede ser doble, uno para mandar información y otro para recibirla. Esto tiene como gran ventaja que el ancho de banda de la comunicación pueda quedar garantizado, sin embargo, la red queda desaprovechada (sin uso) cuando no hay comunicaciones entre los nodos extremos. Este tipo de redes se denominan de conmutación de circuitos.

nodos intermedios utilizan mecanismos de recuperación de errores temporizados, podría llegar la misma información por varios caminos. Estas cuestiones serán estudiadas en profundidad en los siguientes capítulos, concretándolas para la tecnología de implementación práctica que vamos a usar en este trabajo de investigación.

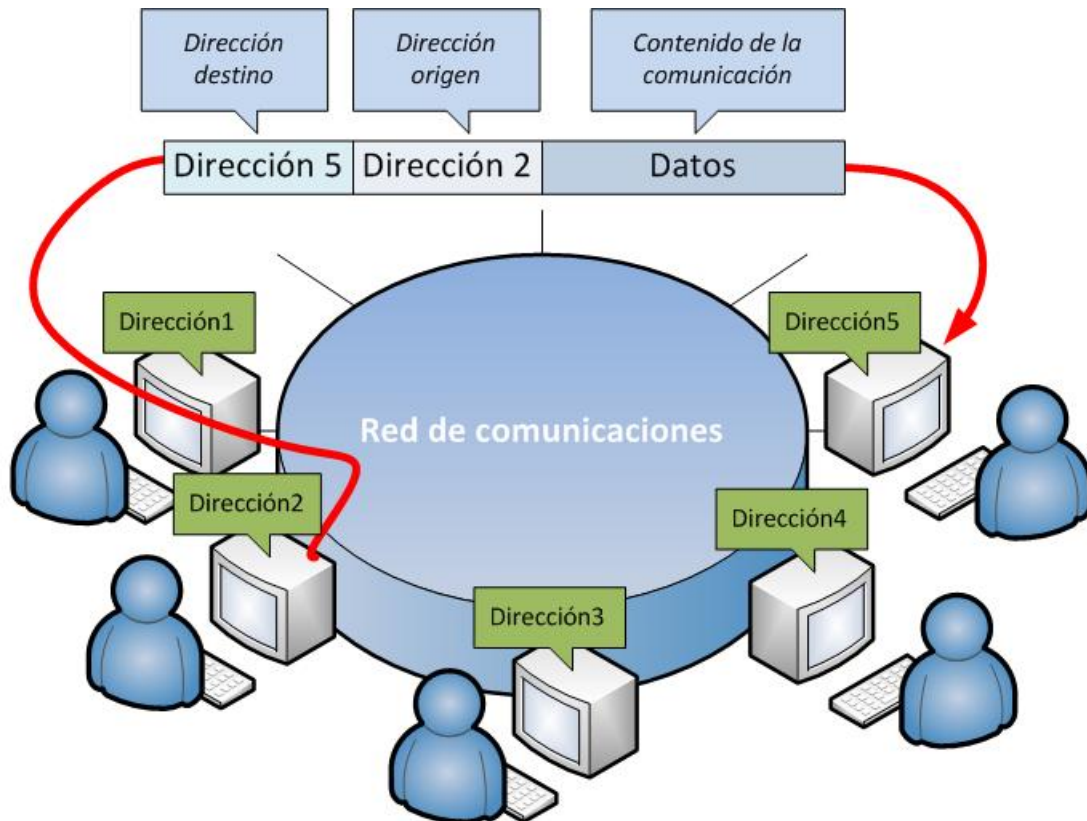


Figura 4.3: Ejemplo de transmisión de un paquete de datos por la red.

Como ya se ha dicho, una red de paquetes gestiona en sus enlaces tráfico mezclado de múltiples usuarios, lo cual, si bien permite optimizar el uso de estos enlaces ya que siempre tienen tráfico que gestionar, exige el reparto del *ancho de banda* disponible entre los usuarios participantes. Esto nos lleva a remarcar el hecho siguiente:

El ancho de banda de un usuario en una red de paquetes es variable y, como consecuencia, la velocidad de transmisión no es ni constante ni determinista, ya que no

se puede conocer de forma exacta la demanda de cualquier enlace de la red en un momento temporal concreto, [Hua et al. 2013], [Juanping and Xianwen 2009].

Dicho esto, hay que reconocer el avance tecnológico constante en las redes de paquetes, el cual permite aminorar en gran medida sus limitaciones. En concreto y para las redes públicas, que son las de mayor interés en este trabajo de investigación, ya que garantizan la universalidad del servicio: (1) El aumento de ancho de banda es continuo, tanto vía cable como sin él y tanto con estructura fija como móvil. (2) El aumento de la calidad es también continuo, lo cual se traduce en la posibilidad de priorizar el tráfico que no pueda soportar retraso variable. (3) Es posible garantizar un *ancho de banda* mínimo, lo cual permite establecer velocidades de transmisión garantistas. No obstante, la calidad del servicio depende fuertemente de la ubicación geográfica de los usuarios, de modo que su universalidad a nivel mundial no es hoy por hoy garantizable, [Pérez 2008], [Myakotnykh and Helvik 2009], [Szymaniak et al. 2008].

Puesto que el modelo que se está desarrollando en esta investigación no tendrá concreción tecnológica hasta los capítulos siguientes, los conceptos de ancho de banda y latencia no pueden ser considerados aún, debido precisamente a su variabilidad e indeterminación. Por tanto, para que la solución que se propone sea independiente de la realidad tecnológica actual y pueda adaptarse a la evolución tecnológica de las redes, se parte de una idealización que se denomina red ideal de paquetes de ancho de banda infinito y latencia cero.

4.2.2. Protocolo de comunicaciones

El proceso de comunicación de información entre dos entidades a través de la red, independientemente de la tecnología de comunicación que se utilice, es un proceso muy complejo. Los dispositivos que gestionan las comunicaciones deben encargarse del acceso al medio de transmisión que normalmente está compartido por múltiples

usuarios, evitar la pérdida de información, eliminar errores de transmisión, ordenar el tráfico que llegue desordenado, eliminar el tráfico repetido, resolver la congestión de los nodos intermedios, permitir la compatibilidad de distintas tecnologías de transmisión, recuperar comunicaciones perdidas sin empezar desde el principio, y un largo etc.

En el ámbito de las comunicaciones, el conjunto de reglas que regula el intercambio de información entre dos entidades resolviendo la problemática global expuesta en el párrafo anterior se denomina protocolo de comunicaciones, [Stallings 2011].

Teniendo en cuenta la complejidad del proceso de comunicación de información entre dos entidades a través de la red, es fácil intuir que la arquitectura de un protocolo, esto es, su organización, no es tarea sencilla. Por ello, la organización de un protocolo se aborda mediante una subdivisión funcional por niveles o capas, que suele denominarse arquitectura de protocolo. En ella, cada nivel se encarga de resolver una cuestión concreta de la problemática global del proceso de comunicaciones, esto es, tiene una función concreta que lo distingue de los demás. Como características generales de estos niveles se podría resaltar para nuestro estudio las siguientes:

- ✓ El número de niveles debe ser finito y mínimo.
- ✓ Cada nivel se ocupa de realizar una parte de todo el proceso de comunicación.
- ✓ Cada nivel se comunica con sus dos niveles contiguos a través de una interfaz.
- ✓ Todos los niveles ofertan servicios al nivel superior y demandan servicios del nivel inferior solo a través de las interfaces.

En la comunicación entre dos entidades con una misma arquitectura de protocolo, la información (paquete de datos) en el ente transmisor desciende verticalmente por

niveles (ya que ha de llegar hasta el nivel más bajo, el que controla las conexiones de su computador con la red, tanto físicamente, como un determinado tipo de conector y cable por ejemplo, como el modo en el que se ubican los datos en el medio físico sobre el que se transmite la información). Cada nivel añade la información que procede a la cabecera del paquete, de modo que cuando éste salga del ente transmisor quede perfectamente establecido qué se debe hacer con el paquete de datos transmitido, tanto en su viaje por la red como en su interpretación por el ente receptor. En el ente receptor la información se mueve en sentido contrario, ya que asciende desde el nivel que representa el medio físico (por ejemplo, otro tipo de cable y conector diferente al presente en el ente transmisor) hasta el nivel superior, el que interactúa con la aplicación informática en red que esté manejando el ente receptor. En su ascenso, la comunicación es interpretada en cada nivel atendiendo a lo establecido en su cabecera.

De lo anterior se deduce que, de forma general, la arquitectura del protocolo debe resolver en sus niveles más bajos el flujo de datos hacia o desde el medio físico de la red y, en sus niveles más altos, la gestión de la información hacia o desde la aplicación informática que maneja el usuario. Por tanto se puede hacer una primera división en la arquitectura del protocolo agrupando por un lado los niveles que atienden al flujo de datos (se denominará a este agrupamiento capa de flujo de datos) y, por otro, los niveles que atienden a la aplicación, denominando a este agrupamiento capa de aplicación. Esta arquitectura de protocolo se muestra en la Figura 4.4.

Por concretar con el estándar actual, si se encaja la arquitectura de protocolo genérica de la Figura 4.4 en el *modelo OSI*, la capa de flujo de datos encapsula el tratamiento de los niveles (de abajo hacia arriba) Físico, Enlace, Red y Transporte, y la capa de aplicación encapsula los niveles de Sesión, Representación y Aplicación.

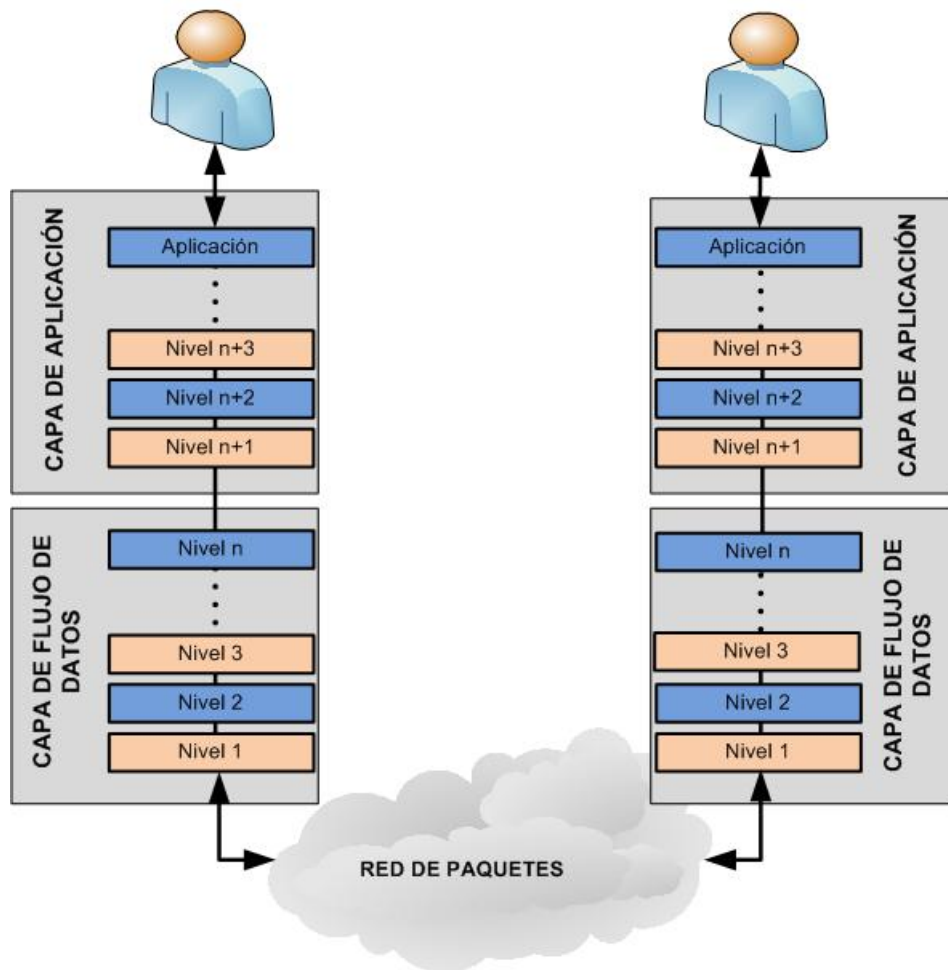


Figura 4.4: Arquitectura de protocolo genérica.

4.2.3. Arquitectura cliente-servidor

En la red y de forma dinámica hay continuamente demandantes de servicios y recursos, con lo cual y para que el sistema funcione adecuadamente, debe haber proveedores de los recursos demandados. Esto, en un sistema distribuido como es la red, conlleva un reparto de tareas entre los proveedores de servicios y recursos, denominados servidores, y los demandantes de los mismos, denominados clientes. Esta organización de la red se denomina arquitectura cliente-servidor.

La arquitectura cliente-servidor es un modelo de aplicación distribuida en red, donde la capacidad de proceso está repartida entre los clientes y los servidores. Esta

arquitectura aporta ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema de comunicaciones, [Tanenbaum 1996].

La separación entre cliente y servidor es una separación de tipo lógica no física, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Por fijar ideas, en el modelo de red actual, los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc.

En la arquitectura de red cliente-servidor los clientes se conectan al servidor. En él se centralizan los diversos recursos y aplicaciones con que se cuenta y los pone a disposición de los clientes cada vez que éstos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor. Éste actúa como sujeto pasivo especializado en realizar ciertas tareas bajo los requerimientos de los clientes que son el sujeto activo.

El modelo propuesto utilizará la arquitectura cliente-servidor en la capa de aplicación del protocolo de comunicaciones. El cliente (alumnos, profesores y gestores del aprendizaje, ver Figura 3.12) está en el sistema de enseñanza-aprendizaje y accede mediante su aplicación cliente al correspondiente servidor, que forma parte del sistema de comunicaciones y ejecuta su aplicación servidor. Esto puede verse esquemáticamente en la Figura 4.5.

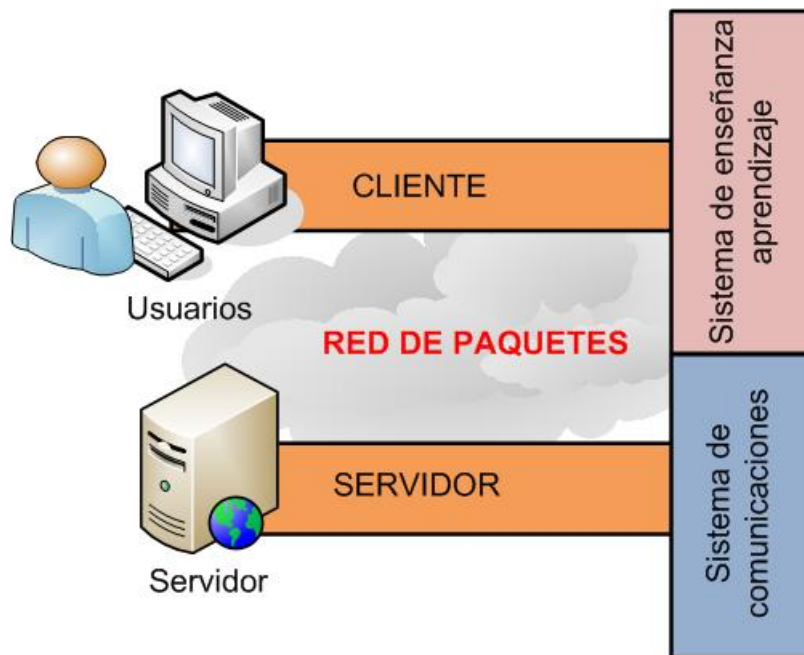


Figura 4.5: Arquitectura cliente-servidor aplicada al modelo de la Figura 3.12.

4.3. Desarrollo experimental a través de una red de paquetes

En los dos puntos anteriores ha quedado establecido que la convergencia de todas las variables experimentales (sensores, actuadores, interfaces de programación, etc.) presentes en una experiencia de cualquier naturaleza convierte el problema del desarrollo del modelo tecnológico genérico en un problema de comunicaciones. Este problema presenta asimismo dos vertientes: por un lado, el alumno debe controlar la evolución de la experiencia (telecontrol) y por otro debe tener en todo momento la sensación de estar en presencia física del fenómeno experimental. Para ello, se deberá conseguir la realización de la experiencia remota de la forma más parecida posible a la experiencia real, lo que permite maximizar los resultados del aprendizaje de forma remota.

4. Modelo General Aplicado de un Laboratorio Remoto

En este apartado se pretende por tanto introducir los condicionantes de las redes de paquetes expuestos en el apartado anterior en el lazo de desarrollo experimental. Este lazo se podrá cerrar a través de una red de paquetes siempre que los dos extremos de la comunicación (el subsistema de análisis, diseño y programación en uno y el sistema experimental en el otro) dispongan de interfaces convergentes. El lazo de desarrollo experimental remoto, indicado en la figura 3.12, se presenta en la 4.6 a través de una red de paquetes.

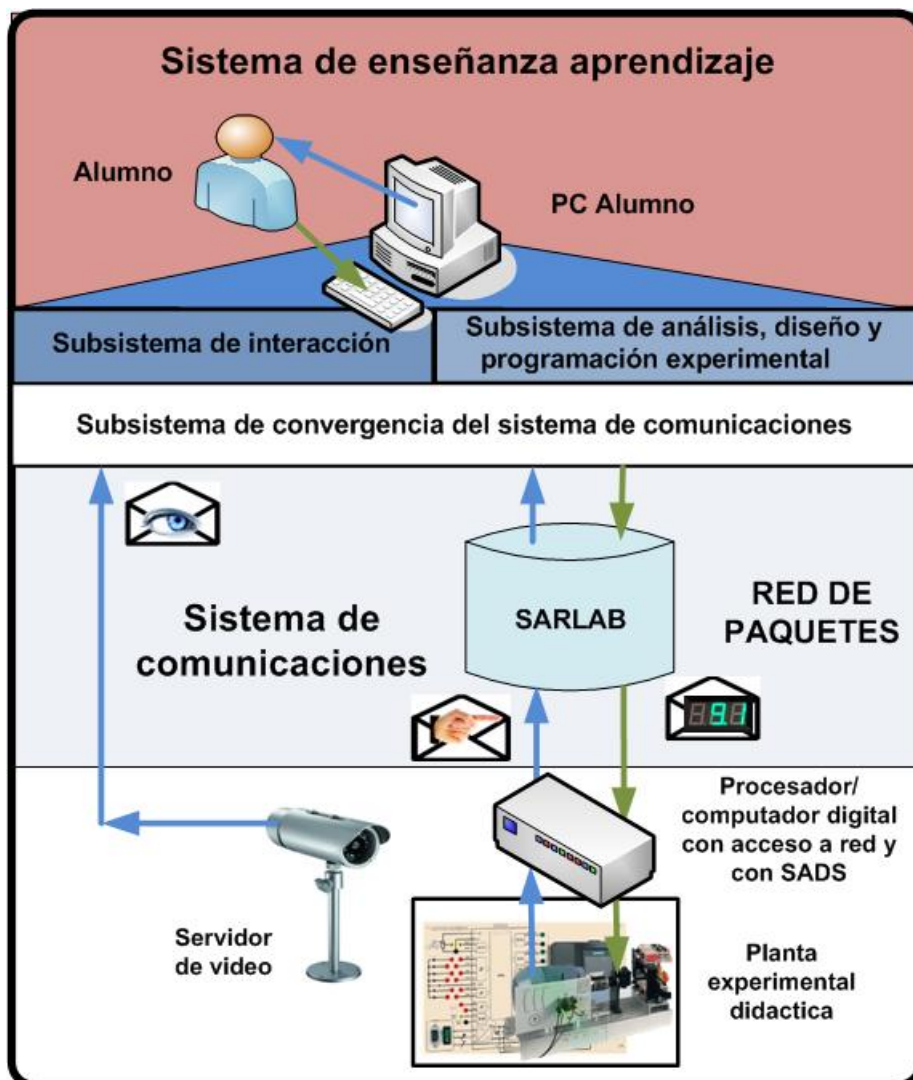


Figura 4.6: Lazo de desarrollo experimental remoto a través de una red de paquetes, que incluye el servidor de vídeo y audio.

En la figura 4.6 se muestra, de un lado, el PC con el que el alumno accede a las experiencias de laboratorio. En él se ejecutarán las aplicaciones necesarias para aportar las funcionalidades del *subsistema de interacción* y del de análisis, diseño y programación. Estos dos subsistemas son accesibles desde red a través del subsistema de convergencia del sistema de comunicaciones. Se presenta en la figura 4.6, además, el sistema experimental con la estructura genérica del sistema SADS (indicado en el apartado 4.1) y el subsistema de transmisión multimedia encargado de proporcionar al alumno la visión de la evolución de la experiencia. A partir de estos elementos, la experiencia remota evoluciona como consecuencia de las acciones que el alumno realiza en el PC sobre el subsistema de interacción. Estas acciones se envían en forma de paquetes a través de la red al procesador/computador digital mostrado en la figura 4.6. El computador recibe estos paquetes por el subsistema de convergencia del sistema de comunicaciones y los traduce a las correspondientes acciones sobre el experimento, utilizando el SADS. Para cerrar el lazo, la evolución del experimento es detectada por los sensores y adquirida desde el SADS de forma que el procesador/computador digital la transforma en paquetes y la envía al PC del alumno. Allí, la información sobre la evolución es procesada y presentada por el subsistema de análisis, diseño y programación. Para reforzar la percepción experimental, el subsistema de transmisión multimedia envía imágenes y audio del desarrollo de la experiencia de forma continua y encapsulada en forma de paquetes.

A partir de este punto, un sistema experimental es convergente si se pueden realizar todas las acciones y recibir toda la información necesaria para el control de la experiencia a través de su interfaz de red. Por tanto, en lo sucesivo el término convergente se asociará a cualquier dispositivo que tenga accesibles sus funcionalidades desde la interfaz de la red de paquetes.

4.4. Estructura genérica de un sistema experimental convergente

En este apartado, se desarrolla la estructura del sistema experimental convergente más genérico. Para ello, se considera que la complejidad del sistema experimental está directamente asociada a la complejidad del experimento, y está por tanto asociada al aumento del número de procesadores/computadores con SADS convergentes. Por tanto, la estructura más general sería la correspondiente a una planta experimental didáctica cuyos sensores y actuadores tuvieran que estar conectados a varios procesadores/computadores con SADS convergentes, que presentarían un conjunto de interfaces de red, como el mostrado en la figura 4.7.

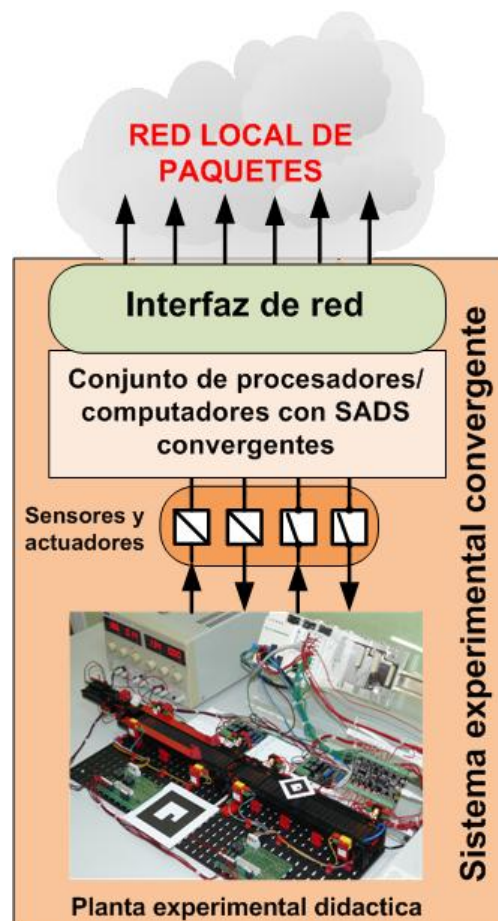


Figura 4.7: Sistema experimental convergente con varios procesadores/computadores con SADS convergentes.

Por otro lado, existen cada vez más dispositivos digitales programables desde red de propósito específico, como por ejemplo instrumentos de medida programables, autómatas, controladores digitales programables, sistemas de desarrollo programables, etc. La presencia de estos dispositivos en el sistema experimental, aporta, además de las funcionalidades propias de los procesadores/computadores con SADS convergentes, el valor didáctico del aprendizaje de su programación y configuración. Por tanto, su uso en las experiencias de laboratorio aumenta la complejidad de la misma, además de por suponer un elemento más, por las necesidades de acceso que requiere su uso didáctico: habrá que acceder a ellos, no solo como gestores de la información de sensores y actuadores, sino como elementos finales de la información para su programación y configuración por parte del alumno a través de las herramientas propietarias proporcionadas por los fabricantes. El nuevo sistema experimental, más general que el anterior porque incluye este nuevo tipo de dispositivos, se muestra en la figura 4.8,

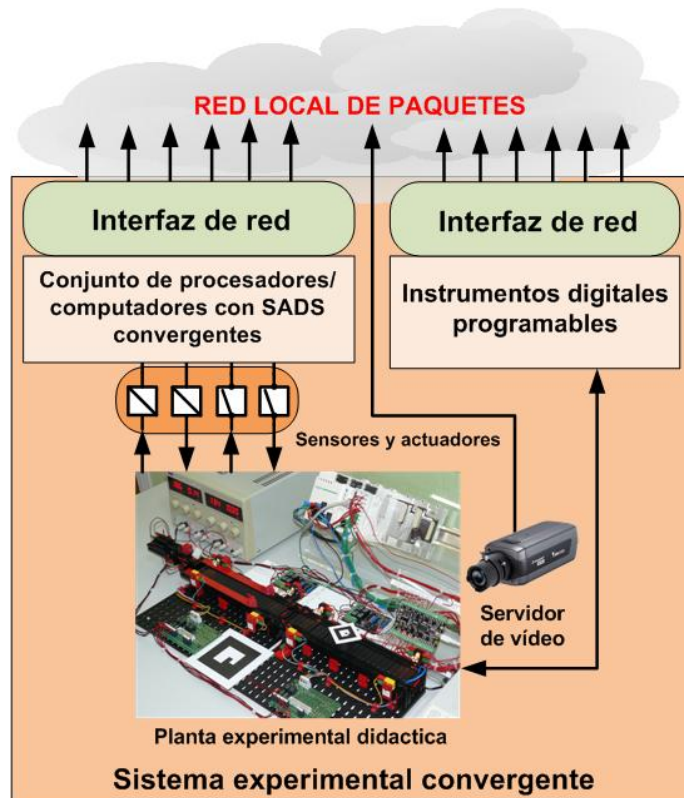


Figura 4.8: Sistema experimental convergente con procesadores/computadores con SADS convergentes e instrumentos digitales programables.

donde se han incluido además los servidores de vídeo y audio, presentes en la figura 4.6. El modelo mostrado en la figura 4.8 es el más complejo que se considera en este estudio, y básicamente está compuesto por un conjunto de procesadores/computadores con SADS convergentes y un conjunto de instrumentos digitales programables. Desde el punto de vista de su acceso remoto, este modelo se reduce a un sistema experimental con una interfaz de convergencia a red, como se muestra en la figura 4.9. La interfaz de convergencia a red proporciona la capacidad de telecontrol de la experiencia y la percepción de la evolución del experimento a través de la red. La interfaz tendrá asimismo los puntos necesarios para acceder a cada uno de los elementos convergentes de la figura 4.8.



Figura 4.9: Representación externa del sistema experimental convergente más general.

4.5. Estructura física y funcional del modelo tecnológico genérico

En los apartados anteriores se ha establecido que el lazo de desarrollo experimental puede ser gestionado a través de una red de paquetes y se ha desarrollado la estructura más genérica posible de un sistema experimental convergente. En este apartado se van a incluir estos resultados en el modelo conceptual desarrollado en el capítulo 3. En este modelo, se introduce SARLAB como el túnel de comunicaciones que debe implementar

todas las funcionalidades necesarias para que el lazo de desarrollo experimental remoto se pueda cerrar de forma transparente para el usuario, en cualquier escenario de redes públicas/privadas e independientemente de los controles de acceso y gestión que éstas impongan. En este sentido, para que el experimento sea accesible desde cualquier punto, el PC del alumno deberá estar conectado a la red pública. Asimismo, para que el acceso al sistema experimental pueda ser controlado, el sistema presentado en la figura 4.9, que representa el sistema experimental convergente, deberá ser accesible desde una red local. La necesidad de controlar el acceso remoto a experiencias es consecuencia de que el usuario interactúa con elementos reales que, o bien no pueden ser manejados de forma simultánea por varios de ellos, o bien ese uso simultáneo debe hacerse de forma colaborativa.

En el modelo tecnológico general se propone que SARLAB sea el sistema frontera entre estas dos redes, como se muestra en la estructura física presentada en la figura 4.10. La estructura funcional mostrada en esa misma figura es la presentada en la 4.6, donde, además de introducir las dos redes descritas, se ha introducido el modelo cliente-servidor para el establecimiento del lazo de desarrollo experimental remoto mediante una red de paquetes (como condicionante de esta tecnología para el desarrollo de aplicaciones) y para conseguir la percepción experimental.

Este modelo permite, además, gestionar varias experiencias a partir de un mismo sistema SARLAB, como se muestra en la figura 4.11. En efecto, cada experiencia de laboratorio queda definida en el servidor SARLAB por una estructura de datos formada por los pares de puntos conectados. Un componente del par representa el punto de conexión en el PC del alumno y el otro el punto de conexión en el sistema experimental convergente. El conjunto de los pares de puntos necesarios conforman la estructura de datos de una experiencia. El conjunto de varias estructuras de datos conforman un laboratorio remoto. De esta forma, SARLAB se convierte en un gestor de laboratorios remotos en lugar de ser un simple gestor de experiencias. Esto introduce la posibilidad

Estructura funcional | Estructura física

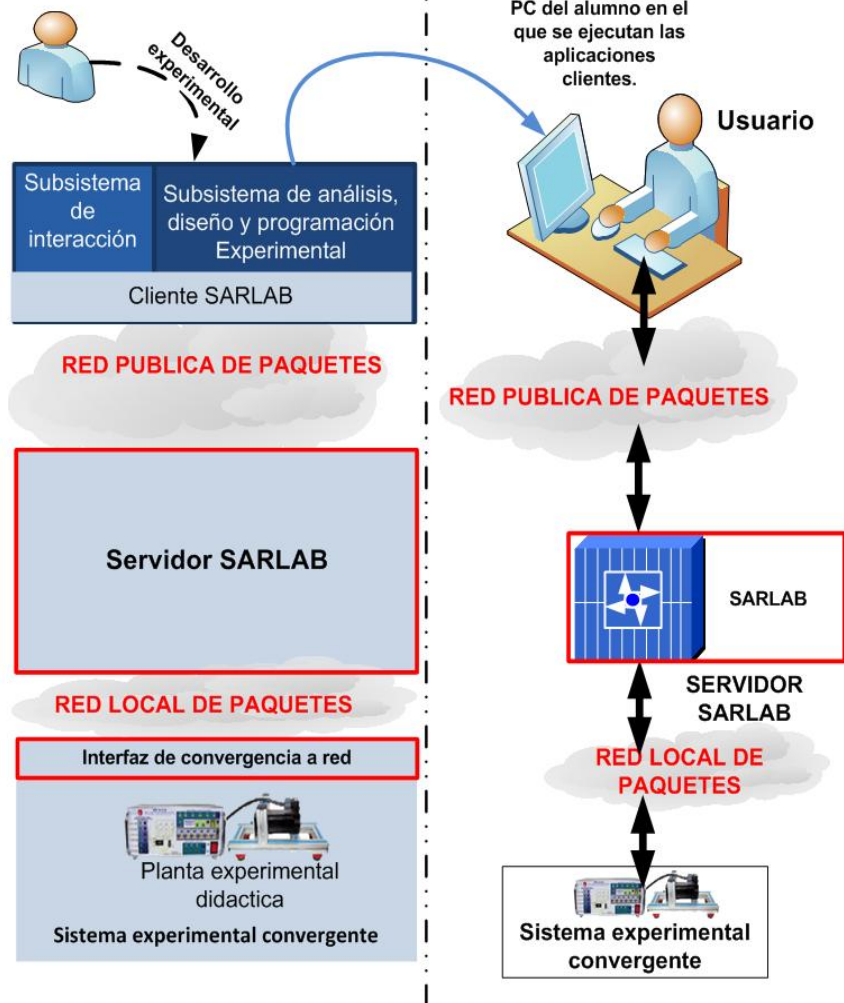


Figura 4.10: Estructuras funcional y física del modelo tecnológico general.

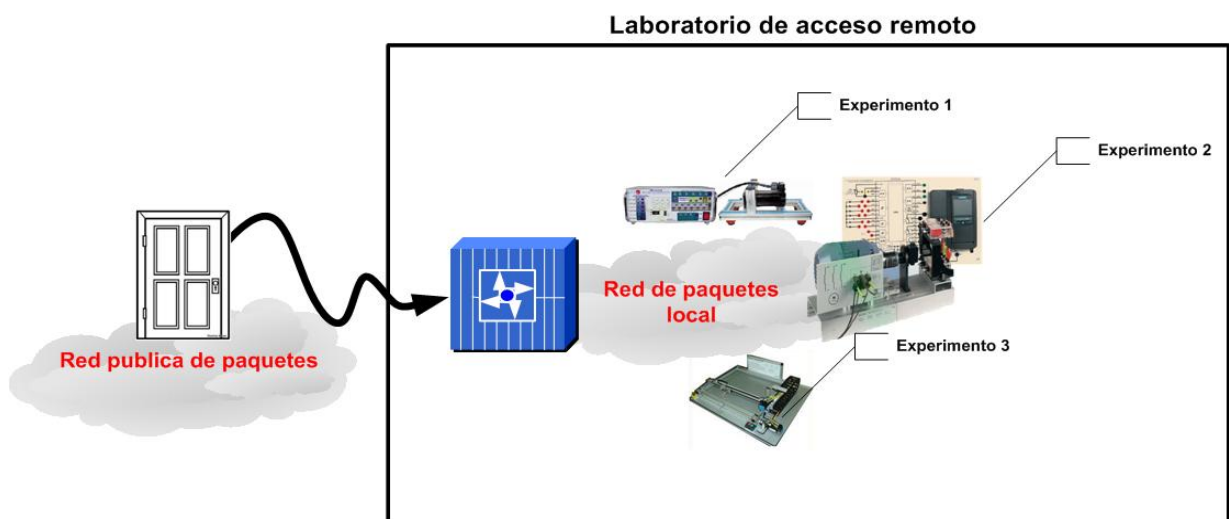


Figura 4.11: Estructura de un laboratorio de acceso remoto.

de unificar los criterios de organización de distintas instituciones y establecer políticas de uso común a todas las experiencias, o bien la de delegar la gestión de los laboratorios entre múltiples instituciones.

4.5.1. Estructura funcional del servidor SARLAB para accesos múltiples

Para desarrollar la estructura funcional del servidor SARLAB, hay que considerar por un lado el sistema experimental convergente como el caso más general posible de una experiencia remota, y por otro la posibilidad de que SARLAB gestione varios de ellos de forma concurrente.

En este escenario de comunicaciones la funcionalidad principal de SARLAB es establecer las conexiones necesarias entre las aplicaciones que permiten la interacción, análisis y diseño que se ejecutan en el PC del alumno y los componentes correspondientes del sistema experimental convergente asociados a la práctica de laboratorio. En el caso más general, cada conexión puede establecerse con un protocolo de comunicación distinto, al cual debe ser transparente la interconexión realizada por SARLAB. Esta funcionalidad de establecimiento de conexiones se denomina *CONEXLAB* (*CONexionado de EXperiencias de LABORatorio*). En este sentido, la estructura de datos definida en el apartado 4.5 representa la configuración matricial de conexiones que el servidor SARLAB, y más concretamente CONEXLAB, debe establecer para permitir la realización de una experiencia.

Por tanto, una experiencia queda definida por su estructura matricial de conexiones y un laboratorio queda definido por el conjunto de las estructuras matriciales de sus experiencias. Así, la gestión de un laboratorio se limita a incorporar, eliminar y modificar experiencias, lo que se traduce en nuestro modelo en incorporar, eliminar y modificar estructuras matriciales. Estas tareas se corresponden con los procedimientos

asociados a una base de datos (altas, bajas y modificaciones), por lo que el conjunto de experiencias será denominado *base de datos de representación experimental*.

Las funciones de gestión del laboratorio (altas, bajas y modificaciones de las estructuras matriciales) se asocian a una nueva funcionalidad denominada *DIGEXLAB* (*Diseño y Gestión para EXperiencias de LABoratorio*), que se encargará de gestionar las estructuras matriciales de configuración de las experiencias correspondientes a un laboratorio de acceso remoto.

Esto es, DIGEXLAB gestiona la base de datos de representación experimental. En la figura 4.12 se representa la 4.11 con la introducción en la estructura funcional de las dos

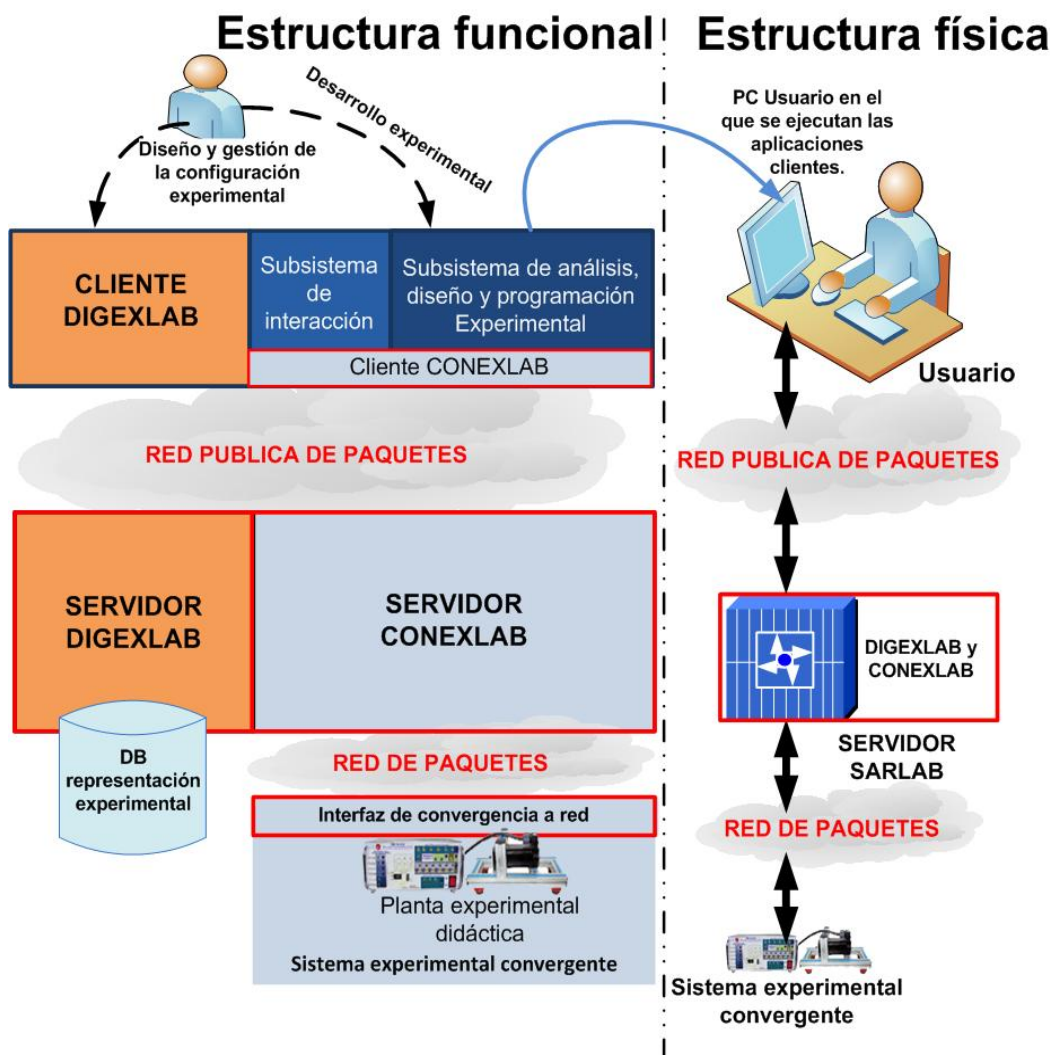


Figura 4.12: Estructuras funcional y física del servidor SARLAB.

funcionalidades presentadas en este apartado, CONEXLAB y DIGEXLAB con una estructura cliente-servidor.

4.5.2. CONEXLAB

El propósito de este apartado es profundizar en los aspectos constructivos de la funcionalidad de SARLAB que se ha denominado CONEXLAB. Éste es el encargado de establecer de forma transparente las conexiones necesarias para configurar las experiencias que se especifican en la estructura matricial definida en el apartado 4.5. El método usado en el modelo tecnológico general para establecer conexiones de forma transparente entre dos sistemas a través de una red pública es el establecimiento de túneles de comunicaciones entre ellos.

Los túneles de comunicaciones son un concepto genérico que puede utilizarse en todas las redes de paquetes para conseguir la unión de dos redes privadas, aprovechando la infraestructura de comunicaciones de una red pública de datos de manera totalmente transparente a la tecnología de red de cada una de ellas. Además, los túneles de comunicaciones son transparentes al protocolo de comunicaciones transportado, [Tanenbaum and Wetherall 2014].

Por otro lado, la estructura más general del escenario de comunicaciones está constituida por una red pública formada por la interconexión de un conjunto de redes públicas administradas por entidades distintas. Cada una de estas redes se denomina *sistema autónomo*. La interconexión de un conjunto de sistemas autónomos permite que cualesquiera dos nodos de esos sistemas puedan estar comunicados entre sí. Asimismo, cada sistema autónomo puede imponer condiciones de paso al tráfico que lo atraviesa. El conjunto de reglas que limita el paso de información a través del sistema autónomo se denomina *política de seguridad de tráfico*. El tráfico que logra superar las políticas de seguridad de un sistema autónomo se denomina *tráfico amigable*. En este sentido, el uso

de túneles proporciona, además de la transparencia indicada anteriormente, la posibilidad de superar las políticas de seguridad de cada sistema autónomo de la red, [RFC 1930].

Los túneles dirigen todo el tráfico encapsulado de la misma forma, con independencia del protocolo transportado, facilitando el cumplimiento de todas las políticas de seguridad en el tránsito entre los distintos sistemas autónomos de una red pública. Lo único necesario para ello es conseguir que el tráfico del túnel sea considerado amigable a todas las políticas. Una ventaja adicional del uso de túneles, es la posibilidad de encriptación de la información, protegiendo así el contenido de los paquetes en el tránsito por la red pública, [Dusi et al. 2009].

El principio de funcionamiento de los protocolos de implementación de túneles en las redes de paquetes, consiste en que el equipo promotor encapsula la información a transmitir en paquetes con encabezados específicos para viajar y alcanzar el destino en la red remota, a los cuales se les añade una segunda cabecera adicional y específica para poder viajar por la red pública, figura 4.13.

Esta cabecera permite que la carga útil doblemente encapsulada pueda ser encaminada a través de los nodos intermedios de la red pública y pueda alcanzar la red de destino. La trayectoria lógica a través de la que viajan los paquetes doblemente encapsulados por la red pública se denomina túnel, y se ha representado en la figura 4.13 con un punto de entrada en el nodo promotor y un punto de salida en el nodo destino, situado en la red local. Cuando los paquetes doblemente encapsulados alcanzan la red destino a través de la red pública, se procede a la eliminación de la cabecera específica de la red pública para restaurar el paquete original, que se envía por la red destino al receptor de la comunicación. Las aplicaciones que implementan túneles deben ser capaces de realizar el encapsulado, la transmisión y el desencapsulado de los paquetes para poder establecer las comunicaciones sobre una red pública.

La funcionalidad de CONEXLAB consiste por tanto en controlar, a través del establecimiento de los correspondientes túneles, el tráfico de acceso desde la red pública a la red interna del laboratorio. Esto implica que los sistemas experimentales no reciben tráfico directamente desde la red pública, sino que su acceso se consigue de forma indirecta a través de los túneles creados por CONEXLAB.

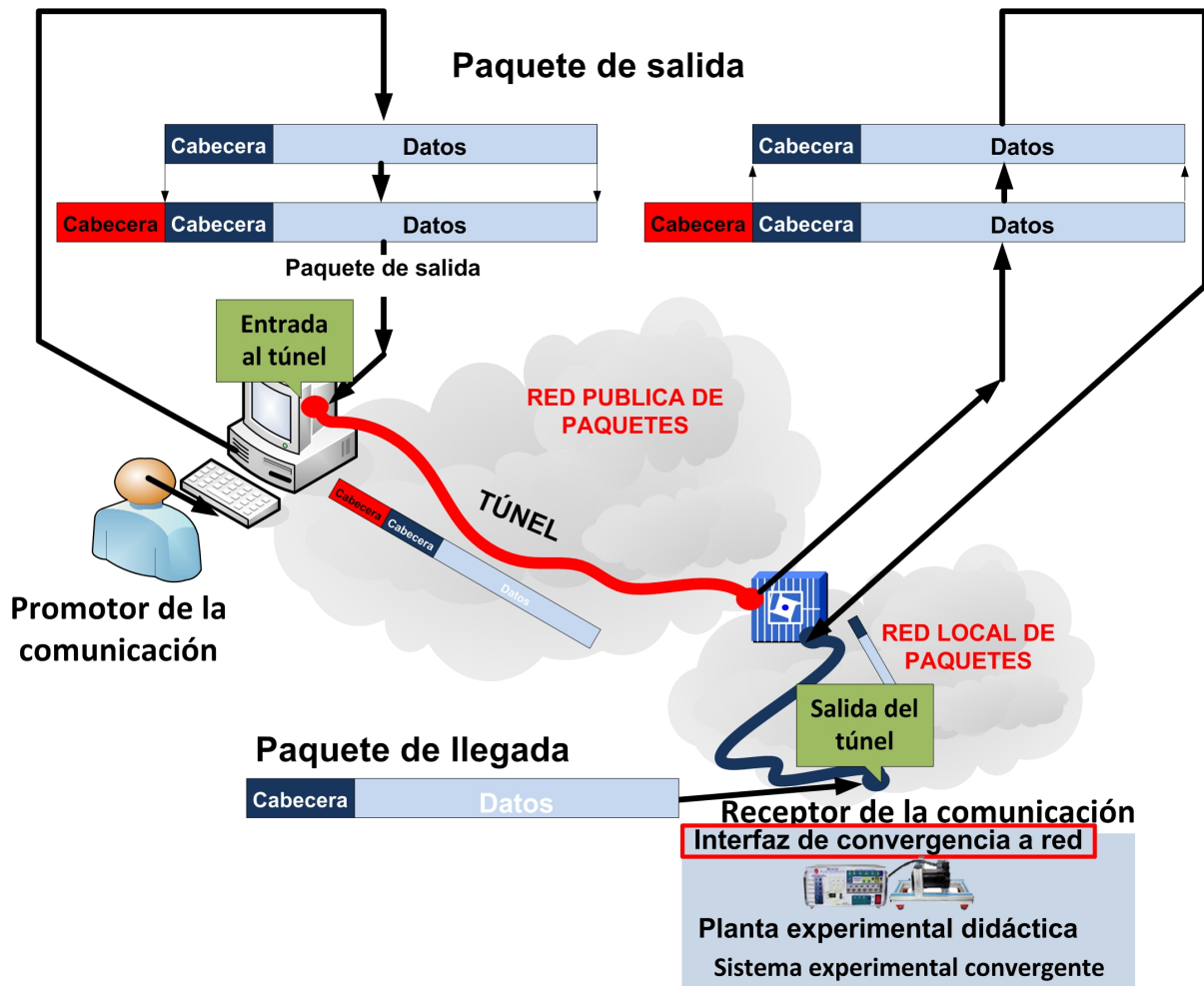


Figura 4.13: Túnel de comunicaciones a través de una red de paquetes.

Cada circuito de tránsito de información que se establezca a través de un túnel realizado sobre una red pública de paquetes queda totalmente definido por el punto de entrada al túnel en el nodo promotor de la comunicación y el punto de salida en la red interna del laboratorio, figura 4.13.

Por tanto, la funcionalidad de CONEXLAB se limita al establecimiento de un conjunto de túneles que faciliten las conexiones necesarias entre el PC del alumno y el sistema experimental. Además, cada una de estas conexiones queda perfectamente definida por el par formado por el punto de entrada en el PC del alumno y el punto de salida en la red interna. Así, una experiencia queda totalmente definida por el conjunto de puntos de entrada y salida al túnel. La base de datos de representación experimental está formada por el conjunto de puntos de entrada y salida del túnel que permiten la configuración de una experiencia.

El tratamiento de la información propuesto en el modelo tecnológico general y representado en la figura 4.13, permite que una experiencia de laboratorio pueda ser especificada de forma totalmente independiente de su naturaleza experimental. En efecto, su estructura consiste en una matriz de puntos de entrada y salida de los componentes necesarios para controlar el desarrollo de la práctica desde el sistema enseñanza-aprendizaje.

Considerando la estructura cliente-servidor de CONEXLAB indicada en la figura 4.12, el cliente (cliente de túneles CONEXLAB) se ejecuta en el PC del alumno y el servidor (servidor de túneles CONEXLAB) estará situado en la red pública de transporte y se denominará servidor SARLAB. Para establecer las conexiones de una experiencia, el cliente recibe la matriz de conexiones con los puntos de entrada a los túneles en el lado del cliente, e informa al servidor de los puntos de salida en el lado del servidor. De esta forma, la experiencia queda completamente definida y queda disponible el intercambio de información entre los subsistemas de interacción y de análisis, diseño y programación que se ejecutan en el PC del alumno con el sistema experimental situado en el laboratorio, figura 4.14.

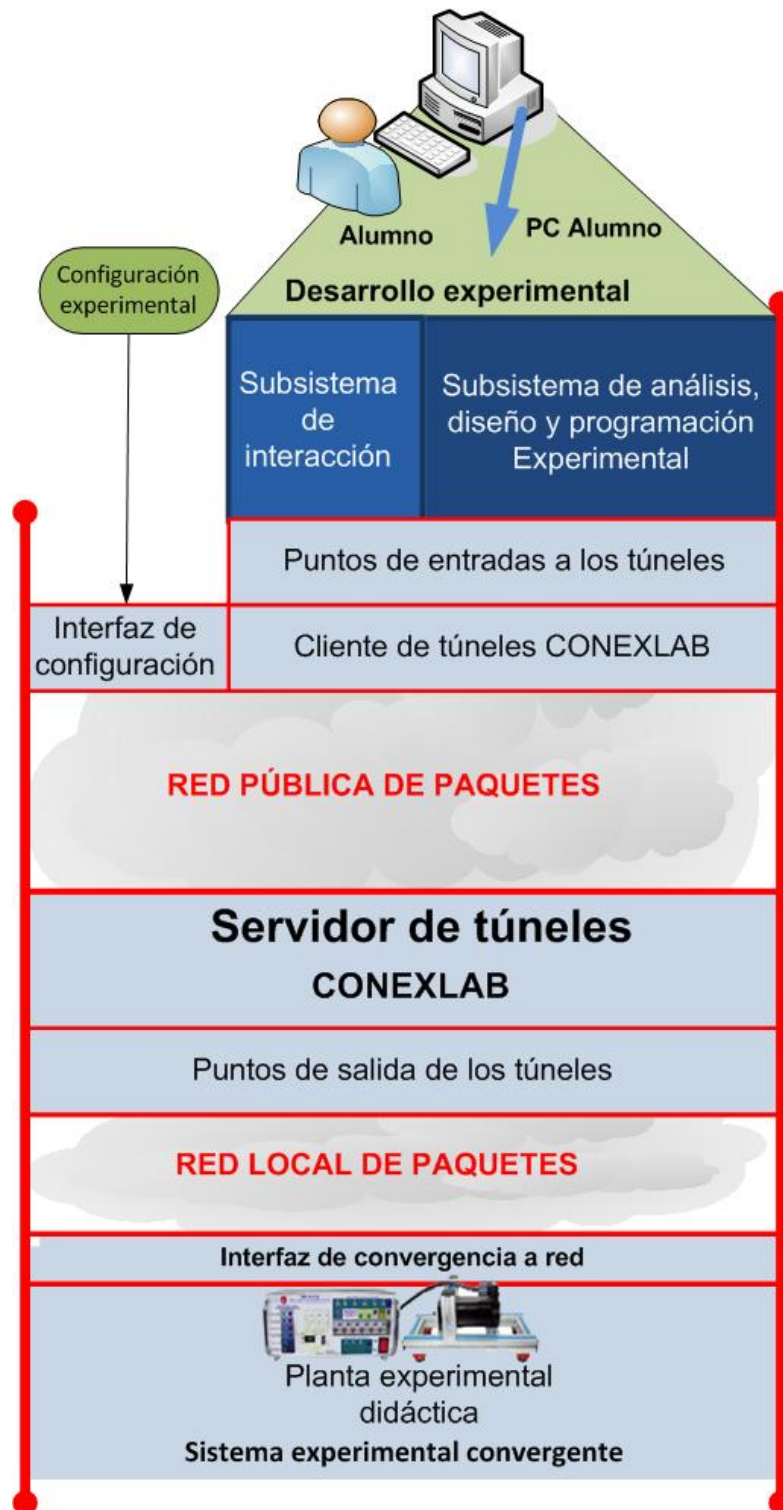


Figura 4.14: Estructura funcional de CONEXLAB.

En la figura 4.14 se muestra que CONEXLAB adopta la estructura de un servidor de túneles. Asimismo, en esta figura se ha representado la interfaz por la que el cliente de túneles CONEXLAB recibe la estructura matricial con la configuración de la

experiencia. Con esta información, el cliente conoce los puntos de entrada a los túneles y puede informar al servidor sobre los de salida. La estructura matricial de configuración de la experiencia, será enviada a la interfaz por el cliente DIGEXLAB, figura 4.12.

En la figura 4.15 se presenta un esquema de las interacciones producidas entre los distintos componentes del modelo y que se explica a continuación. DIGEXLAB transmite la estructura matricial de configuración al cliente de túneles CONEXLAB. Éste crea los puntos de entrada a los túneles e informa al servidor de túneles CONEXLAB sobre los de salida. En el momento en que los túneles quedan establecidos, se produce el flujo de información entre los subsistemas de interacción y de análisis, diseño y programación, y los recursos convergentes del sistema experimental (sensores, actuadores e interfaces de monitorización, configuración, programación, etc.). El servidor de túneles CONEXLAB controla además el fin de los túneles y verifica las condiciones de fin de experiencia que concluyen el tráfico de información.

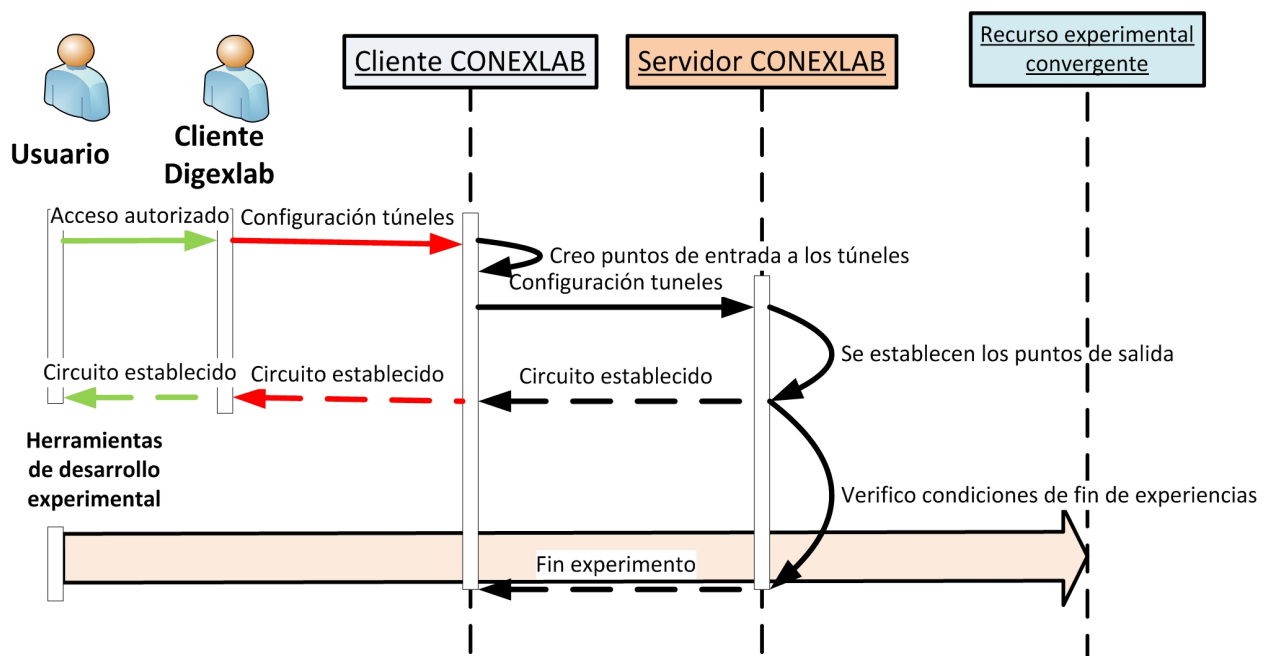


Figura 4.15: Diagrama de interacción de CONEXLAB.

4.5.3. DIGEXLAB

Una vez desarrollada la funcionalidad de CONEXLAB, se aborda en este apartado el desarrollo de DIGEXLAB, que se encarga de gestionar la información de configuración de las experiencias remotas y por tanto de administrar la base de datos de representación experimental.

La gestión de la información de configuración plantea dos perspectivas diferenciadas. Por un lado, las experiencias deben ser diseñadas en sus inicios y actualizadas en su explotación, lo que se traduce en el mantenimiento de la bases de datos de representación experimental, figura 4.16. Por otro lado, los usuarios tienen que acceder a las experiencias, lo que se traduce en validar su acceso y seleccionar la estructura matricial de configuración de la experiencia que debe transmitir a CONEXLAB, figura 4.16. Por último, DIGEXLAB controla el subsistema de alimentación eléctrica de la experiencia, desarrollado en el apartado 4.7.

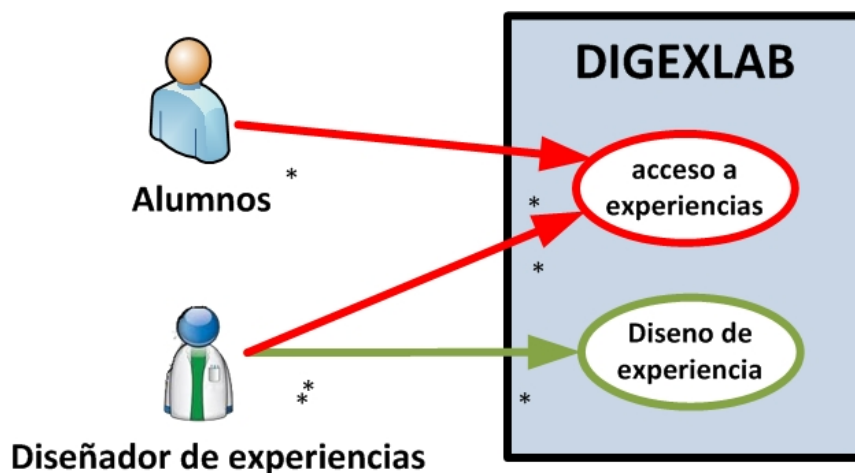


Figura 4.16: Casos de usos de DIGEXLAB.

En la figura 4.17 se presenta la parte correspondiente a DIGEXLAB mostrada en la figura 4.12. En ella se introducen las dos perspectivas de uso de DIGEXLAB: diseño de

experiencias y gestión de las mismas (validación del acceso, control de alimentación y configuración de CONEXLAB). Así, en la figura 4.17 se diferencian claramente estas dos perspectivas de uso. Por un lado, se muestra la funcionalidad de diseño de las experiencias, que se traduce en el mantenimiento de la base de datos de representación experimental: dar de alta una experiencia (definirla), darla de baja (eliminarla), y actualizarla en su explotación (modificarla). Esta funcionalidad debe traducirse en proporcionarle al diseñador de la experiencia una interfaz de acceso a las funciones de gestión de la base de datos de presentación experimental. Por otro lado, se muestra la funcionalidad de validar el acceso, seleccionar la estructura matricial de la configuración a transmitir a CONEXLAB, así como activar los dispositivos necesarios a través del subsistema de alimentación eléctrica.

La gestión del acceso resuelve el problema general de administrar el uso compartido de recursos conectados a una red de paquetes: qué usuarios pueden usar esos recursos, cuándo, cómo, e incluso si ese usuario puede realizar funciones de diseño. Para ello, se controla el acceso mediante el uso de perfiles. El perfil de usuario es una estructura de datos que lo identifica y le asigna los derechos de uso de los recursos compartidos.

En el modelo conceptual genérico definido en el capítulo 3, el lazo de gestión del aprendizaje se delegó en el LMS, que es el responsable de todas las cuestiones relacionadas con el proceso enseñanza-aprendizaje a través de la red. Si además se considera la visión sistémica propuesta en el modelo conceptual genérico, se deriva la necesidad de integrar todas las aplicaciones relacionadas con el lazo de desarrollo experimental como módulo del LMS.

Por tanto, el acceso a una experiencia debe ser iniciado en el LMS. En primer lugar, el alumno debe validar su perfil de acceso en el LMS para iniciar el lazo de gestión del aprendizaje. A continuación, podrá acceder a las experiencias a las que le da acceso su

perfil. Y en ese momento, se pone en marcha la gestión de acceso por parte de DIGEXLAB, que iniciará la configuración del lazo de desarrollo experimental.

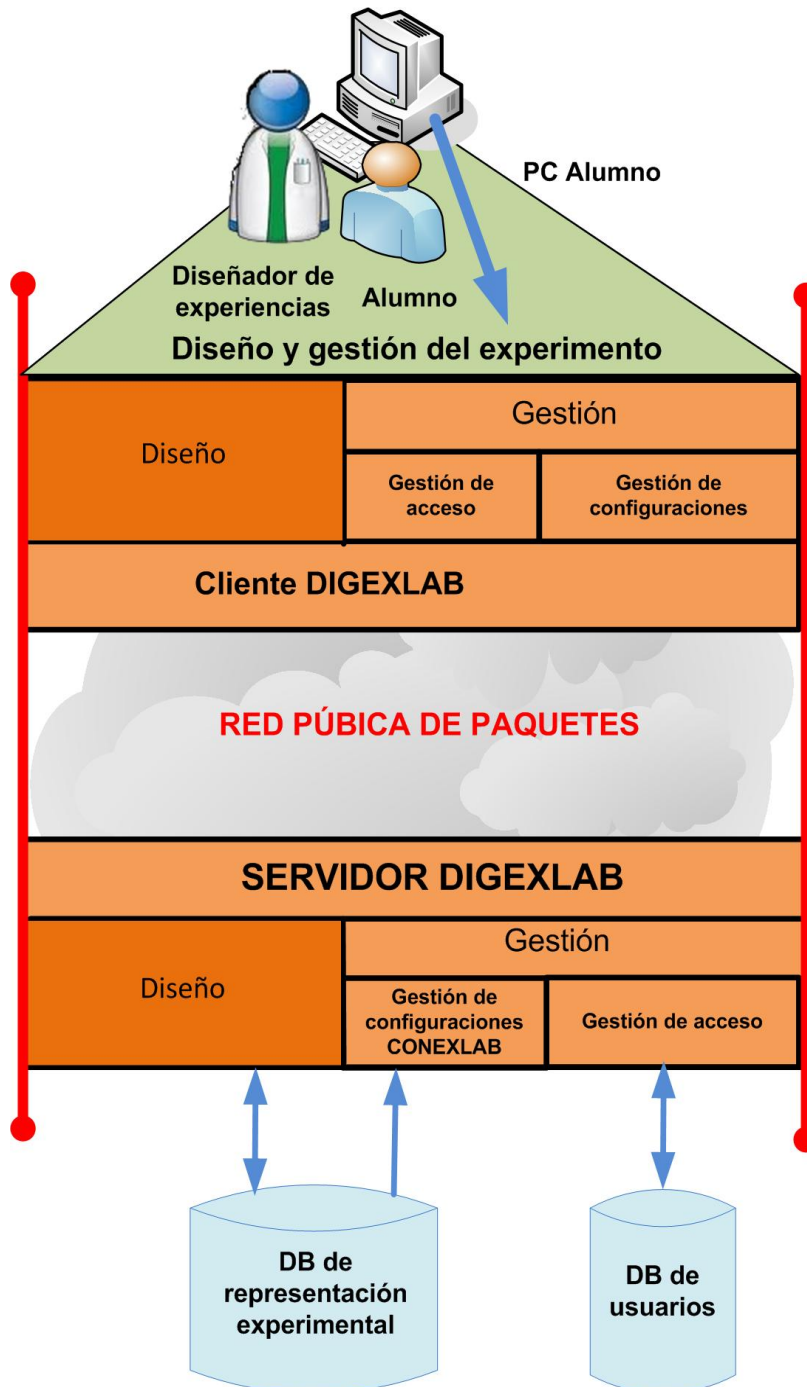


Figura 4.17: Estructura funcional de DIGEXLAB.

La gestión del acceso debe ser única aunque sea gestionada por dos aplicaciones (el LMS y DIGEXLAB). Por tanto, la responsabilidad de la misma se le asigna al LMS y

DIGEXLAB colabora con él para la validación y configuración del acceso a una experiencia concreta.

De esta forma, la validación del acceso a una experiencia controlada por SARLAB tiene dos etapas. Primero se inicia una sesión en el LMS, con lo que se activan las funciones de seguimiento de la traza de aprendizaje de la actividad y la correspondiente asignación de los perfiles de usuario. La segunda etapa comienza al iniciar la realización de una experiencia concreta y la gestiona DIGEXLAB, que aprovecha la información del perfil para gestionar los niveles de acceso (de uso o de diseño), figura 4.18.

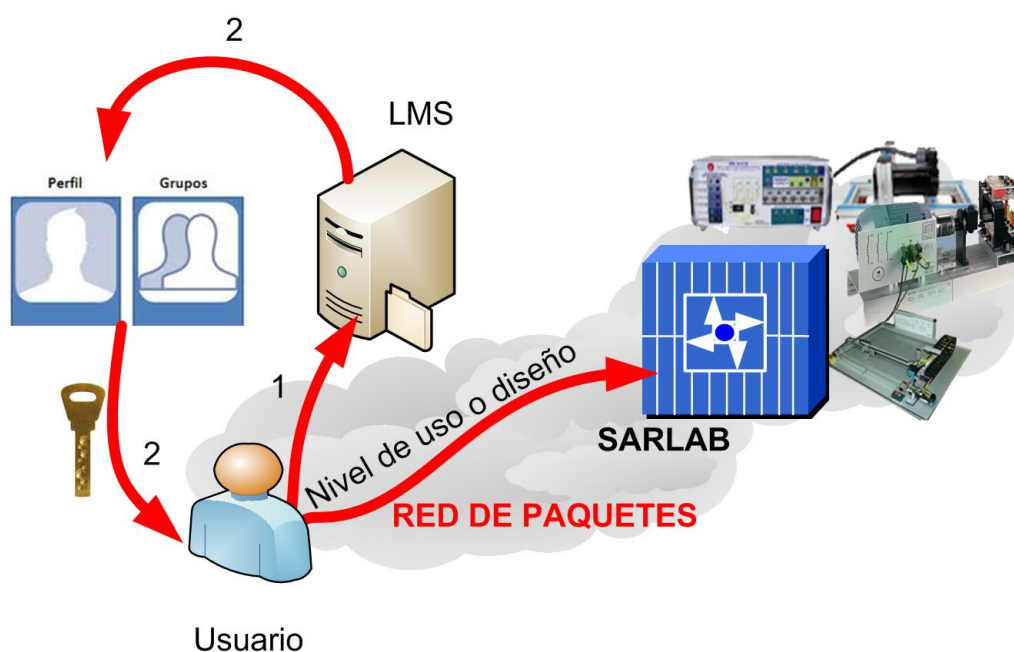


Figura 4.18: Gestión del acceso a un sistema experimental controlado por SARLAB.

En la figura 4.18 se muestra además como el acceso a SARLAB está condicionado por el acceso previo al LMS. Las dos etapas del proceso de validación de acceso permiten el uso distribuido de los laboratorios controlados por SARLAB, estando todos en la misma o en distintas instituciones y/o localizaciones geográficas, ofertándose todas las experiencias desde un único LMS. Esta configuración se denomina red de laboratorios distribuidos.

Además de las atribuciones contenidas en el perfil de usuario, para validar el acceso a una experiencia se debe tener en cuenta que, en general, los recursos experimentales no toleran el uso concurrente en el tiempo de más de un alumno. Es decir, dos alumnos no pueden hacer una práctica al mismo tiempo. Por tanto, el uso por parte de un alumno de un sistema experimental bloquea dicho sistema para el uso de los demás alumnos. Esto obliga a gestionar en qué instante y durante cuánto tiempo puede un alumno usar un recurso. Esta necesidad de gestión de recursos en el tiempo nos obliga a incluir en el esquema de componentes un nuevo módulo del LMS que se denomina *gestor de reservas*. Éste tendrá la función de ofertar a los alumnos la reserva de los sistemas experimentales por intervalos de tiempo concretos. El módulo gestor de reservas es en sí una aplicación que mantendrá una base de datos de la utilización de los sistemas experimentales. Por todo ello, la validación de acceso a una experiencia debe considerar, además de la autorización contenida en el perfil, la reserva temporal de uso.

El procedimiento normal de acceso a las experiencias remotas comienza cuando el usuario introduce en el cliente LMS su identificador y su *password*. A continuación, el servidor LMS identifica al usuario y valida su *password*. A partir de ahí, la sesión se ha iniciado y el LMS le muestra al alumno la oferta de aprendizaje correspondiente a su perfil. En el momento en que el usuario intenta iniciar una experiencia, se inicia la segunda parte de la validación del acceso, que le corresponde a DIGEXLAB. Previamente, el LMS comprueba el estado de reservas de la experiencia.

Una vez completada la validación por el LMS, se ejecuta el cliente DIGEXLAB, cuya primera tarea es contactar con el servidor DIGEXLAB, que confirma con el servidor LMS los requisitos de validación y procede a la configuración de CONEXLAB para acceder a la realización o el diseño de la experiencia, según la información del perfil (como usuario o diseñador).

A pesar de que, en general, los recursos experimentales no soportan el acceso concurrente, el hecho de que dos alumnos puedan acceder a una misma experiencia de forma colaborativa tiene especial interés en el desarrollo de capacidades genéricas, como el trabajo en equipo. Por tanto, DIGEXLAB debe considerar esta opción. En efecto, la idea más general de desarrollo colaborativo es la posibilidad de que un alumno con sesión abierta pueda invitar a cualquier otro con acceso al LMS. Este tipo de gestión de acceso deberá por tanto estar incluida en el esquema de control de validación de DIGEXLAB.

Como medida de seguridad ante la pérdida de conexión con el servidor LMS, DIGEXLAB dispone de una base de datos local para garantizar el acceso de administradores.

Por tanto, la validación de acceso de DIGEXLAB comprende tres procedimientos: la validación de acceso a la base de datos local, la confirmación de validación con el LMS y la validación de acceso colaborativo, figura 4.19.

En la figura 4.20 se presenta el diagrama de interacciones correspondiente al procedimiento de validación, que finaliza con la configuración de CONEXLAB mediante el paso de la matriz de conexiones.

La gestión del final de una experiencia puede ser causada por la finalización del periodo de reserva o porque se produce un corte en las comunicaciones. En el primer caso, el hecho de que las reservas sean de duración fija y conocida, y sabiendo el momento de acceso del usuario a la experiencia, se calcula el momento de su final. El tiempo de reserva restante al acceder a una experiencia, se denomina *tiempo de vida* de la misma. Esta tarea de desconexión la realiza el servidor CONEXLAB. Para que este procedimiento funcione, es necesaria la sincronización de los relojes del servidor LMS y del servidor SARLAB (que contiene a los servidores DIGEXLAB y CONEXLAB).

En el segundo caso, en el que la conexión se pierde por un corte en las comunicaciones, el servidor CONEXLAB detecta el corte y provoca la finalización de la experiencia.

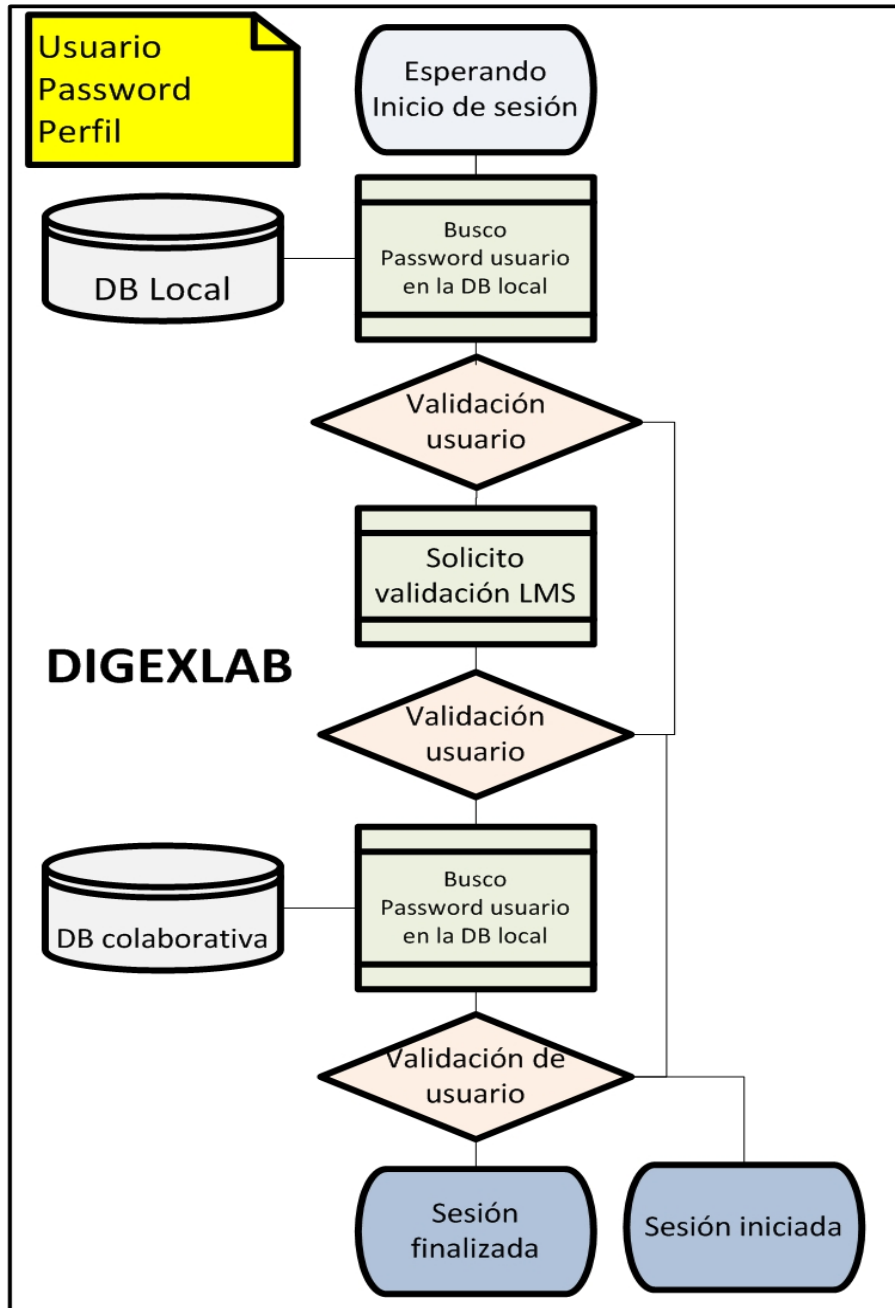


Figura 4.19: Diagrama SDL del protocolo genérico de validación de acceso a una experiencia desde DIGEXLAB.

Una vez desarrollados los componentes de SARLAB, las comunicaciones han quedado resueltas al nivel del modelo tecnológico general. Se plantea en los siguientes apartados el desarrollo del resto de los subsistemas del modelo conceptual general.

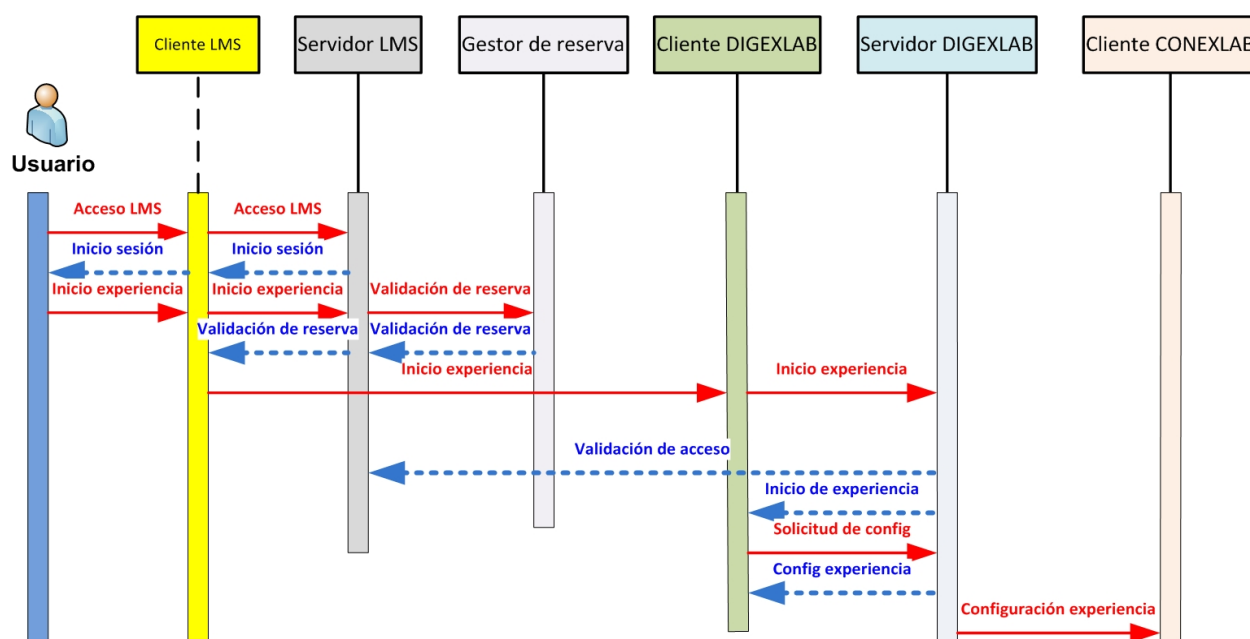


Figura 4.20: Diagrama de interacciones completo de realización de una experiencia remota.

4.5.4. Subsistema de interacción

El subsistema de interacción traslada las acciones del alumno sobre el experimento al sistema experimental. En general, esas acciones consistirán en pulsar una serie de botones, introducir valores, etc., según una determinada secuencia para controlar la evolución de la experiencia. Las acciones que el alumno realiza en su PC deben ser trasladadas a las correspondientes acciones sobre los recursos experimentales ubicados en el laboratorio. Por otra parte, la evolución producida en el escenario experimental debe ser percibida por el alumno, que deberá ser consciente de estar interactuando con elementos reales. El responsable de crear esta percepción experimental en el alumno es asimismo el subsistema de interacción.

El subsistema de interacción debe por tanto ser accesible para el usuario y estar integrado como un módulo del LMS. Por ello, el subsistema consistirá en una interfaz que le muestre al usuario la evolución del sistema experimental y le permita interactuar con él. Sin embargo esta interfaz depende de la naturaleza experimental de cada

experiencia y debe ser realizada a medida. Además, debe ser desarrollada por el diseñador de la experiencia, que no tiene porqué tener un alto conocimiento en lenguajes de desarrollo de aplicaciones informáticas. Por eso, la interfaz deberá ser desarrollada con una herramienta de desarrollo rápido, capaz de crear de manera asistida un GUI (*Graphical User Interface*) con el experimento en el que figuren objetos que puedan modelar las acciones del alumno sobre la experiencia, como botones, pulsadores, reguladores deslizantes, etc., y disponer de métodos de representación gráfica 2D, 3D y vídeo real, entre otros.

Dada la complejidad de algunos procesos, esta herramienta debe disponer de funcionalidades de tratamiento de datos para analizar los resultados que se obtengan directamente de la evolución de la experiencia. Por ejemplo, deberá integrar componentes de tratamiento matemático.

En resumen, esta herramienta de desarrollo rápido debe integrarse con facilidad en el LMS, aportar funcionalidades matemáticas y permitir la creación de GUI de forma fácil e intuitiva. Además, deberá ser de código abierto que permita la incorporación de funcionalidades para abarcar toda la gama de naturalezas experimentales.

4.5.5. Subsistema de análisis, diseño y programación

Si bien en muchos casos será posible que la herramienta de desarrollo rápido para implementar la interfaz del *subsistema de interacción* permita asumir funciones de análisis, diseño y programación, en otros casos el uso de ciertos dispositivos comerciales en el sistema experimental, hará necesario el uso de software específico para análisis de los datos que proporciona o para el diseño y la programación del dispositivo.

En este sentido, cada vez son más los sistemas de desarrollo y los dispositivos de medida que incorporan una interfaz de comunicaciones (no de red) de conexión a un computador. El uso natural de estos dispositivos consiste en la ejecución de una

herramienta en la que se desarrolla su programación y/o configuración o que simplemente analiza los resultados aportados por el dispositivo. Estas herramientas van desde los compiladores de microprocesadores, hasta el software de tratamiento de analizadores o los sistemas de representación de medidas, y son proporcionadas por los fabricantes. En muchos casos son de libre distribución o existen versiones para estudiante.

Pero en el caso de nuestro modelo, no se va a hacer un uso natural del dispositivo, sino que se va a acceder al mismo de forma remota, para lo que debe ser convergente. Para eso existen pasarelas de las interfaces de programación a interfaces de red, que permiten el acceso convergente de este tipo de dispositivos. Además, el software del fabricante deberá contar con el controlador necesario para acceder al dispositivo a través de la red. Estos requerimientos han sido ya incorporados por los fabricantes de muchos de estos dispositivos, lo que los convierte directamente en convergentes.

Por tanto, en nuestro modelo tecnológico general, el subsistema de análisis, diseño y programación puede ser una aplicación desarrollada por la herramienta rápida usada en la implementación del subsistema de interacción, o bien una aplicación creada por el fabricante de ciertos dispositivos comerciales que forman parte del sistema experimental.

4.6. Percepción experimental

La realización de prácticas desarrolla las capacidades experimentales del alumno y aumenta su motivación. En este sentido, el modelo tecnológico general debe conseguir que el alumno sea consciente de estar manejando dispositivos físicos a través de la red. Por ello, en este apartado se presentan los elementos que consiguen esa percepción experimental y las técnicas que la mejoran.

En todo ello, hay que tener en cuenta que al acceder de forma remota a una experiencia, el proceso de enseñanza aprendizaje se puede producir al ritmo de construcción del conocimiento de cada alumno (la experiencia está a su disposición las veces que lo necesite). Por otro lado, la evolución de la experiencia debe ser alterada de forma mínima al traducir las acciones entre la interfaz y el sistema experimental. En este sentido, tiene especial importancia la información visual y sonora cuando proceda aportada por el subsistema de transmisión multimedia.

Por tanto, la transmisión de audio y video crea en el alumno la percepción experimental. Sin embargo, su motivación puede ser mejorada con el uso de herramientas multimedia en el desarrollo del subsistema de interacción. Entre esas herramientas se encuentran, por ejemplo, los mundos virtuales y la realidad aumentada, que ya el usuario usa habitualmente en su desarrollo personal.

Los mundos virtuales son escenarios en los que el alumno puede asumir la personalidad de un avatar. La realidad aumentada es una tecnología multimedia de última generación que permite ampliar la realidad que percibe el alumno con elementos virtuales conectados a ésta, con objeto de conseguir una relación más amigable y espectacular que facilita la interpretación de la información visual que recibe el alumno de la experiencia remota.

El uso de estas tecnologías requiere que la herramienta de desarrollo rápido usada para crear el subsistema de interacción permita el uso, no solo de la información proporcionada por el subsistema de transmisión multimedia, sino del desarrollo de elementos virtuales.

4.7. Subsistema de alimentación eléctrica

La realización de una experiencia conlleva la puesta en marcha de la iluminación del experimento, así como de la alimentación de la planta, de los actuadores y sensores y de los sistemas de control, entre otros. La correcta gestión proporciona la eficiencia energética del sistema experimental. Por otro lado, la pérdida de control de algunos de estos dispositivos puede generar situaciones peligrosas como por ejemplo el sobrecalentamiento de un motor eléctrico. Por todo esto, es necesario el control de la alimentación eléctrica del sistema experimental mediante el denominado subsistema de alimentación eléctrica. Asimismo, ese control debe ser remoto. Además, teniendo en cuenta que los sistemas de comunicaciones no son totalmente confiables, el control de la alimentación eléctrica debe realizarse desde el lado del servidor SARLAB.

Teniendo en cuenta que DIGEXLAB es el encargado de poner en marcha la experiencia, debe ser el responsable de iniciar la alimentación del sistema experimental. Por otro lado, considerando que CONEXLAB es la funcionalidad que se encarga de mantener las comunicaciones, será el responsable de realizar la desconexión o bien por finalización del tiempo de vida o por pérdida de las comunicaciones.

De esta forma, el subsistema de alimentación eléctrica deberá situarse entre la capa de gestión de DIGEXLAB y el servidor CONEXLAB, figura 4.21, donde se representa el modelo tecnológico general propuesto.

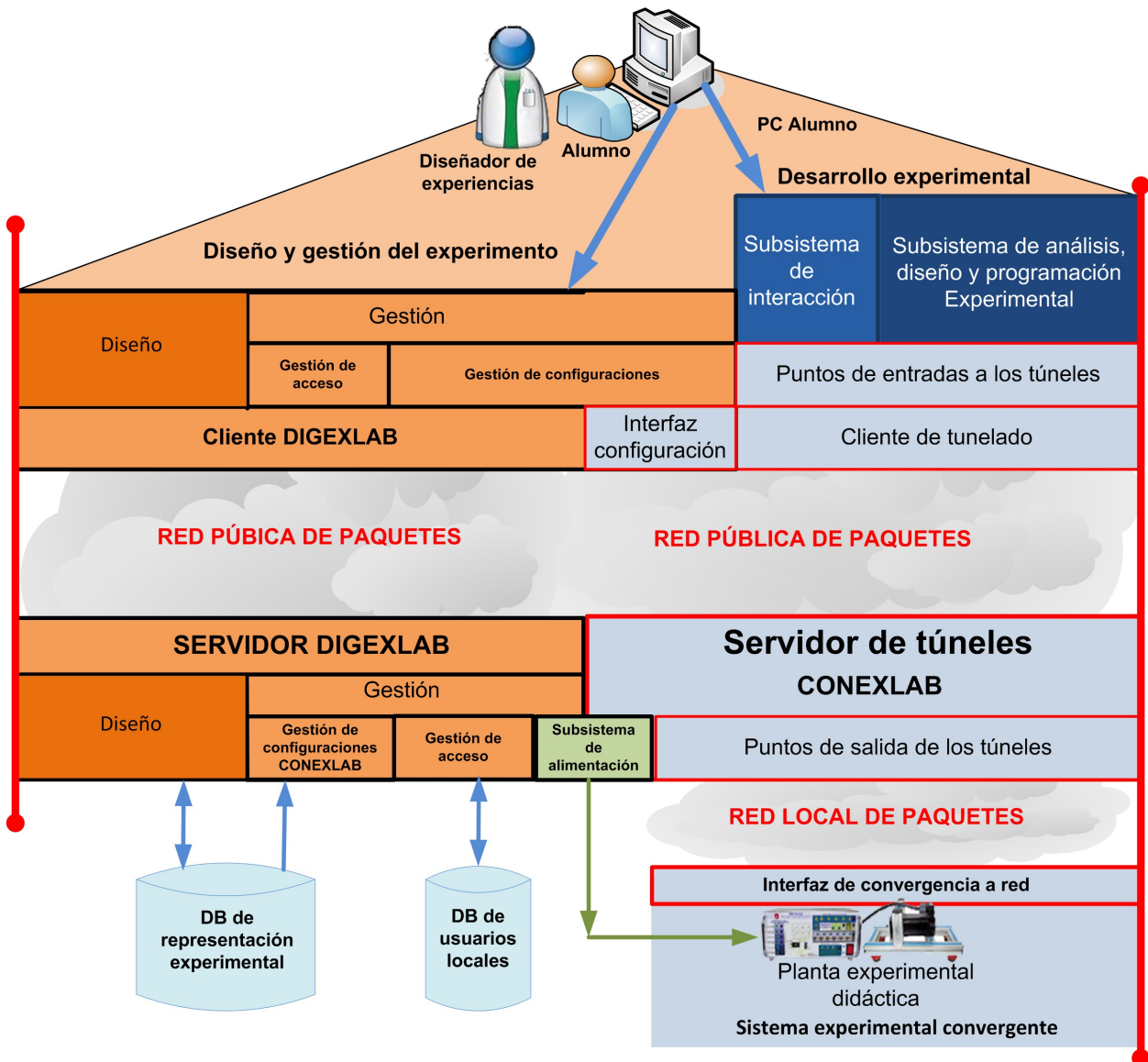


Figura 4.21: Modelo tecnológico general de sarlab.

4.8. Control topológico

La convergencia propuesta en nuestro modelo homogeneiza la estructura de la información intercambiada en la evolución de los procesos experimentales. Es decir, la red interna actúa como un bus de comunicaciones entre los componentes que conforman un sistema experimental. Los intercambios de información producidos en el progreso de una experiencia de laboratorio se producen por el intercambio de paquetes

a través de esta red. Por todo ello, los recursos no tienen que estar asociados de forma rígida a los sistemas experimentales, sino que pueden ser compartidos por varios de ellos. O sea, una planta experimental didáctica convergente puede ser controlada por varios dispositivos alternativamente, como por ejemplo PLCs, microcontroladores, FPGA, etc. Para ello es necesaria una aplicación que se encargue del control topológico (de conectar los flujos de información necesarios para constituir el sistema experimental convergente).

4.9. Movilidad experimental

El acceso remoto a sistemas experimentales no tiene porqué estar restringida a la realización de prácticas, sino que puede ser usada por el docente en otros tipos de metodologías didácticas como por ejemplo clases magistrales, seminarios, tutorías, etc. En definitiva, el modelo tecnológico general permite el acceso a los sistemas experimentales cuándo, cómo y dónde el usuario y/o diseñador lo necesite, lo que puede ser definido como *movilidad experimental*, propiedad que puede ser atribuida, según este razonamiento, a SARLAB.

4.10. Conclusiones

En este capítulo el modelo conceptual general desarrollado en el capítulo 3 ha evolucionado obteniéndose el modelo tecnológico general de acceso a experiencias de laboratorio remoto. Para ello se ha concretado el sistema de comunicaciones y se ha especificado la estructura del sistema de control de las mismas, que está constituido por SARLAB. En primer lugar se ha considerado la utilización de una red genérica de paquetes como la arquitectura de red posible, así como la estructura cliente-servidor para todas las funcionalidades necesarias. En este contexto, se ha introducido la estructura funcional de SARLAB que comprende las funcionalidades de CONEXLAB

(que se encarga de establecer las conexiones necesarias entre el sistema experimental y el usuario) y DIGEXLAB (que controla la base de datos de gestión experimental y colabora con el LMS en la validación del acceso del usuario). También se han especificado los subsistemas de interacción, de análisis, diseño y programación y de alimentación eléctrica. En todos los casos se han determinado las especificaciones que debe cumplir cada uno sin entrar en las restricciones concretas que introducen tecnologías determinadas (la implementación final en tecnologías concretas se presenta en el capítulo siguiente). Por último, en este capítulo se ha hecho referencia a conceptos importantes en el modelo como la percepción experimental por parte del usuario, la convergencia a red, el control topológico o la movilidad experimental.

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

El enfoque utilizado a lo largo de los capítulos anteriores para abordar el problema del acceso remoto a experiencias de laboratorio de forma global, se ha concretado en el capítulo 3 en la definición del modelo conceptual general. Este modelo ha sido especificado en el capítulo 4 para obtener el modelo tecnológico general, que aporta una visión más constructiva del problema y más cercana al objetivo final del presente capítulo y el siguiente: conseguir una implementación concreta y operativa utilizando el estado actual de la tecnología.

El modelo conceptual general presentado en el capítulo 3 proporciona una visión cualitativa de la dinámica de las relaciones necesarias para la realización y el diseño de una experiencia de laboratorio de forma remota. En la evolución del modelo, la inclusión de un LMS en el modelo conceptual permite delegar todas las funciones necesarias para el desarrollo del aprendizaje a través de red en este componente.

El modelo tecnológico general constituye una idealización tecnológica. Para llegar a él, en el capítulo 4 se ha concretado la red definida en el modelo conceptual en una red de paquetes genérica (idealizada), en la que se han introducido los condicionantes básicos que esta técnica de comunicaciones impone al tratamiento de la información.

Por otro lado, el modelo tecnológico general plantea la estructura más genérica que puede adoptar el sistema experimental convergente. Además, en el capítulo 4 se ha profundizado en los aspectos físico y funcional de SARLAB, sistema introducido en el modelo conceptual para la gestión de las comunicaciones en el lazo de desarrollo experimental. Finalmente en el capítulo 4 se han analizado, de forma genérica, los aspectos funcionales del resto de subsistemas presentes en el modelo conceptual (subsistema de interacción, subsistema de análisis, diseño y programación, subsistema de alimentación).

5.1. Introducción

Para llegar a un sistema que sea explotable, es necesaria la implementación física del modelo tecnológico general del capítulo 4 con los recursos que en cada momento aporte el estado de la tecnología. En este sentido, en este capítulo se presenta el diseño del esquema general de implementación física del modelo (elementos hardware del sistema y aspectos de integración en la infraestructura de red) y en el capítulo siguiente se aborda el desarrollo de las funcionalidades y sus correspondientes aplicaciones informáticas (software). El diseño del esquema general de construcción física del modelo plantea la elección o implementación (en caso necesario) de todos los componentes físicos comprendidos en el modelo tecnológico general. Por otro lado, los aspectos relacionados con la elección o el desarrollo (en caso necesario) de las aplicaciones a ejecutar por los componentes físicos del hardware, con el objeto de realizar las funcionalidades a ellos asociadas, se desarrollan en el siguiente capítulo.

Entre los aspectos relacionados con el hardware están la elección de las redes de paquetes (pública y local), la elección de los servidores y sus sistemas operativos, la elección del servidor LMS y del servidor SARLAB, la elección o implementación de los dispositivos necesarios para conseguir la convergencia de los sistemas experimentales, y

la elección o implementación de los dispositivos de control del subsistema de alimentación así como del servidor multimedia.

Asimismo, para la elaboración de la solución propuesta habrá que considerar los condicionantes que se han ido introduciendo en los capítulos 3 y 4, así como otros que son consecuencia directa de las limitaciones que aparecen al sustituir los componentes idealizados por componentes reales. El conjunto de todos ellos puede organizarse como se indica a continuación:

- Condicionantes derivados de la sustitución de los componentes idealizados del modelo tecnológico general por elementos físicos reales. Por ejemplo, la latencia de la red, el tiempo máximo de funcionamiento de los dispositivos, etc.
- Condicionantes derivados de la necesidad de que las herramientas de diseño se manejen con facilidad, para que los creadores de experiencias no precisen de una elevada formación en tecnologías de comunicaciones y en lenguajes de programación.
- Condicionantes derivados de la viabilidad económica. Para propiciar el uso masivo del modelo es necesario que su implementación tenga costes asumibles. Esto llevará al uso de recursos de bajo coste como el software libre, las plataformas abiertas de hardware, las librerías de alto nivel que faciliten el desarrollo rápido, el uso de protocolos estándar de comunicaciones que facilitan la integración de componentes, o la reutilización de software.

5.2. Implementación de la estructura física de comunicaciones

En el capítulo 4 se planteó de manera genérica la estructura física del sistema de comunicaciones con componentes ideales para implementar el lazo de desarrollo

experimental. En esta estructura se diferenciaban dos redes, una pública y otra de ámbito local, separadas por el servidor SARLAB. El modelo incluía además un servidor LMS responsable del lazo de la gestión del aprendizaje.

En este apartado se plantea por tanto la concreción tecnológica de cada uno de estos componentes: red pública, red local y servidores SARLAB y LMS. Como las experiencias con acceso remoto se instalan usualmente en universidades y centros de investigación, en su construcción física habrá que considerar los condicionantes derivados de su integración en las infraestructuras de comunicaciones de estas instituciones.

5.2.1. Elección de la red pública de datos

La primera tarea en la construcción de la arquitectura física del modelo tecnológico general es la elección de la red pública, que será en sí un factor determinante en toda la tecnología de red para la arquitectura física del sistema.

Del análisis realizado en el Capítulo 2 acerca del conjunto de experiencias con acceso remoto publicadas en la literatura, se observa que la práctica totalidad de ellas facilitan el acceso remoto desde la red pública Internet. Además, actualmente Internet es la única red que garantiza un carácter global. Esta red de comunicaciones ha transformado los usos y costumbres de la sociedad hasta el punto de que cualquier actividad profesional o personal tiene una importante dependencia de Internet. Esta red es parte fundamental en el ámbito educativo, socio-económico y del ocio, incluso en el de las relaciones sociales. Por tanto, debe ser esta red la que se elija para implementar el medio de comunicaciones del modelo.

La elección de Internet como la red pública de paquetes del modelo, impone en la estructura de comunicaciones el uso de una familia concreta de protocolos que regirá el flujo de datos: la familia de protocolos TCP/IP. De igual forma es importante destacar

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

que TCP/IP se ha convertido en el estándar de cualquier tipo de red porque para poder conectarse a Internet, las redes locales han tenido que adoptar este mismo protocolo. En este sentido, el protocolo de flujo de datos usado para la implementación de la red local del modelo será también TCP/IP. La consecuencia de eso es que los paquetes que permiten el desarrollo de una experiencia remota viajaran desde el PC del alumno al sistema experimental y viceversa encapsulados en esta tecnología.

En los siguientes apartados se estudian las características de la familia de protocolos TPC/IP que permiten la concreción de las funcionalidades del modelo tecnológico general: la forma en la que se definen las conexiones entre los clientes y los servidores, las políticas de seguridad que deben superar los túneles para conseguir que no se bloquee el tránsito de información, y la forma de incluir el esquema físico de SARLAB como un servicio más de los ofertados desde las universidades y/o centros de investigación.

5.2.1.1. Definición de una conexión TCP/IP

En una red con tecnología TCP/IP, todo nodo de comunicaciones queda identificado por una dirección denominada IP. Una dirección IP es un número binario que permite identificar unívocamente un equipo dentro de una red. Existen dos versiones de direcciones IP: la versión 4 con una longitud de 32 bits y la versión 6 que extiende el número de bits a 128. La versión elegida es la 4 porque es la más utilizada actualmente tanto en las redes locales como en las redes públicas de los operadores de comunicaciones, en las universidades y en los centros de investigación. Para facilitar la representación de una dirección IP se suele utilizar la denominada notación de punto, que convierte el número binario en cuatro números decimales separados por puntos. Cada uno de esos números decimales representa la conversión a decimal de los 32 bits tomados de 8 en 8. Desde el punto de vista del acceso a Internet, las direcciones se pueden dividir en dos tipos: las públicas y las privadas. Las direcciones públicas son

válidas para que los paquetes puedan circular por Internet y las direcciones privadas son rangos de direcciones destinadas a las redes privadas, no validadas para su uso en la red pública.

Por tanto, las direcciones asignadas a los sistemas experimentales convergentes deben ser direcciones IP privadas y SARLAB debe tener al menos una dirección IP pública en la interfaz que lo conecta a Internet, y una dirección en el rango de direcciones privadas correspondiente a la red del laboratorio remoto, por la que estará en contacto con los sistemas experimentales.

Otra característica importante del direccionamiento IP en la estructura de comunicaciones del modelo es que todos los sistemas conectados a red a través de la familia de protocolos TCP/IP disponen de una interfaz interna denominada bucle local. Esta interfaz permite acceder a una red interna con un rango de direcciones 127.0.0.0 /8, de clase A. Este rango habilita la comunicación de aplicaciones cliente-servidor en equipos sin interfaz de red física. De esta forma, el extremo del túnel situado en el PC del alumno puede ser identificado independientemente de la forma en que el alumno esté conectado a Internet. Es decir, todos los sistemas con protocolo TCP/IP aceptan conexiones por la dirección 127.0.0.1, entre otras del rango 127.0.0.0/8. Por tanto, los túneles que definen una experiencia siempre pueden utilizar la dirección 127.0.0.1 como punto de entrada y solo será necesario definir el punto de salida en el laboratorio.

Por otro lado, la dirección IP identifica el nodo origen o destino de una comunicación, pero en cada nodo se pueden estar realizando múltiples comunicaciones. Para identificar el destino de una comunicación en cada nodo, la tecnología TCP/IP añade un parámetro de direccionamiento que se denomina puerto. Es decir, en una comunicación, la dirección IP determina el equipo y el puerto identifica la aplicación dentro del equipo que participa en la comunicación. Por tanto, toda comunicación establecida mediante el uso del protocolo TCP/IP queda identificada por los pares

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

<IP,puerto> de los dos extremos de la comunicación. El par formado por la dirección IP y el puerto se denomina *socket* de la comunicación.

De este modo, en la tecnología TCP/IP los túneles gestionados por SARLAB quedan especificados por el *socket* de entrada al túnel, situado en el PC del alumno y el *socket* de salida en el sistema experimental convergente, situado en el laboratorio. Asimismo, siempre se puede establecer la IP del extremo del túnel situada en el PC del alumno utilizando la dirección del bucle local 127.0.0.1.

El establecimiento de una conexión TCP/IP queda finalmente definida así: se inicia en una aplicación cliente que se ejecuta en un sistema a través de la creación de un *socket* formado por la dirección IP del sistema, y un puerto asignado de forma dinámica entre los puertos libres de uso del sistema comprendido entre 1025 y 65535. En el otro extremo de la comunicación se encuentra el servidor correspondiente a la aplicación cliente, localizado por su correspondiente *socket*, formado por su dirección IP y un puerto que debe ser fijo y conocido, comprendido entre 0 y 1024.

5.2.1.2. Políticas de seguridad en Internet

Las restricciones impuestas por las políticas de seguridad establecidas en el ámbito de Internet afectan al tráfico de datos necesario en el desarrollo remoto de una experiencia. Por tanto, estas políticas son un aspecto importante a considerar en el desarrollo de SARLAB.

El modelo tecnológico general minimiza el impacto de estas políticas mediante el uso de túneles que encapsulan bajo un mismo protocolo todo el tráfico cursado por la red pública. El establecimiento del túnel garantiza que el tráfico es amigable en el tránsito por la red pública desde el PC del alumno hasta SARLAB y viceversa, independientemente del protocolo usado en el intercambio de información por el túnel.

Internet es una red de redes y en consecuencia está formada por un conjunto de sistemas autónomos con tecnología de red TCP/IP interconectados entre sí. La seguridad en Internet es un campo muy especializado y complejo que afecta a la configuración de las aplicaciones, sistemas operativos, electrónica de red, cortafuegos, detectores de intruso, etc. En este sentido, es necesario para la implementación del modelo identificar las situaciones en que el tráfico de los túneles puede quedar bloqueado. Por tanto este apartado se centra en el proceso de filtrado que sufren los paquetes en el tránsito por Internet.

A efectos de seguridad, el tráfico por Internet queda identificado por los puertos origen y destino del cliente y el servidor que interviene en una comunicación. En este sentido, entre los puertos que identifican los servicios más utilizados en Internet, se pueden citar el puerto 80 asociado al servicio web, el puerto 25 asociado al servicio de envío de correo (SMTP), los puertos 20 y 21 asociados al servicio FTP, etc. Esto es, un paquete TCP/IP que viaja buscando el puerto 80 será considerado por la red como un acceso a un servidor Web, y un paquete que viaje por la red desde un puerto 23 será considerado como la contestación de un servidor telnet a la petición de un cliente.

Por tanto, el objetivo en la implementación del modelo es localizar el puerto por el que debe escuchar SARLAB para que su tráfico sea considerado amigable en todo Internet. Para eso, se analizarán los condicionantes al tráfico que se pueden dar en el escenario más general de desarrollo de una experiencia. Esto comprende las formas en la que un estudiante puede acceder a Internet, la forma en las que las experiencias son ofertadas desde las redes de las universidades o centros de investigación, y finalmente la forma en que son tratadas las comunicaciones en las redes de tránsito.

Las restricciones al tráfico se pueden producir en algún punto del tránsito por la red, desde que el paquete sale del PC del alumno hasta que llega a la experiencia situada en algún laboratorio de una universidad o viceversa. Se estudiarán por tanto las

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

restricciones que se pueden producir en el acceso, en el tránsito o en la llegada a la experiencia. Para esto será necesario conocer también la estructura básica de interconexión de los sistemas autónomos que conforman el núcleo de Internet.

5.2.1.2.1. Restricciones en el acceso del usuario a Internet

El acceso de los usuarios a Internet se puede dar en dos situaciones distintas. La conexión se puede establecer a través de un operador de comunicaciones o a través de un segmento de red de una entidad privada o pública, como por ejemplo la red de alumnos de una universidad u otra institución educativa, la red wifi pública de un ayuntamiento o diputación, la red wifi pública de un centro comercial, la red privada de una empresa, etc. En el primer caso, el operador de comunicaciones no suele imponer restricciones al acceso de datos de los servicios habituales como WEB, FTP, SMTP, POP3, IMAP, etc. Únicamente en algunos casos puede limitar el acceso a servicios de comunicaciones que el operador tarifica aparte como vozIP, videoconferencia, etc.

Así, el escenario más desfavorable para el acceso se da desde una red que tiene una política de seguridad restrictiva como las redes abiertas de los campus, las redes de empresas, o los centros públicos de acceso. Estas redes suelen limitar las conexiones básicamente al uso de la tecnología web. Es decir, a los puertos 80 y 443, porque si se limita también el acceso a estos dos puertos la red no tiene utilidad en Internet. Por tanto, esta es la condición más restrictiva para el acceso a la red y son estos dos puertos los que hay que utilizar para la conexión con el servidor SARLAB.

5.2.1.2.2. Restricciones en las redes de tránsito

La red de los operadores de comunicaciones tampoco restringe el tránsito de ningún protocolo. En todo caso, solo se esperan restricciones en los servicios que son tarificados aparte por dicho proveedor. De hecho, la red de cada operador se conecta al resto del mundo través de los enlaces internacionales a los que no se aplica restricción ninguna.

En esta línea, todos los sistemas autónomos de un mismo país están interconectados en los denominados puntos neutros de la red, IXP (*Internet Exchange Point*) o NAP (*Network Access Point*), por lo que el intercambio de información entre ellos pasa siempre por el punto neutro, a falta del cual, los operadores estarían conectados mediante los enlaces internacionales. De esta forma, la existencia del punto neutro reduce el tránsito de información por los enlaces internacionales y por tanto el coste promedio del servicio del operador (que está principalmente constituido por el mantenimiento de los enlaces internacionales). Las conexiones implicadas en esta estructura no suelen filtrar el tráfico de tránsito, limitándose el filtrado a la protección de acceso a los equipos de comunicaciones de los operadores.

Paralelamente a esta red de operadores de comunicaciones, existe una red científico-educativo de nivel mundial a la que están conectadas las de las universidades y centros de investigación. En Europa esta red se encuentra soportada por el proyecto GÉANT, que engloba la red paneuropea de investigación y educación que interconecta los centros de investigación y educación europeos (Redes Nacionales de Europa NRENs). La red europea tiene troncales (conexiones) con todo el mundo, Fig. 5.1, y su ancho de

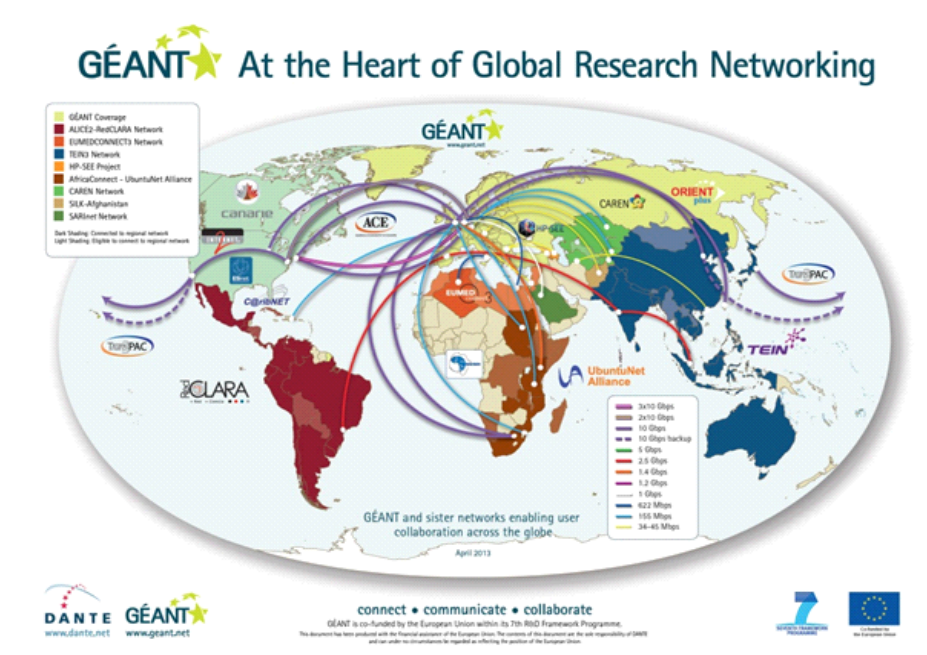


Figura 5.1: Topología internacional del proyecto GÉANT.

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

banda está en continua evolución. Actualmente está fijando en sus conexiones entre 10 y 40 Gbps, y mantiene un estado permanente de monitorización para detectar el estado de saturación de las comunicaciones.

GÉANT es la encargada de conectar las redes científicas europeas, Fig. 5.2, que engloban más de 50 millones de usuarios en 10.000 instituciones de toda Europa. Funciona a velocidades de hasta 500 Gbps, y ofrece una cobertura geográfica muy amplia. GÉANT mantiene la red de investigación más avanzada del mundo.

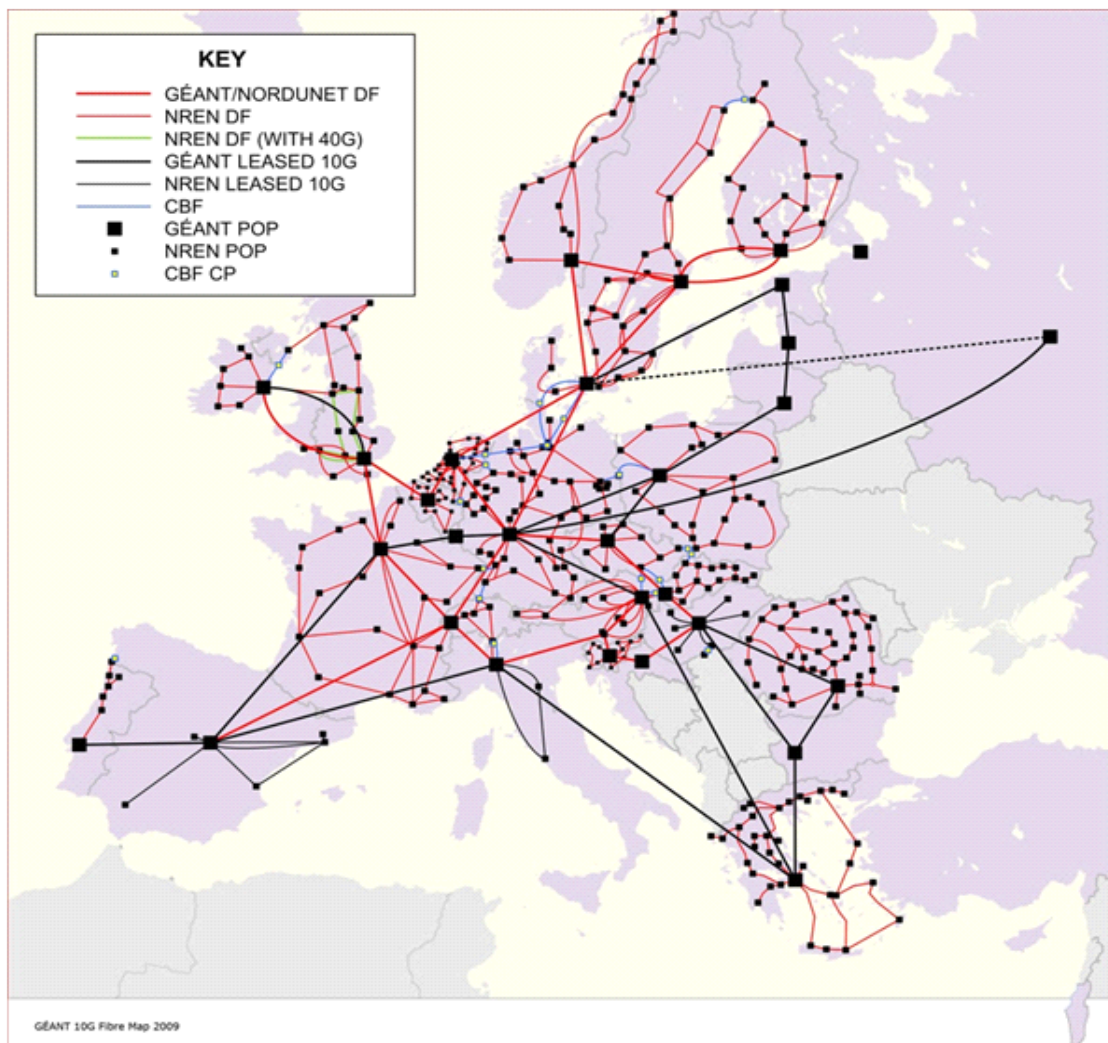


Figura 5.2: Topología de red europea.

En el caso concreto de nuestro país, Rediris es la encargada del desarrollo de la red científica, interconectando las universidades y centros de investigación a través de enlaces que se establecen con cada comunidad autónoma, figura 5.3. Esta red constituye la red académica y de investigación española, y proporciona servicios avanzados de comunicaciones a la comunidad científica y universitaria nacional. Está financiada por el Ministerio de Economía y Competitividad, e incluida en su mapa de Instalaciones Científico-Técnicas Singulares (ICTS). La gestiona directamente la entidad pública empresarial Red.es, del Ministerio de Industria, Energía y Turismo. RedIRIS cuenta con más de 450 instituciones afiliadas, principalmente universidades y centros públicos de investigación.

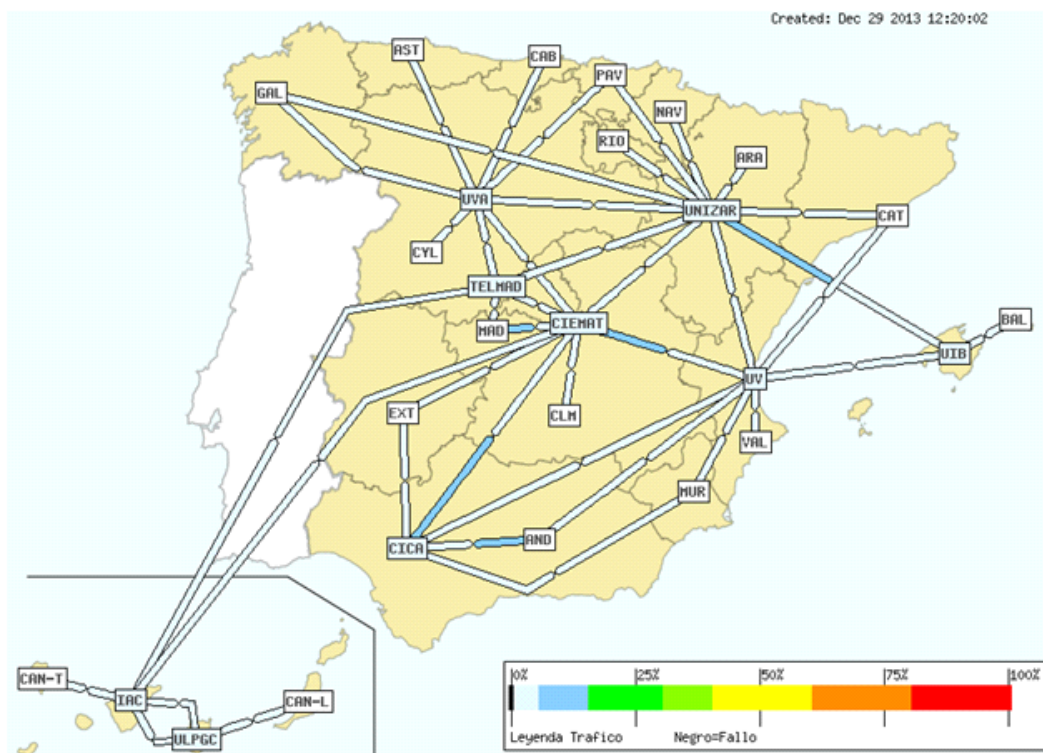


Figura 5.3: Topología de Rediris.

Finalmente, en Andalucía, la Red Informática Científica de Andalucía RICA es la encargada de interconectar todas las universidades y centros de investigación de la

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

región. En la figura 5.4 se representa un esquema de las conexiones entre las universidades de las ocho provincias andaluzas.

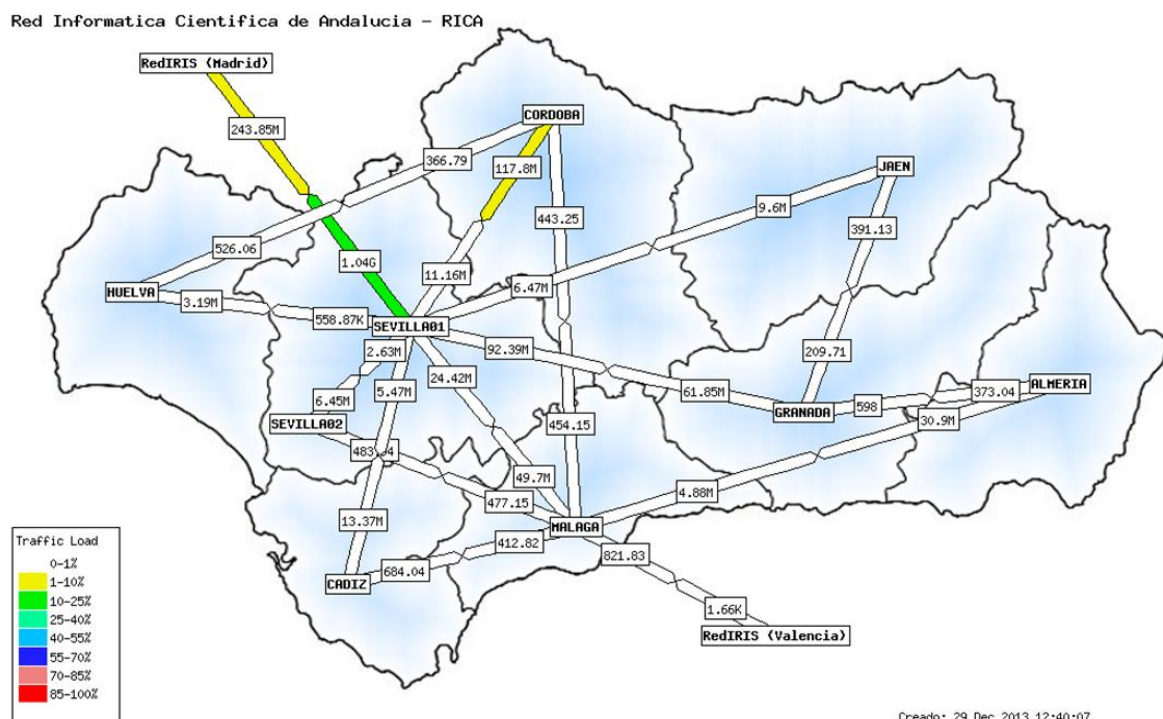


Figura 5.4: Red RICA

En resumen, las redes de tránsito no imponen ningún tipo de filtrado al paso de información, aunque sí al acceso a la electrónica de red que constituye la infraestructura de las mismas para su seguridad. Esto garantiza que las comunicaciones del modelo no pueden ser bloqueadas en este tramo de Internet.

5.2.1.2.3. Restricciones de acceso en universidades y centros de investigación

Las experiencias involucradas en el modelo propuesto se ofertan en Internet a través de la red científica de la comunidad en la que se ubique. La red propia de cada universidad constituye en sí misma un sistema autónomo dentro de aquella, y como tal tiene sus propias políticas de seguridad, que afectan tanto a las redes internas de acceso

como a los servicios ofertados. Este sistema autónomo se gestiona habitualmente en un servicio centralizado de informática.

Para garantizar la seguridad del tráfico de entrada o salida en el sistema autónomo de la universidad o centro de investigación, la arquitectura de red incorpora un cortafuego encargado del establecimiento de sus políticas de seguridad, figura 5.5. El cortafuego es el encargado además de gestionar un segmento de red donde se sitúan todos los servidores de la institución que ofertan algún servicio en Internet. En este segmento el cortafuegos sólo permite el tráfico de acceso a un servidor del segmento por los puertos de los servicios que son ofertados. Este segmento suele estar dotado de algunas medidas adicionales de seguridad, como son los detectores de intrusos o los sistemas de detección y eliminación de virus informáticos. Recibe el nombre de zona desmilitarizada o DMZ, y es desde donde el servidor SARLAB debe ofertar el acceso a experiencias a través del puerto elegido (80 o 443).

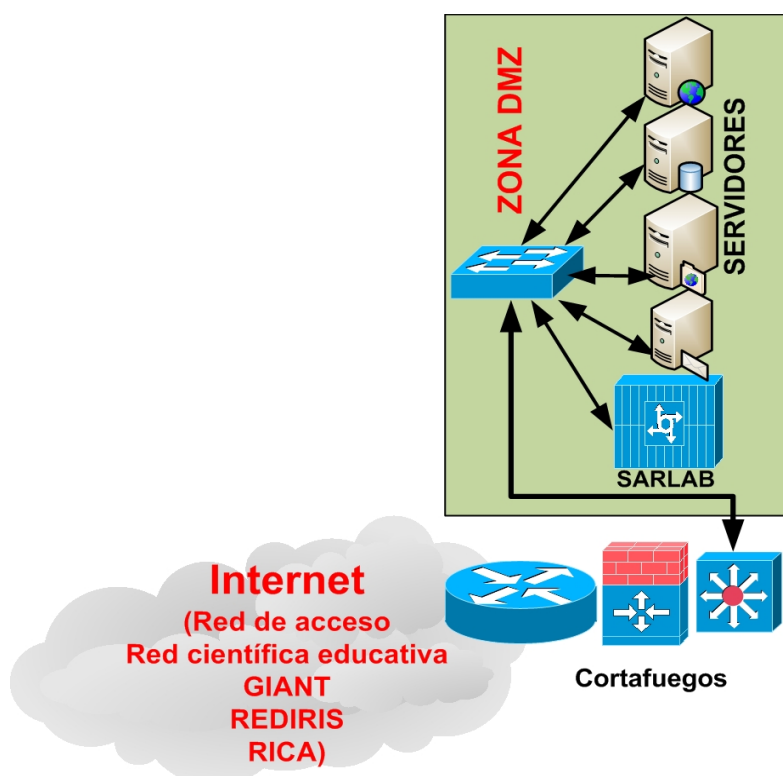


Figura 5.5: Estructura de seguridad de una institución científica.

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

En resumen, el análisis de las políticas de seguridad establecidas en el ámbito de Internet establece dos condiciones para la implementación del modelo:

- ✓ SARLAB ofertará sus servicios a través del puerto 80 o del puerto 443.
- ✓ SARLAB tendrá su interfaz de red pública situada en la DMZ de la institución.

5.2.1.3. Integración de SARLAB en la red de una institución científica

En principio, para que SARLAB sea el servidor desde el que se ofertan todas las experiencias de acceso remoto de una misma institución, sería necesario disponer de la infraestructura básica de laboratorios de un amplio número de disciplinas (física, química, resistencia de materiales, automática, electrónica, maquinas eléctricas, etc.) en una única instalación. Para evitar la complejidad de una medida como ésta, se plantea en el modelo propuesto que la red local a la que SARLAB está conectada, sea la que esté accesible desde cada laboratorio de la institución que oferta experiencias. Esto garantiza la existencia de la infraestructura básica necesaria para cada experiencia según su naturaleza experimental, ubicada en el laboratorio del área de conocimiento correspondiente.

La accesibilidad de la red local de SARLAB desde un conjunto de laboratorios distribuidos por las instalaciones de la institución, obliga a estudiar las tecnologías de las redes locales de las instituciones que se denominarán a partir de ahora *redes de campus*.

5.2.1.3.1. Red de campus de una institución

La tecnología utilizada en la conexión de las distintas dependencias de una institución es, salvo escasas excepciones, la tecnología Ethernet, que se ha convertido en la tecnología hegemónica para el desarrollo de redes locales. Prácticamente el 100% de

las redes de ámbito LAN y MAN las utilizan, e incluso se empieza a utilizar en redes de ámbito WAN.

La utilización de Ethernet impone unos condicionantes a la integración de SARLAB en la red de campus, que se pueden englobar en el control de la acotación de la latencia, que requiere el diseño jerárquico de las redes de campus, y en la capacidad de segmentación virtual (VLAN).

Aunque el diseño exacto de cada red de campus puede presentar sus particularidades, en general todas presentan un diseño jerárquico por capas. En este tipo de diseño, cada capa aporta un grupo de funcionalidades al conjunto y la información se estructura de forma ordenada. Así, es posible acotar los límites de funcionamiento de cualquier punto de acceso a la red, como por ejemplo el ancho de banda esperado, el retardo en el tránsito por la instalación, etc.

La estructuración de la red facilita además la escalabilidad, la redundancia, la seguridad, la administración y el mantenimiento de la misma. El modelo jerárquico más utilizado organiza la red en tres capas, figura 5.6: núcleo, distribución y acceso. En el núcleo se produce la máxima concentración de tráfico, y por tanto es en el que se deben garantizar las tasas de transferencia más altas. El núcleo aporta además la posibilidad de redundancia a la capa de distribución (lo que asegura la tolerancia a fallos) y es responsable de gestionar los enlaces externos con otras redes, incluida la red de tránsito a Internet. Por último, el núcleo es el encargado de ofertar los servicios de la institución y de gestionar su seguridad creando la correspondiente DMZ.

La capa de distribución aporta la redundancia a la capa de acceso, y soporta el nivel de agregación necesario con el núcleo para que la capa de acceso pueda ofertar el ancho de banda del cable en cada punto de acceso. En la capa de distribución se imponen además los niveles de seguridad del tráfico de los distintos segmentos en los que esté organizada la institución.

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

Finalmente, la capa de acceso proporciona puntos de conexión a los elementos finales de la red desde todas las instalaciones y dependencias del campus.

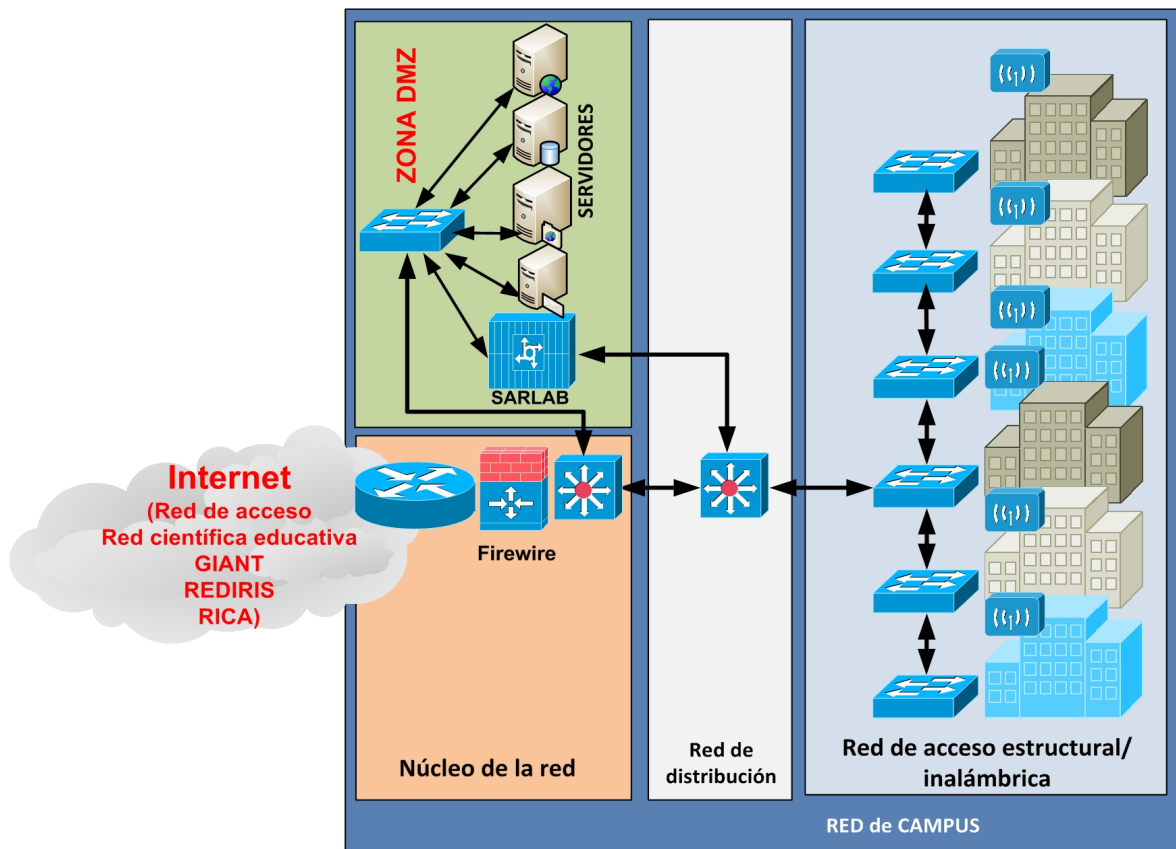


Figura 5.6: Estructura de la red de campus.

El componente clave para el despliegue de una red jerárquica es el conmutador Ethernet, tanto porque permite la comunicación full-dúplex como porque permite la elección de la velocidad de conmutación más apropiada en cada caso. En efecto, la estructura jerárquica, en sí no garantiza la acotación de la latencia ni el ancho de banda mínimo del cable en cada punto de la capa de acceso. Para tener esta garantía, es necesario además dimensionar el nivel de agregación entre los niveles de la estructura jerárquica así como elegir de forma adecuada la velocidad de la matriz de conmutación de los conmutadores Ethernet. Estas características, unidas a la posibilidad de establecer criterios para definir la calidad del servicio al tránsito de información, permiten acotar

la latencia en estas estructuras de redes. Es decir, se puede conocer a priori el retraso máximo que introduce la red aunque su variación no tenga carácter determinista.

Una característica que tienen los conmutadores gestionables utilizados en el despliegue de las redes de campus, es la posibilidad de segmentar el tráfico que entra o sale a través de cualquiera de sus puertos (puntos de acceso). Cada uno de estos segmentos divide la red de campus en un grupo de redes aisladas, a las que se puede asociar por programación cualquier puerto de cualquier conmutador situado en cualquier punto de la red. En consecuencia, se puede crear un segmento de red formado por puertos situados en los laboratorios de toda la institución. Como la definición de los puertos que participan en el tráfico de un segmento se determina por programación, este tipo de red se denomina virtual o VLAN. Aplicado al modelo propuesto, esto permite definir la red virtual VLAN SARLAB a la que se le agregará un puerto en cada laboratorio desde los que se tengan que ofertar experiencias, figura 5.7.

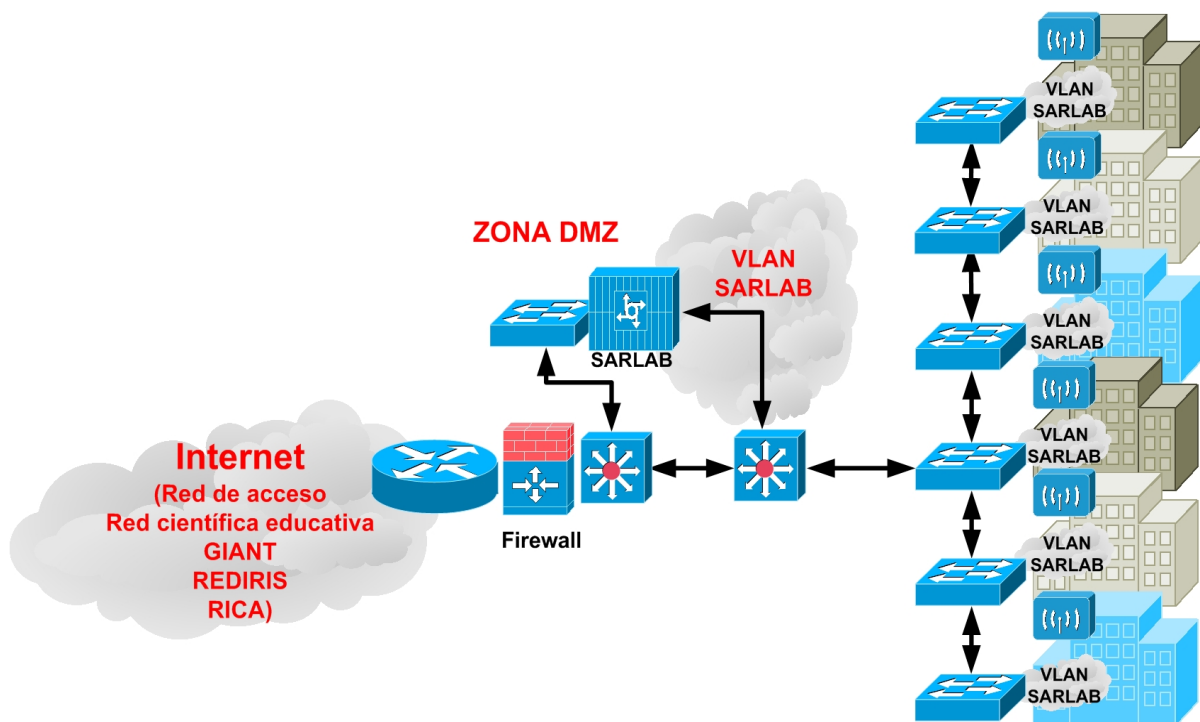


Figura 5.7: Estructura de una VLAN de SARLAB en una institución genérica.

5.2.2. Características a cumplir por los servidores en la red de comunicaciones elegida

Una vez determinadas las redes pública y local del modelo, e integrado SARLAB dentro de la infraestructura de red de la instalación, en este apartado se establecen las recomendaciones necesarias en la configuración física de los servidores SARLAB y LMS. Las soluciones de implementación física son muchas y pueden verse condicionadas por muchos factores, aunque ninguno de ellos está suficientemente avalado. Por tanto, se establecen aquí las recomendaciones mínimas para los dos servidores y se indican las distintas arquitecturas implementadas en las cuatro universidades en las que actualmente se trabaja con SARLAB.

El servidor SARLAB debe disponer de dos interfaces Ethernet, una para conectarlo a Internet (a través del DMZ de la institución) y la otra para conectarlo a la red local de los laboratorios, que será una VLAN desplegada por todo el campus de la entidad y que se ha denominado VLAN SARLAB. El servidor LMS puede estar conectado a cualquier punto de Internet y puede integrarse en el LMS de la entidad o no. Por otro lado, el dimensionado físico de los servidores está sujeto tanto a las necesidades como a los recursos de la institución en la que se va a implantar el sistema. En este sentido, los fabricantes proporcionan tablas de decisión de arquitecturas en función de la utilización prevista y de los recursos económicos. En cualquier caso, los sistemas deben estar monitorizados para advertir sobre saturaciones en el uso de las CPUs, memorias, acceso a disco, etc.

En el caso del modelo propuesto, siguiendo las recomendaciones de diseño que se expusieron en la introducción de este capítulo, estos dos servidores deben utilizar el sistema operativo Linux. Este sistema operativo es paradigma del software libre y tiene acreditado tanto un alto rendimiento en aplicaciones de red, como la posibilidad de ejecutarse en múltiples plataformas de hardware, desde pequeños procesadores a

potentes ordenadores basados en varios procesadores y con varios núcleos. Esto permite adaptar las arquitecturas a las necesidades de demanda. En la tabla 5.1 se pueden ver las arquitecturas de los dos servidores actualmente en explotación en las distintas universidades.

<i>Universidad</i>	<i>Servidor</i>	<i>Procesador</i>	<i>Ethernet</i>
<i>UHU</i>	SARLAB	Intel i5 - 8 GB RAM	1 Gbps
	LMS	Intel Xeon (x2) - 16 GB RAM	1 Gbps
<i>UNED</i>	SARLAB	Intel Xeon (x2) -16 GB RAM	1 Gbps
	LMS	Intel Xeon (x2) -16 GB RAM	1 Gbps
<i>UCM</i>	SARLAB	Intel i5 - 4 GB RAM	1 Gbps
	LMS	-----	----
<i>UCA</i>	SARLAB	Intel i5 – 4 GB RAM	1 Gbps
	LMS	-----	----

Tabla 5.1: Servidores SARLAB y LMS en las universidades en las que está en explotación el modelo.

5.3. Implementación del sistema de convergencia genérico

Conseguir la convergencia de los sistemas experimentales es la condición necesaria y suficiente para que SARLAB pueda realizar el control de las comunicaciones. Hasta ahora, se han considerado los sistemas experimentales convergentes de forma genérica, es decir con una arquitectura de bloques que comprende todas las implementaciones posibles. Una visión constructiva de esta estructura fue introducida en el capítulo 4, donde se estableció que el sistema convergente está constituido por una planta experimental didáctica con sensores y actuadores conectados a varios procesadores/computadores con SADS convergentes, que presentan un conjunto de interfaces de red. Este requisito no supone un condicionamiento de diseño restrictivo, de hecho la mayoría de los sistemas experimentales de las publicaciones analizadas en el capítulo 2 de esta Tesis pueden ser representados con pequeñas matizaciones por este esquema y, en consecuencia, son susceptibles de que SARLAB pueda gestionar sus

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

comunicaciones. Sin embargo, en la mayoría de estos sistemas experimentales la convergencia a red ha sido diseñada para adaptarse a las necesidades particulares de una planta experimental didáctica concreta, en contraposición con la generalidad propuesta en el modelo desarrollado en esta Tesis.

En este apartado se pretende implementar la estructura de bloques del sistema experimental genérico a partir de componentes físicas reales que puedan considerarse genéricas. Es decir, dicho sistema debe conseguir la convergencia de cualquier planta experimental didáctica que cumpla con las condiciones desarrolladas a lo largo de este apartado.

Dado que la naturaleza de la planta experimental didáctica es el elemento diferencial y por tanto no es generalizable, en este apartado se aborda el desarrollo de la arquitectura constructiva de los componentes que permiten aportar la convergencia a cualquiera de ellas. A esta arquitectura se le denominará a partir de ahora *sistema de convergencia genérico*. Por tanto, para obtener el esquema de implementación del sistema de convergencia genérico se eliminarán todas las cuestiones relacionadas con la implementación de la planta experimental didáctica. En este sentido, la planta experimental se considera como una caja negra a la que se le supone un conjunto de n variables de entradas y m variables de salida, figura 5.8. Las salidas serán señales que provienen de los sensores de la planta didáctica y las entradas serán señales que se aplicarán a los actuadores. Se supone por tanto resuelta la sensorización de la planta experimental, que aportará el estado de la evolución del proceso y la actuación en la planta para modificar su evolución. En este sentido, las señales de entrada y salida deben estar normalizadas y su formato debe ajustarse a las posibilidades de los SADS.

El problema de la convergencia queda limitado por tanto a la implementación física de los procesadores/computadores SADS. Para que la solución propuesta sea escalable se presentarán dos arquitecturas de implementación física, una para sistemas sencillos y

otra para sistemas más complejos desde el punto de vista de la concurrencia y procesamiento de la información.

Los SADS utilizados permiten tratar entradas y salidas digitales 0/3.3 V, 0/5 V y entradas y salidas analógicas 0/3.3 V, 0/5 V y salidas PWM 0/3.3 V, 0/5 V y -12/+12 V. Además, como cada vez son más los sensores y actuadores que utilizan señales digitales en formato serie, el sistema DAQ debe permitir entradas y salidas de buses digitales serie RS232, I2C, SPI y DALLAS. Queda implícito por tanto, que las salidas analógicas y digitales tienen cubiertas las necesidades de acondicionamiento como amplificación, filtrado, linealización, optoacoplamiento, etc, en la caja negra que constituye la planta experimental. Se denomina por tanto en este apartado a la planta experimental como *planta experimental sensorizada y acondicionada*. Es importante destacar que las maquetas didácticas que constituyen la base de estas experiencias suelen cumplir todos estos requisitos.

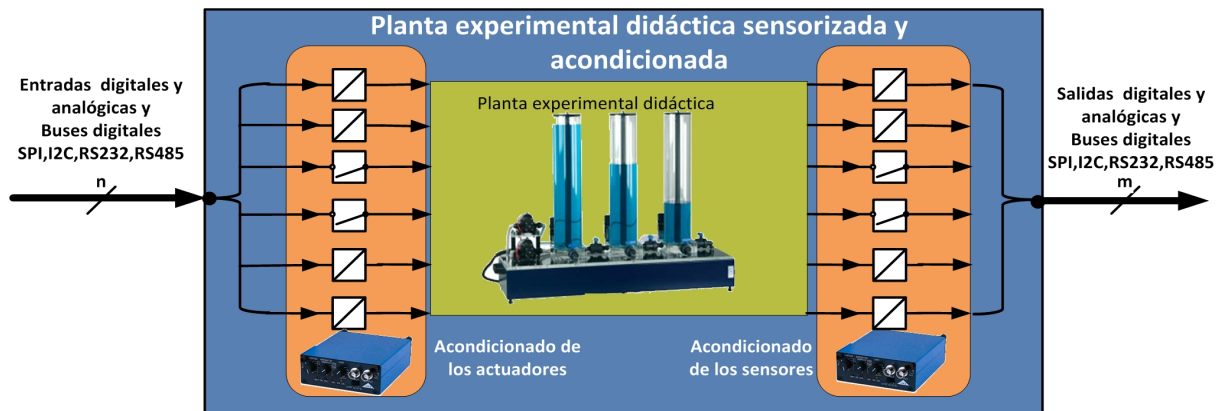


Figura 5.8: Planta experimental didáctica sensorizada y acondicionada.

Para el desarrollo del sistema de convergencia genérico, hay que establecer en primer lugar las distintas capas funcionales de los procesadores/computadores con SADS convergentes, necesarias para conseguir el acceso desde red a la planta experimental didáctica sensorizada y acondicionada.

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

En la figura 5.9 se ha representado el esquema por capas de las distintas funcionalidades que debe aportar el sistema de convergencia genérico. En ella se observan:

- La interfaz de red, que debe ser de tipo Ethernet, por la que el sistema experimental se conectará a la red local SARLAB.
- La capa de procesamiento/computación de datos, que aporta la torre de protocolos TCP/IP, la capacidad de procesamiento necesaria para el mantenimiento de las comunicaciones por la red y las necesidades de procesamiento de los datos de la planta experimental didáctica. También establece las funciones de seguridad necesarias en la planta experimental didáctica, y aquellas otras funciones del bucle de control que deban quedar del lado seguro de la red.
- La capa de adquisición de datos, denominada *convertidores* en la figura 5.9, que proporciona la interfaz necesaria para que la capa de procesamiento y computación reciba y envíe la información necesaria en la evolución de la experiencia.
- Finalmente se ha introducido una capa de adaptación de niveles, denominada acondicionador 0/5V en la figura 5.9, que libera a la planta experimental de ajustar los rangos de tensiones a los óptimos de entrada a la capa de adquisición.

La implementación física de la estructura propuesta no es única. Algunas soluciones utilizan un componente distinto para cada capa, otras utilizan componentes que agrupan varias de ellas, e incluso hay dispositivos que incorporan todas las funcionalidades. SARLAB puede gestionar las comunicaciones de cualquier arquitectura que permita convertir la planta experimental didáctica en un sistema experimental convergente, con independencia de los criterios de diseño utilizados. Sin

embargo, la solución propuesta en este apartado debe estar enmarcada en los principios de desarrollo de esta Tesis, que se resumen en los siguientes puntos:

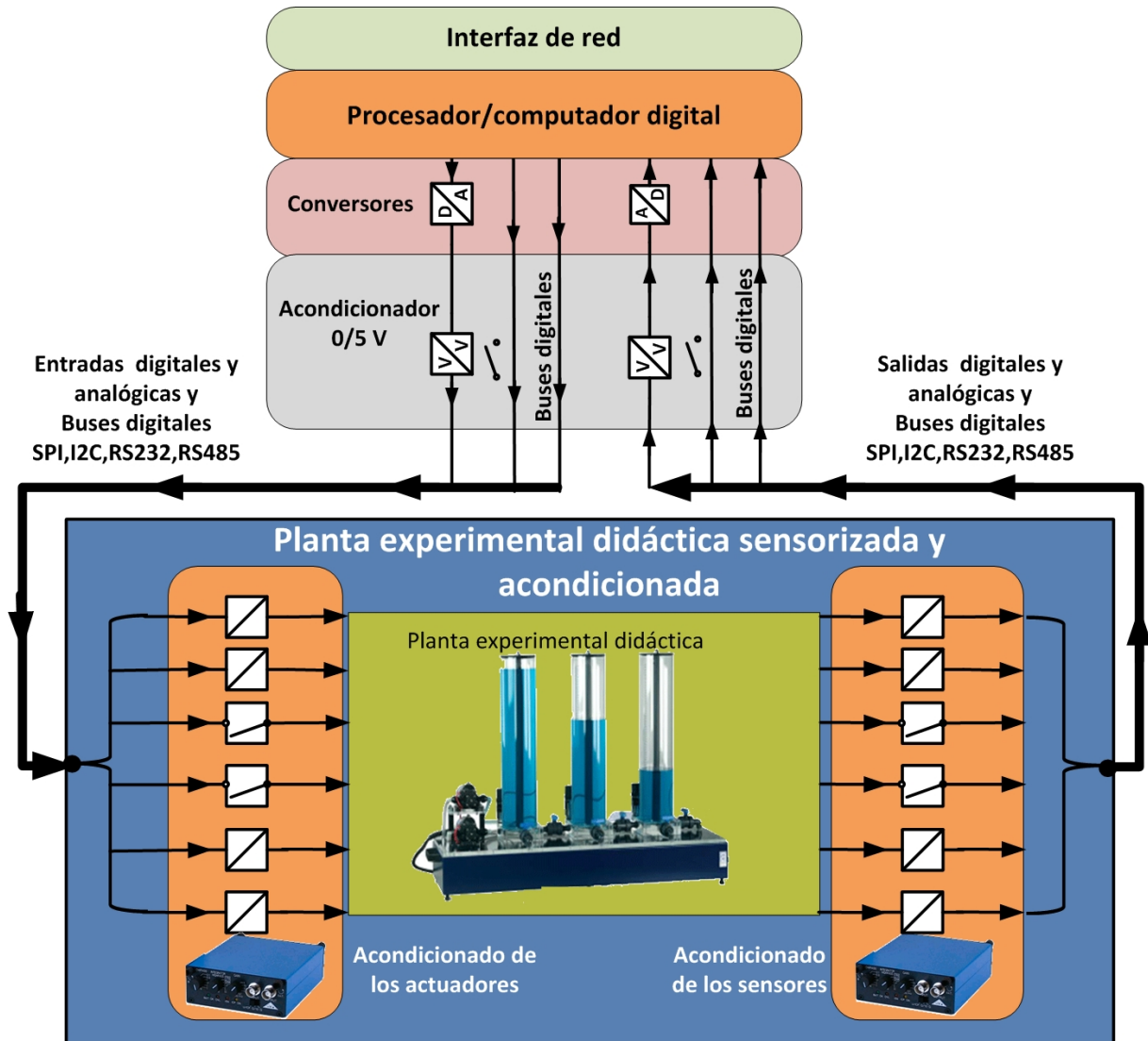


Figura 5.9: Estructura funcional del sistema de convergencia genérico.

- ✓ Generalidad del modelo. A efectos de implementación este principio se concreta en el uso de componentes con capacidad de escalabilidad y modularidad. Así, la solución propuesta se adapta a las necesidades de la planta experimental de cada caso en cada momento.

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

- ✓ Viabilidad económica. A efectos de implementación este principio se concreta en el uso de plataformas abiertas de hardware de bajo coste y software libre. Esto además facilita la generalización del uso de la solución propuesta.
- ✓ Generalidad didáctica. A efectos de implementación este principio se concreta, entre otras alternativas, en la posibilidad de acceso concurrente a las experiencias como una alternativa didáctica.
- ✓ Universalidad del diseño. Para que el diseñador de experiencias no necesite conocimientos avanzados en electrónica, programación y comunicaciones, el diseño debe estar asistido en todas sus etapas.

Se aborda a continuación la implementación de la capa de adaptación de niveles, común a las dos soluciones físicas que se propondrán en este apartado. Esta capa adapta las escalas de tensión de entradas y salidas de la planta experimental didáctica sensorizada y acondicionada a los niveles de la capa de adquisición de datos. Para las señales de tipo digital, el problema es que los niveles lógicos de entrada de los actuadores o de salida de los sensores no se correspondan con los de la capa de conversión, y por tanto sea necesaria la adaptación entre los rangos de tensiones normalizados (0/3.3 V a 0/5 V o viceversa). Para ello, existen dispositivos bidireccionales de bajo coste, como por ejemplo el representado en la figura 5.10.

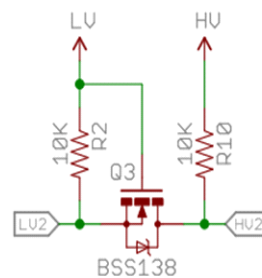
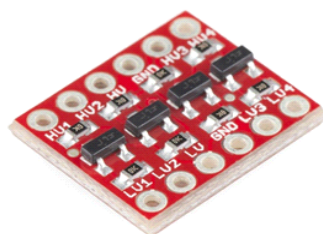


Figura 5.10: Placa de adaptación de niveles digitales bidireccional 3.3 a 5 V.

En el caso de señales analógicas, es necesaria su adaptación a los rangos normalizados de los conversores A/D (0/3.3V o 0/5 V), teniendo en cuenta las necesidades de entrada del convertor A/D utilizado, figura 5.11.

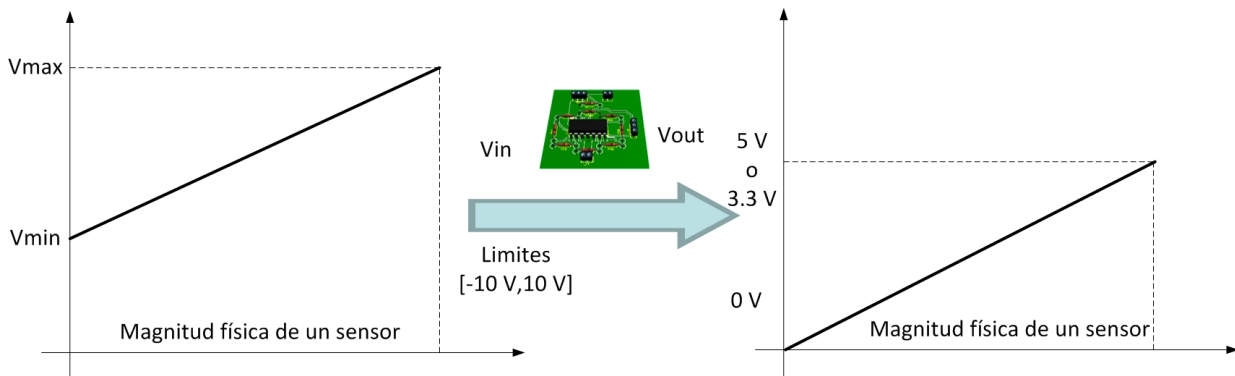


Figura 5.11: Mapeo de rangos de sensores a 0/3.3 0 0/5V.

5.3.1. Implementación práctica de la capa de adaptación de niveles

Para realizar el mapeo se ha diseñado el adaptador de niveles presentado en la figura 5.12 (circuito de mapeo de tensiones). Este circuito, además de mapear tensiones, actúa como convertidor de intensidad a tensión, por ejemplo señales 4/20 mA a 0.4/2 V o 0/20 mA a 0/2 V sin más que cerrar el *jumper* JP3. Por otro lado, el circuito mostrado en la figura 5.12 presenta dos ventajas frente a los comerciales más usados. En primer lugar, está alimentado únicamente con tensión simple de 5 V frente a la alimentación simétrica necesaria por los otros. En segundo lugar, dispone de ajustes de ganancia y desplazamiento independientes.

El procedimiento de ajuste comienza con el conexionado de los *jumpers* JP1 y JP2. En caso de que la tensión máxima de entrada supere el fondo de escala (3.3 o 5 V) se unen los pines 1 y 2 de los *jumpers* JP1 y JP2. En caso contrario, se unen los pines 2 y 3. A continuación se introduce el valor mínimo a mapear en la entrada y se ajusta R6 hasta obtener 0 V en la salida. Posteriormente se introduce el valor máximo de la tensión a

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

mapear en la entrada y se ajusta, hasta obtener el máximo del rango (3.3 V. o 5 V), la resistencia R10 si los *jumpers* están unidos entre 1 y 2 o la R16 si la unión es entre 2 y 3.

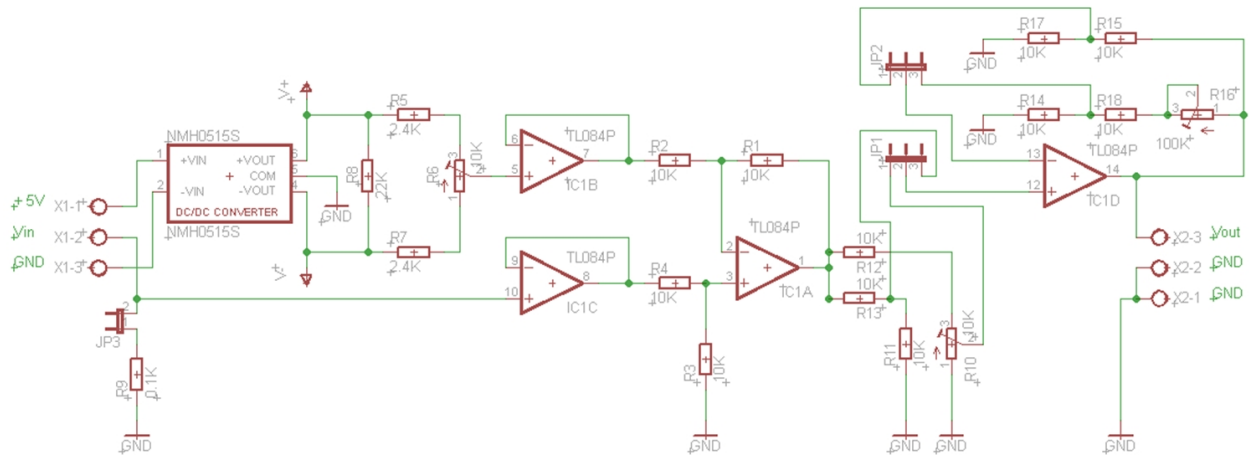


Figura 5.12: Circuito de mapeo a 0/3.3 V o 0/5 V.

La capa de adaptación de niveles debe mapear también las señales de salida de los D/A a las entradas de la planta experimental didáctica sensorizada y acondicionada. Las tensiones normalizadas de entrada de los actuadores suelen ser 0/10 V o -10/10 V, y las salidas de tensión de los D/A están en el rango 0/3.3 V o 0/ 5 V. Para resolver este problema se ha diseñado un circuito que permite realizar los correspondientes mapeos, figura 5.13. Este circuito presenta, como el anterior, tensión de alimentación simple y ajustes independientes (R10 para valores negativos y R11 ó R8 para los valores positivos, según la posición de los *jumpers* JP1 y JP2).

Una vez resuelta la implementación de la capa de adaptación de niveles, la de las tres capas restantes (la interfaz de red, la capa de procesamiento/computación de datos y la capa de adquisición de datos), se puede abordar con dos enfoques que llevan a las dos soluciones mencionadas anteriormente.

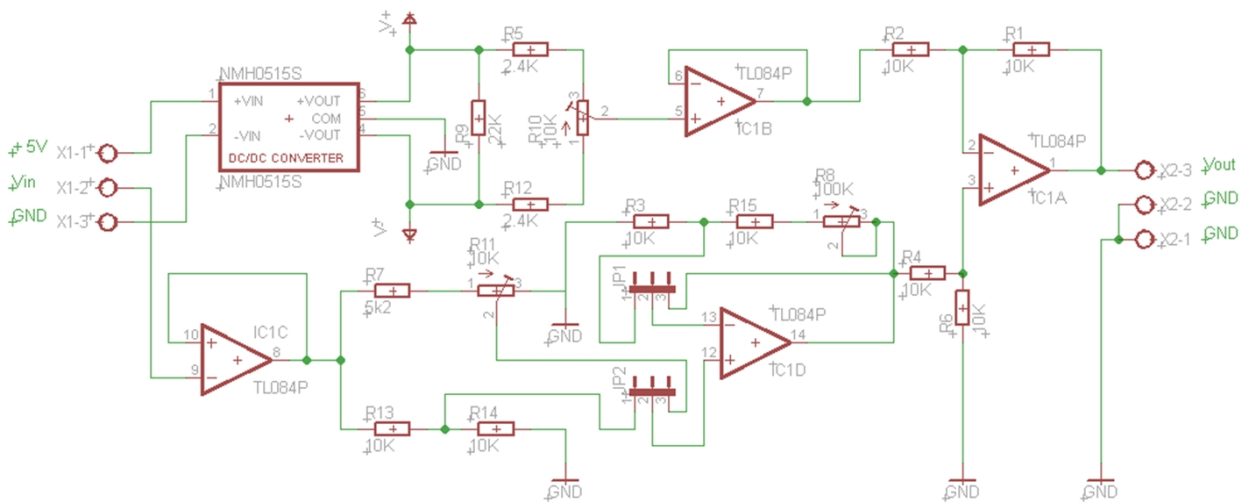


Figura 5.13: Circuito de mapero a rangos 0/10 V -10/10 V o en general $-V/+V$.

5.3.2. Implementación monolítica del resto de capas para sistemas sencillos

En muchas de las aplicaciones, el algoritmo de funcionamiento del sistema de convergencia genérico se limita a poner en marcha los equipos mediante la activación de una señal digital cuando el usuario pulsa un botón en la interfaz de usuario, o a imponer un nivel de tensión cuando el usuario decide el valor de consigna de una variable, o a mostrar a través de la interfaz el valor adquirido por un sensor de la planta experimental didáctica. En estos casos la necesidad de tratamiento de la información asociada al experimento por parte del procesador/computador es baja, y los algoritmos de tratamiento de datos y seguridad de la planta experimental didáctica son muy simples. Por tanto se propone un único componente para completar la arquitectura de las tres capas restantes del sistema de convergencia genérico: una placa de desarrollo de bajo coste basada en microcontroladores. Estos dispositivos disponen de entradas y salidas digitales, entradas y salidas analógicas y buses digitales de comunicación serie, además de capacidad de procesamiento e interfaces de red.

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

Dentro de las posibilidades que ofrece el mercado, la elegida para el desarrollo de la solución propuesta es la plataforma de hardware libre *Arduino*, que en los últimos años se ha convertido en una de las más utilizadas. El sistema de desarrollo de Arduino le oferta al diseñador una familia completa de tarjetas basadas en microcontroladores. Además, existe en Internet un gran número de proyectos de construcción de infinidad de dispositivos totalmente documentados basados en este sistema. Esta documentación aporta información sobre sensores, actuadores y desarrollo constructivo de proyectos de diversa naturaleza, que facilita la creación tanto del sistema de convergencia como de la planta experimental en sí.

A nivel de especificaciones técnicas, las tarjetas de desarrollo de Arduino aportan entradas y salidas tanto analógicas como digitales. Según el modelo, manejan rangos de tensiones 0/5 V o 0/3,3 V y disponen además de buses serie con los protocolos I2C, SPI y RS232. Además existen modelos que disponen de una interfaz Ethernet. Para los que no disponen de ella, existe un módulo de ampliación (shield Ethernet) que proporciona interfaz de red a todas las tarjetas de la familia. Las tarjetas de Arduino disponen además de un gran número de librerías que permiten implementar con facilidad la torre de comunicaciones TCP/IP y las funciones de adquisición y procesamiento. Todo esto la hace adecuada para el desarrollo de la capa de adquisición de datos, de la de procesamiento/computación de datos y de la interfaz de red.

En resumen, esta es la solución propuesta como la arquitectura más adecuada cuando la necesidad de procesamiento es moderada o baja.

5.3.3. Implementación modular del resto de capas para sistemas complejos

Aunque las placas Arduino permiten el desarrollo de programas de control bastante complejos, este hardware no soporta un alto número de accesos concurrentes por la red.

Además, el almacenamiento masivo de información o su procesado se vuelve un proceso excesivamente complejo. Por tanto, en el caso de las aplicaciones con estos requerimientos, se propone una arquitectura alternativa para la implementación de las tres capas restantes del sistema de convergencia genérico.

En esta arquitectura alternativa, se ha incluido un sistema operativo embebido para la gestión del almacenamiento de información y para aumentar la concurrencia de los accesos por red. Asimismo, este sistema dota a la arquitectura de la capacidad de ejecución concurrente, e incluso posibilita el uso de herramientas de desarrollo rápido para diseñar las aplicaciones necesarias en el control la experiencia. Este sistema operativo deberá cumplir con los principios de desarrollo de esta Tesis, enumerados anteriormente. Por tanto, se propone el uso de Linux, que es el sistema operativo de software libre más utilizado. El siguiente paso es elegir un hardware que soporte Linux y aporte las funcionalidades de las capas del sistema de convergencia genérico.

Dentro de la oferta existente se propone el uso de placas microcomputadoras basadas en procesadores de la familia ix86 y ARM. La ventaja principal de esta arquitectura frente a otras es su adaptabilidad y escalabilidad. Además son de las más utilizadas en aplicaciones industriales.

La familia ix86 (32 y 64 bits) dispone de un gran número de placas base con factores de forma Mini-ITX (17x17 cm), Nano-ITX (12x12 cm), Pico-ITX (10x7,2 cm), NUC (10x10 cm), etc., figura 5.14. Estas placas base permiten el escalado de la arquitectura física según las necesidades de procesamiento, mediante el uso de procesadores de 32 bits hasta procesadores de varios núcleos de 64 bits, como los Intel Atom, i3 e i5. También soportan procesadores de Intel, AMD, VIA, etc.

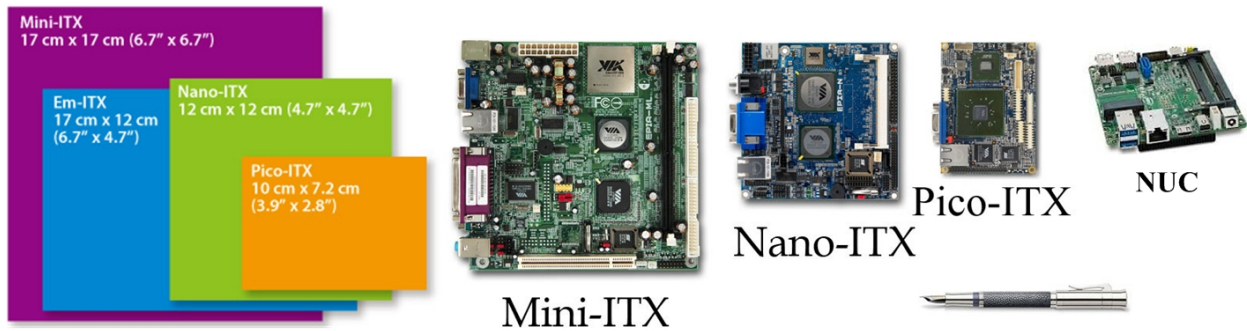


Figura 5.14: Placas microprocesadoras ix86 (32 y 64 bits).

De idéntica forma, para el procesador ARM existe un gran número de placas base para usos industriales fabricadas con los mismos factores de forma que las placas ix86, que igualmente pueden ser utilizadas para implementar esta segunda solución. Pero además, existen placas basadas en este procesador con tamaños y precios muy reducidos, como por ejemplo el modelo Phidget SBC3 y Raspberry Pi, figura 5.15. Estas placas permiten embeber el sistema operativo Linux y constituyen una solución intermedia entre la solución basada en Arduino y el uso de placas ix86.

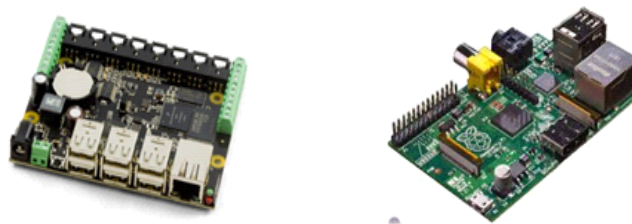


Figura 5.15: Placas basadas en procesadores ARM (PhidgetSBC3 y Raspberry Pi).

Todas las arquitecturas de esta segunda solución incorporan un sistema de almacenamiento para la instalación del sistema operativo Linux y una interfaz de red, además de al menos un puerto USB. Con eso, cualquiera de ellas aporta las funcionalidades de la capa de interfaz de red y de la de procesamiento/computación de

datos. Falta por tanto la implementación de la capa de adquisición de datos para completar la estructura del sistema de convergencia genérico.

Dado que todas las arquitecturas disponen de un bus USB, para el que existen tarjetas de adquisición de datos de bajo coste, serán éstas las utilizadas para implementar la última capa. De entre todas las posibilidades, se ha optado por los dispositivos de adquisición y control de la firma Phidgets. Esta firma dispone de una gama de tarjetas USB con drivers para los sistemas operativos Windows, Linux y OS X, además de librerías para el desarrollo de aplicaciones en cualquier lenguaje de programación (incluyendo Java), característica que supone una ventaja frente a las otras alternativas. Esta firma aporta placas de adquisición con entradas y salidas digitales y entradas analógicas (1018_2 - PhidgetInterfaceKit 8/8/8), placas con salidas digitales -10/10 V (1002_0 - PhidgetAnalog 4-Output), placas con salidas PWM (1064_1 - PhidgetMotorControl HC), etc. Estas placas se pueden conectar a cualquier arquitectura ix86 o ARM con el sistema operativo Linux, lo que permite adecuar la capacidad de procesamiento a las necesidades del problema, utilizando la placa que mejor se adapte a las dificultades del algoritmo implementado. Estas placas de la firma Phidget permiten implementar la capa de adquisición de datos del sistema de convergencia genérico. Pero además esta firma ofrece una placa (1073_0 - PhidgetSBC3) basada en un procesador ARM con Linux embebido, que incorpora una interfaz 8/8/8 (8 entradas digitales, 8 salidas digitales, 8 entradas analógicas) y 6 puertos USB, donde puede instalarse cualquiera de las demás tarjetas para desarrollar la estructura completa del sistema de convergencia genérico.

Dentro de la solución modular propuesta, existe la alternativa de utilizar las placas de desarrollo de Arduino como esclavos DAQ conectados a un puerto USB de cualquiera de las placas de procesamiento/computación propuestas. Para ello, es necesario cambiar el firmware original de la placa Arduino por el denominado Firmata, que permite trasladar el procesamiento de las entradas y salida de Arduino al

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

computador al que esté conectado mediante un puerto USB. La figura 5.16 recoge la figura 5.9, en la que se han incluido las dos soluciones propuestas de implementación de las tres capas superiores del sistema de convergencia genérico. La solución monolítica para sistemas sencillos se representa en el centro de la figura 5.16, abarcando las tres capas cuyas funcionalidades implementa. La solución modular se presenta a la derecha de la figura abarcando igualmente las mismas capas. En ella se observa la placa base procesadora y las distintas alternativas indicadas para la adquisición de datos. Se observa además en la figura la placa que representa la adaptación de niveles.

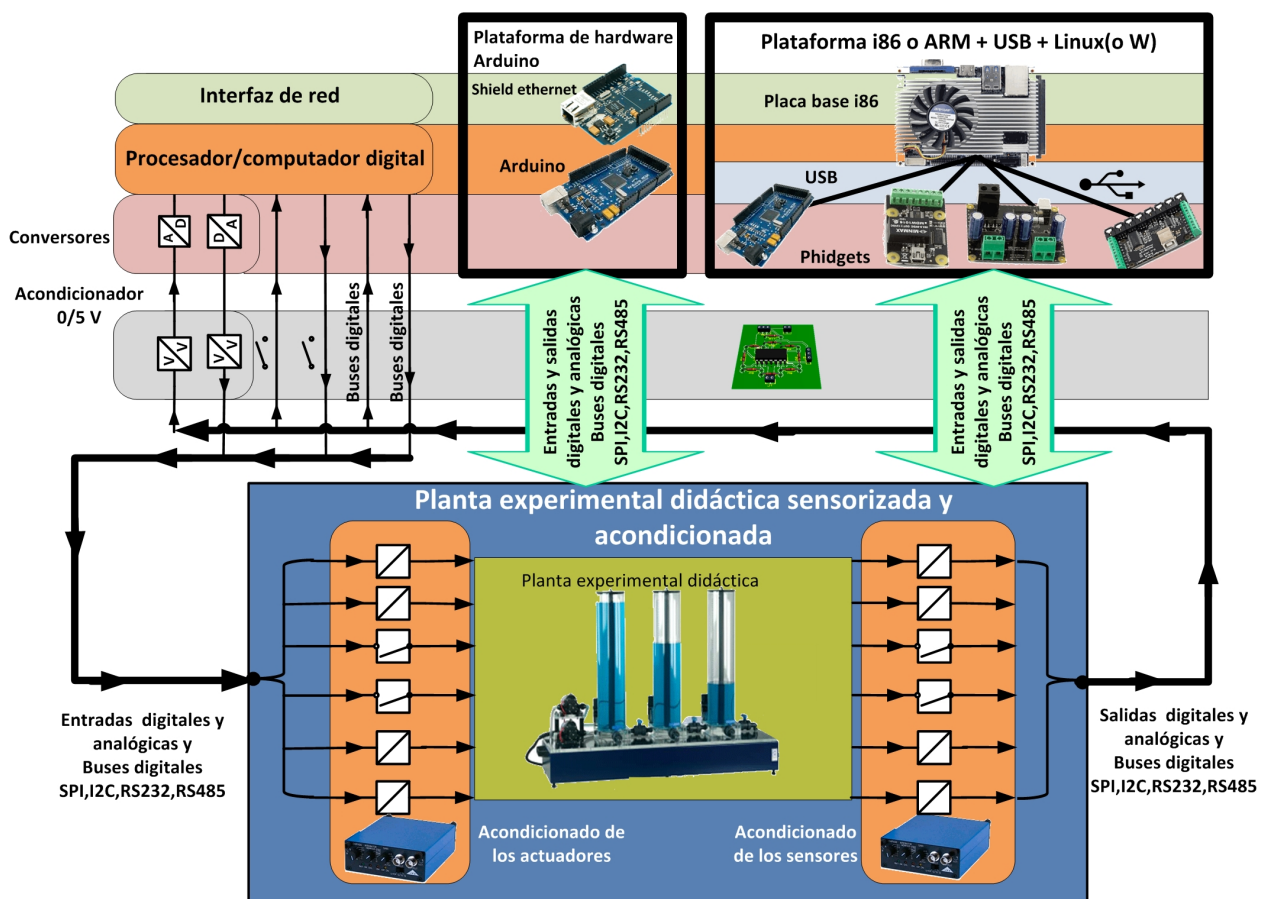


Figura 5.16: Esquema de implementación de un sistema experimental convergente SARLAB.

5.4. Implementación del subsistema de alimentación

El subsistema de alimentación es el dispositivo físico encargado de proporcionar alimentación eléctrica a los componentes del sistema experimental, cuyo encendido y apagado debe ser controlado por el servidor SARLAB por razones de seguridad y eficiencia energética. La estructura de este subsistema se corresponde con un conjunto de relés controlados desde red por el servidor SARLAB, figura 5.17.

En la figura se observan las distintas bases de alimentación en las que se conectarán los componentes del/los sistema/s experimental/es. Cada una está controlada por un relé cuya señal de control, convenientemente amplificada, proviene de un puerto de Entrada/Salida de Propósito General (General Purpose Input/Output, GPIO). El puerto GPIO está conectado a su vez a un procesador con interfaz de red con el que se comunica SARLAB.

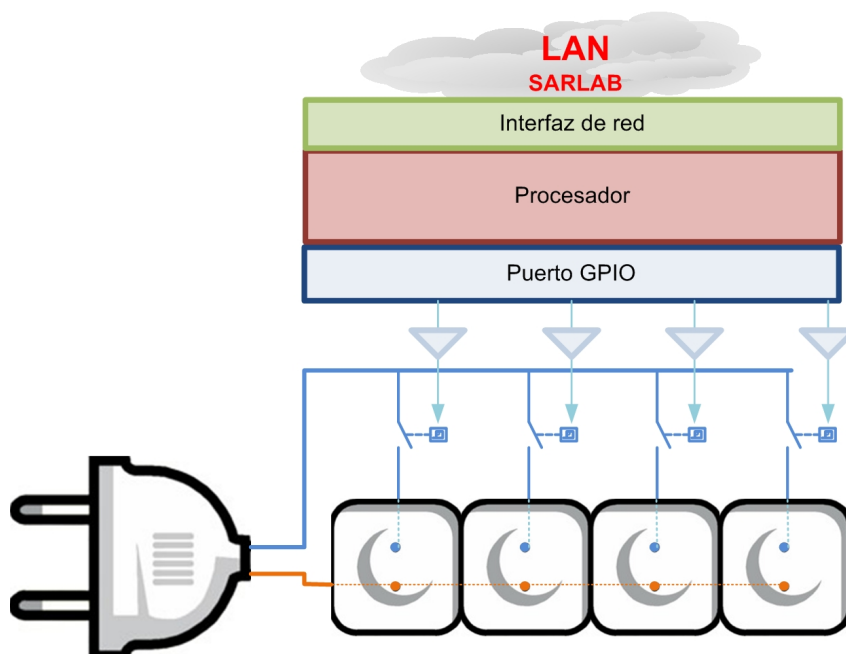


Figura 5.17: Esquema de bloques del subsistema de alimentación.

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

Este esquema es el que presentan algunos dispositivos comerciales que se denominan regletas de alimentación, y que han sido diseñados para controlar la alimentación de los servidores informáticos de forma remota. Pueden ser por tanto usados como subsistema de alimentación del modelo propuesto en esta Tesis. Efectivamente, SARLAB permite el uso de cualquier regleta de alimentación de la firma APC. La figura 5.18 muestra un modelo de 8 salidas controladas de alimentación de esta firma.



Figura 5.18: Regleta de alimentación de la firma APC.

Sin embargo, este tipo de dispositivos tiene algunas limitaciones para el control de la alimentación de los sistemas experimentales. Por ejemplo, no soporta el acceso concurrente. Para solventar este problema, se ha desarrollado en este trabajo un conjunto de soluciones que sí lo permiten, y que además mejoran los protocolos de acceso desde SARLAB e incorporan entradas y salidas de nivel TTL, para realizar el apagado coordinado de los componentes del sistema experimental, figura 5.19.

En algunas aplicaciones, la alimentación de algunos componentes del sistema experimental no debe ser cortada de forma abrupta, para evitar daños en algún dispositivo o el bloqueo del sistema para nuevas ejecuciones. La secuencia de apagado de estos componentes requiere en muchos caso que el sistema de control inicie una determinada secuencia de parada segura, activada por SARLAB mediante la señal TTL. El final de la secuencia se le comunica a SARLAB por las entradas TTL. Para casos menos exigentes, tanto las regletas como los dispositivos desarrollados tienen la opción de introducir retrasos en el encendido/apagado de los sistemas.

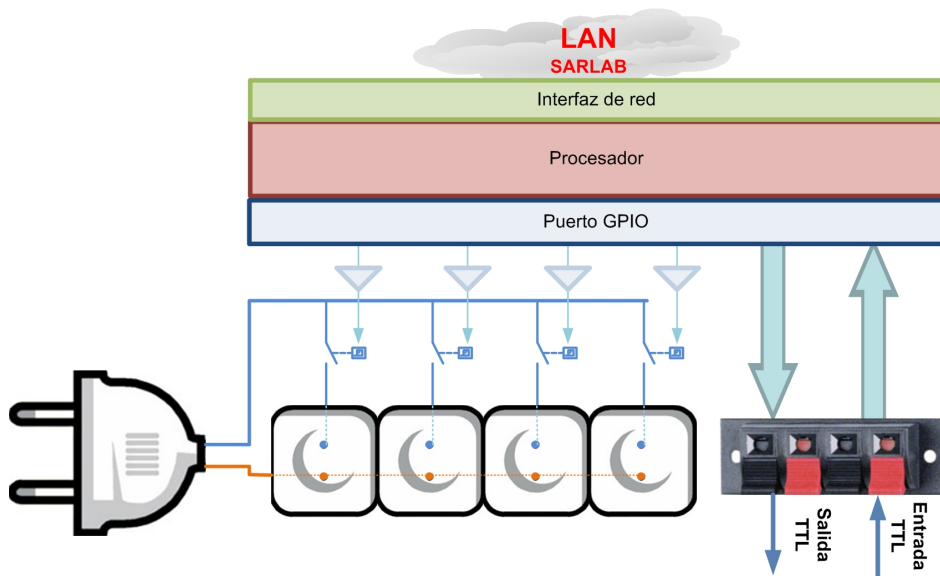


Figura 5.19: Subsistema de alimentación con entradas/salida TTL.

Los dispositivos desarrollados se han implementado con tres arquitecturas distintas, usando el mismo protocolo. Dos de las arquitecturas son físicas y la tercera se denomina virtual porque no utiliza recursos propios, sino parte de los del sistema de convergencia.

La primera implementación física utiliza una tarjeta Arduino Ethernet con un módulo de expansión de 4 relés, figura 5.20. Este subsistema controla hasta cuatro salidas con un máximo de 2 A por salida y admite hasta 5 conexiones concurrentes.

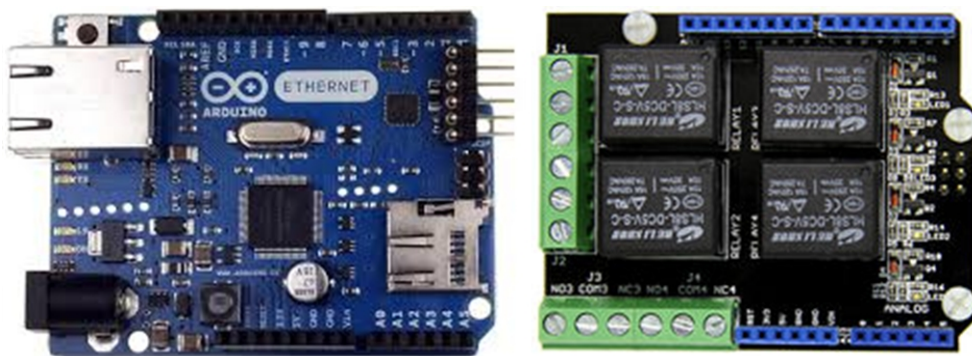


Figura 5.20: Módulos de implementación del subsistema de alimentación con Arduino.

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

La segunda implementación física puede utilizar cualquier tarjeta microprocesadora con sistema operativo Linux, que disponga de un puerto GPIO y que disponga de librerías de acceso al puerto GPIO en Java. Para esta arquitectura se ha llevado a cabo una aplicación en Java que implementa el protocolo de comunicaciones de SARLAB para los subsistemas de alimentación. Además, se incluye un conjunto de relés con entrada de control amplificada y optoaislada de las que se pueden encontrar asociaciones de 1, 2, 4 y 8 relés, figura 5.21.

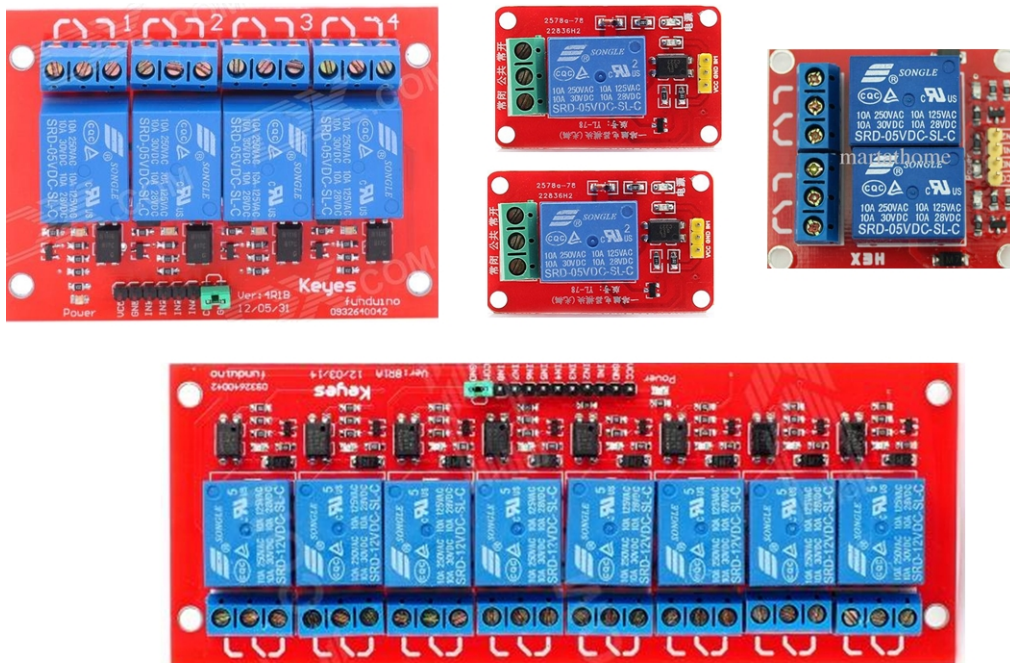


Figura 5.21: Módulos de relés amplificados y optoaislados.

Finalmente, teniendo en cuenta que los sistemas de convergencia se han desarrollado en Java y, en muchos casos, quedan libres entradas y salidas digitales, el usuario puede ejecutar la aplicación que implementa el protocolo de control del subsistema de alimentación mediante esas entradas y salidas libres, eligiendo el módulo de relé que más se ajuste a las mismas. Para intensidades superiores a 10 A se ha diseñado una placa de relé de 16 A optoaislada.

5.5. Implementación del subsistema de transmisión multimedia

El subsistema de transmisión multimedia es el responsable de trasladar a la interfaz de usuario la imagen y el sonido del sistema experimental durante la realización de una experiencia remota, y está constituido por la cámara de vídeo y el servidor de vídeo y audio (en caso de que la cámara no lo incluya). La función de este componente es fundamental para la percepción experimental del usuario (que perciba la sensación de interactuar con equipos reales). En este apartado se estudia la manera de integrar el vídeo en la interfaz de usuario, así como las distintas alternativas de implementación del servidor multimedia.

La manera más fácil de integrar el vídeo en la interfaz de usuario es utilizar las denominadas cámaras IP, ya que estos dispositivos integran todos los componentes necesarios para ello. Una cámara IP tiene un diagrama de bloques genérico como el indicado en la figura 5.22.

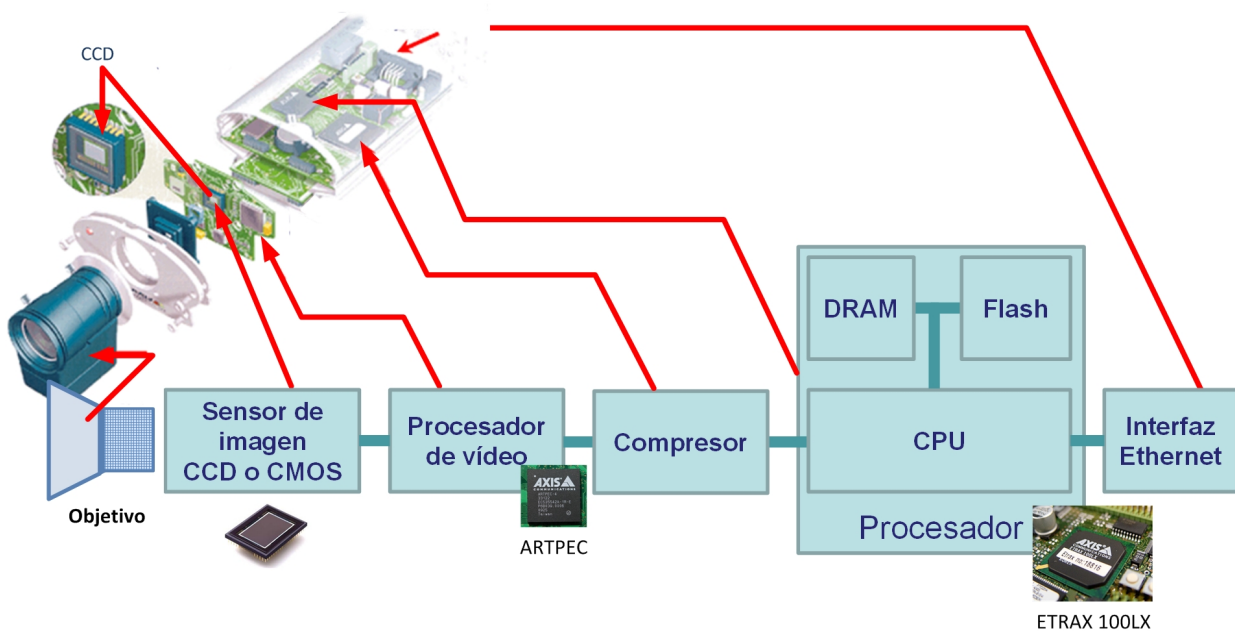


Figura 5.22: Diagrama de bloques de una cámara IP.

5. Implementación Física del Modelo Propuesto basado en la Tecnología Disponible

En la figura 5.22 se observa en la parte inferior izquierda el objetivo, que es el componente encargado de controlar la entrada de luz a la cámara según sean las condiciones de iluminación. Suele disponer de tres ajustes: el zoom, que cambia el tamaño de la imagen captada por la cámara para centrar los detalles; el enfoque, que consigue una imagen nítida y la abertura de diafragma, que regula la luminosidad de la imagen. Estos ajustes se regulan actuando directamente en el objetivo de la cámara, aunque en algunos casos se pueden regular a través de la red. Muchas de estas cámaras disponen del modo autoajuste para el enfoque y la abertura del diafragma. La calidad del objetivo condiciona fuertemente la calidad y el precio de la cámara.

El segundo componente que se observa en el diagrama de bloques es el sensor de imagen, que está constituido por un circuito integrado de tecnología CCD (*charge-coupled device*) o CMOS (*complementary metal oxide semiconductor*), y que se encarga de registrar en forma de tensión la iluminación y el color de la imagen. Este componente condiciona la resolución de la imagen.

El procesador de vídeo es el encargado de recibir la información del sensor de imagen y procesarla para introducirla en el compresor de vídeo, el cual recibe la imagen y la comprime a un formato de transmisión por red (JPEG, MJPEG, H.261, MPEG, H.263, MPEG-4, H.264). El procesador, que incorpora un sistema operativo y dispone de interfaz ethernet, transmite a la red el flujo de vídeo.

Una forma alternativa de integrar el vídeo en la interfaz de usuario consiste en utilizar servidores de vídeo con cámaras externas. Estos servidores tienen una estructura muy parecida a la de las cámaras IP. La única diferencia es que no contienen cámara y por tanto hay que aportar la señal de vídeo desde una cámara externa. De esta forma, el procesador de vídeo queda conectado a la salida de vídeo de la cámara a través de un conector y no de forma interna como en el caso de la cámara IP. La señal de vídeo transmitida por la cámara debe estar en un formato entendible por el

procesador como por ejemplo vídeo compuesto, vídeo RGB, S-Vídeo, USB, IEEE 1394, HDMI, VGA, etc. Existen servidores que disponen de múltiples entradas y admiten múltiples formatos, figura 5.23.



Figura 5.23: Algunos servidores de vídeo comerciales.

Otra alternativa para integrar el vídeo en la interfaz de usuario es implementar la arquitectura de la alternativa anterior sobre una computadora de propósito general, y ejecutar un programa servidor de vídeo. Algunos ejemplos de este software para el sistema operativo Windows son webcamXP, AbelCam, webcam-monitor, Yawcam o tincam. Para Linux se pueden utilizar Motion, MJPG-streamer o ZoneMinder. Finalmente, algunos ejemplos para sistemas Apple son AirBeam, EvoCam y RemoteSight. En esta alternativa se puede incluir una tarjeta capturadora para adquirir entradas analógicas de vídeo, o se pueden utilizar los puertos digitales (USB o IEEE 1394) de la computadora.

En cualquiera de las soluciones mencionadas, el flujo de vídeo será accesible desde la dirección IP y el puerto del equipo que actúe como servidor de vídeo.

Con respecto al servidor de audio, algunas cámaras IP lo llevan incorporado así, como el micrófono.

5.6. Conclusiones

Se ha desarrollado en este capítulo el esquema de implementación física del modelo tecnológico general, entendido como los elementos hardware del sistema y los aspectos de integración en la infraestructura de red, así como la configuración de las redes locales, y los condicionantes que los distintos sistemas de Internet imponen al tráfico de información. Las características que deben tener los servidores LMS y SARLAB, a tenor de estos condicionantes, se han presentado en este capítulo, para garantizar el funcionamiento del modelo propuesto en escenarios reales, tales como universidades y centros de investigación.

En este escenario de infraestructura de red, se han propuesto distintas soluciones para el diseño y la construcción del sistema experimental convergente, incluyendo circuitos electrónicos de acondicionamiento y conversión de niveles, algo completamente necesario debido a la heterogeneidad de sensores y actuadores que pueden coexistir en un sistema experimental remoto.

En un laboratorio remoto, es de vital importancia el control de la alimentación de los distintos elementos que forman el sistema experimental, para asegurar su correcto funcionamiento y la gestión energética. En este capítulo se han presentado tanto soluciones comerciales, gestionables directamente por SARLAB, como soluciones realizadas a medida que permiten cubrir las necesidades tanto de sistemas sencillos como de aquellos de mayor complejidad.

El servidor multimedia es también parte fundamental de un laboratorio remoto, en cuanto a que es el encargado de transmitir en tiempo real la evolución del sistema experimental que está siendo manejado de forma remota por el alumno. También se han presentado distintas alternativas reales de implementación para este sistema que cubren las necesidades del modelo general.

En el capítulo siguiente se aborda el desarrollo de las funcionalidades del modelo tecnológico general, y sus correspondientes aplicaciones informáticas (software).

6. Implementación Funcional del Modelo basado en la Tecnología Disponible

La evolución del modelo tecnológico general hacia una implementación concreta conlleva el desarrollo de la estructura de componentes físicos, necesarios para la integración de SARLAB en la infraestructura de red de las instituciones, utilizando la tecnología actual. Esta tarea se ha realizado en el capítulo anterior. A pesar de ser una implementación concreta, el esquema propuesto tiene carácter general y abierto por dos razones. Por un lado, permite la implementación de cualquier experiencia con independencia de su naturaleza experimental, y por otro, en aquellos componentes del sistema en los que es posible la implementación con múltiples estrategias de diseño, como es el caso de los sistemas experimentales convergentes, se propone un conjunto de soluciones concretas y escalables, pero también se deja abierta la posibilidad al uso de cualquier otra solución que cumpla con las condiciones mínimas del modelo. Para alcanzar el objetivo de obtener un sistema explotable, se deben desarrollar las aplicaciones necesarias para que los distintos componentes del esquema del capítulo anterior aporten las funcionalidades que les asigna el modelo tecnológico general. Esto es lo que se aborda en el presente capítulo.

Por tanto, en este capítulo se establecen en primer lugar las estrategias generales de desarrollo de las aplicaciones necesarias para cumplir con las funcionalidades

impuestas por el modelo, para cada uno de los componentes del esquema de implementación física desarrollados en el capítulo anterior. A continuación se abordará la elección del LMS y la integración de SARLAB en el mismo. Esto garantiza, según el modelo conceptual general, el establecimiento del lazo de gestión experimental. Como siguiente etapa se abordará el desarrollo de las aplicaciones encargadas de cerrar el lazo de desarrollo experimental. Para eso, se analizarán previamente las ventajas que se derivan del uso de protocolos de comunicaciones unificados, y se plantearán las cuestiones relacionadas con la creación de la interfaz de usuario y la elección de la herramienta de desarrollo rápido. Llegados a este punto, se estudiará el desarrollo de las funcionalidades del servidor SARLAB, que es el componente fundamental en la gestión y el control de las comunicaciones, como se estableció en el modelo conceptual general, en el que ya se proponía SARLAB para la gestión del lazo de desarrollo experimental. En este sentido, hay que indicar que las funcionalidades de SARLAB están divididas, según el modelo tecnológico general en dos partes: las relacionadas con el establecimiento de las conexiones (CONEXLAB) y las relacionadas con la gestión de las configuraciones y el control de acceso (DIGEXLAB). De la misma forma, para el desarrollo funcional de CONEXLAB se tendrán en cuenta las consideraciones propuestas en el modelo tecnológico general, esto es, utilizar túneles para el establecimiento de conexiones entre el PC del alumno y el sistema experimental convergente.

6.1. Estrategia de desarrollo de las funcionalidades del modelo tecnológico general

En la introducción del capítulo 5 se concretaron los condicionantes fundamentales para realizar la implementación del modelo tecnológico general utilizando los recursos disponibles y las infraestructuras actuales de las instituciones. Además de tenerlos en cuenta en el desarrollo de los componentes físicos de la implementación concreta,

abordada en el capítulo anterior, también deben considerarse en el desarrollo de las aplicaciones que permitan la concreción de todas las funcionalidades necesarias, abordado en el presente capítulo. Para esto último, además de esos condicionantes, se tendrán presentes recomendaciones propias del desarrollo de aplicaciones, como por ejemplo la reutilización de código, el diseño orientado a objeto, etc.

Por tanto, la construcción de las aplicaciones se aborda en este capítulo desde dos alternativas de desarrollo distintas. Por un lado, el uso de aplicaciones genéricas existentes que hay que adaptar a las necesidades del modelo, bien a través de la definición de nuevos módulos o bien redefiniendo algunas de sus funcionalidades (como es el caso del LMS o de la herramienta de desarrollo rápido de la interfaz de usuario). Por otro lado, y solo cuando sea necesario, el desarrollo completo de la aplicación (como el diseño de DIGEXLAB y CONEXLAB). En este último caso, se crearán tanto las aplicaciones cliente que se ejecutarán en el PC del alumno como las aplicaciones servidoras que se ejecutarán en el servidor que corresponda. Para los desarrollos en los que se parte de una aplicación genérica existente, en su elección se tendrá en cuenta la existencia de mecanismos que faciliten la creación y ampliación de las funcionalidades originales. Para las aplicaciones íntegramente desarrolladas se utilizará un lenguaje de desarrollo de aplicaciones de alto nivel, que cumpla con los condicionantes relacionados en la introducción del capítulo 5, y que se concretará en el siguiente apartado.

6.1.1. Elección del lenguaje de programación

Para el desarrollo íntegro de las aplicaciones se ha optado por el lenguaje de alto nivel de software libre Java. En esta elección se ha tenido en cuenta que Java está entre los lenguajes más utilizados para la implementación del acceso remoto a experiencias de laboratorio, como ha quedado patente en el capítulo 2. Además, se ha tenido en cuenta la fácil integración de las aplicaciones desarrolladas en Java en la tecnología web, que es

la más utilizada como marco de ejecución de la interfaz de usuario. Otra ventaja de Java es la existencia de una gran cantidad de librerías para implementar la estructura cliente-servidor en muchos de los protocolos de aplicación de comunicaciones más usados (http, ftp, smtp, imap, snmp, etc.). Esto facilita el desarrollo de los clientes y los servidores necesarios en el modelo. Asimismo, existen muchas librerías científicas, multimedia, etc., e incluso IDEs (*Integrated Development Environment*) de libre distribución para el desarrollo en este lenguaje. Otra característica destacable es su capacidad de ejecución multiplataforma, lo que permite que cualquier aplicación desarrollada en este lenguaje pueda ejecutarse en los tres sistemas operativos más utilizados: Linux, Windows, y MAC OS X. Por otro lado, este lenguaje está orientado a objetos, e incluye facilidades para el intercambio de estructuras de datos y objetos y para la ejecución remota por red. Finalmente, existen herramientas de desarrollo rápido de aplicaciones como EJS, Processing, MyOpenLab, etc., escritas en Java que pueden ser utilizadas para crear la interfaz de usuario del modelo.

El uso de Java, unido al sistema operativo Linux, es por tanto un recurso importante en el desarrollo de esta tesis, en la que se crea software para dispositivos de una gran variedad de arquitecturas, desde pequeños microprocesadores utilizados para proporcionar la convergencia de los sistemas experimentales, hasta ordenadores basados en procesadores con múltiples núcleos para prestar los servicios de SARLAB.

6.2. Elección del LMS

En este apartado se plantean los condicionantes a tener en cuenta a la hora de elegir el LMS que se encargará de la gestión del aprendizaje en el modelo. Se determinarán además las funcionalidades que debe tener el LMS para conseguir la integración completa de SARLAB, que se aborda a dos niveles. Por un lado la inclusión de SARLAB en la tecnología web como una solución intermedia al problema global de integración

6. Implementación Funcional del Modelo basado en la Tecnología Disponible

de SARLAB en cualquier LMS, y por otro lado la integración total de SARLAB en una plataforma de amplio nivel de utilización, como por ejemplo Moodle. En este segundo nivel, se utilizará como punto de partida la solución del nivel anterior.

Los LMS constituyen la evolución natural de los gestores de contenidos Web en el campo de la educación. Los Sistemas de Administración del Aprendizaje se han convertido en la herramienta fundamental para que el alumno acceda al conocimiento dentro de un sistema de aprendizaje electrónico a través de Internet. Estos sistemas han evolucionado de simples repositorios de información, a los que el alumno accede con su navegador web para bajar o subir información, a ricas aplicaciones web en las que se permiten múltiples roles (profesor, autor, alumnos, tutores, etc.) y múltiples estrategias pedagógicas. Estos sistemas no se limitan a proporcionar los medios para el intercambio de información, sino que permiten gestionar con éxito la colaboración en grupo, la monitorización de la evolución del aprendizaje y la puesta en práctica del aprendizaje adaptativo y generativo, permitiendo usar pautas mucho menos fijas y que pueden evolucionar de manera dinámica según el discurrir del aprendizaje del alumno. Estos sistemas consiguen que el desafío ya no sea la utilización de las nuevas tecnologías para recrear los modelos educativos tradicionales, sino el aprovechamiento máximo de las nuevas posibilidades que ofrecen las Tecnologías de la Información y las Comunicaciones para aplicar las nuevas metodologías didácticas.

Durante la primera década de este siglo se produce un enorme crecimiento en el uso de los LMS, lo que propicia la aparición de un gran número de desarrollos con distintas características. En este escenario las instituciones empleaban herramientas de selección para determinar el LMS que mejor se ajustaba a sus necesidades. Un ejemplo de estas herramientas, quizás la más conocida, es EduTools. En su sitio web se facilitaban matrices de características y criterios de elección de los LMS. Durante el periodo de funcionamiento del sitio (2001 a 2013) fue usada por un gran número de entidades. Sin embargo, los LMS han evolucionado en los últimos años hasta convertirse en

aplicaciones modulares que incorporan mecanismos de ampliación de sus capacidades funcionales en forma de nuevos módulos. La competencia entre los distintos equipos de desarrollo de los LMS los ha llevado a la continua evolución y uniformidad de sus módulos, creados tanto por los propios desarrolladores como por la comunidad de usuarios. Esto configura un escenario en el que existen muy pocas diferencias a nivel de características y los criterios de selección van más a cuestiones de apariencia, costes de instalación, compatibilidad entre instituciones que colaboran, etc.

Este escenario en el que las diferencias entre todas las plataformas a nivel de características son pequeñas, y en el que cada institución dispone de sus propio LMS para la gestión del aprendizaje a través de Internet, hace pensar que lo más razonable sería realizar la integración de SARLAB en el LMS de cada institución. En este caso, las experiencias remotas se realizarían como una actividad más. De esta forma, se facilitaría la labor del profesor, que podría intercalar las actividades prácticas en la misma plataforma donde realiza el resto de las funciones de gestión del aprendizaje a través de la red. Esta estrategia no impone el uso de un LMS para la integración de SARLAB, sino que propone la integración de éste en cualquiera de los LMS que tenga en explotación la entidad.

Si bien esta elección puede parecer la más razonable desde el punto de vista de homogeneización de los servicios ofertados desde una institución, esto puede suponer un freno para aquellas en las que su política de gestión de la plataforma educativa esté restringida. Por tanto, se debe prever la posibilidad de instalar un LMS dedicado solo a la realización de prácticas de forma remota controladas por SARLAB. En este caso, y para que los profesores y alumnos tengan una misma interfaz de relación en la realización de actividades, lo recomendable sería el uso del mismo LMS de la institución.

6.2.1. Integración de SARLAB en el LMS

El primer paso para conseguir la integración completa de SARLAB como parte de un LMS implica que todas las aplicaciones cliente necesarias para la realización remota de una experiencia deben poderse incluir como componentes de una página web. Esto es, tanto la interfaz de usuario de las experiencias remotas, como los clientes de las funcionalidades de CONEXLAB y DIGEXLAB, deben incorporar mecanismos para que se pueda acceder a ellas desde un navegador web ejecutado en el PC del alumno. Esto no conlleva ninguna dificultad de desarrollo de los clientes, ya que su desarrollo en el lenguaje Java permite su inclusión como componente en una página web, compilando las aplicaciones en forma de applet.

Además, se debe disponer de un gestor de reservas para evitar el problema del acceso concurrente no permitido a los sistemas experimentales y habilitar espacios de tiempo razonables para la realización de las prácticas. Por ejemplo, divisiones de una hora de duración, de 45 minutos, etc. dependiendo de la dificultad de la práctica planteada por el profesor.

Teniendo en cuenta lo anterior, la integración completa de SARLAB en el LMS de la institución llevaría a tener que crear un módulo de actividades que facilite la integración de los clientes CONEXLAB y DIGEXLAB, así como de la interfaz de usuario, como componentes de una actividad didáctica. Es necesario además disponer de un módulo de gestión de reservas que integre los requerimientos específicos de la reserva de sistemas experimentales. Esto supone la coexistencia de tres aplicaciones que necesitan validar el acceso del usuario: el LMS, SARLAB y el gestor de reservas. La solución aportada por el modelo debe unificar las tres validaciones.

Por tanto, la integración completa de SARLAB en todas las versiones de los LMS actualmente en explotación en Internet solo sería abordable en el caso de que el código utilizado fuese público. Por otro lado, la integración completa de SARLAB en un único

LMS sería una solución parcial. Así, en el siguiente apartado se propone una solución en la que todas las funcionalidades necesarias para el desarrollo de actividades remotas controladas por SARLAB están soportadas en la tecnología web, accesible por tanto de forma indirecta desde cualquier LMS. Esta solución servirá de punto de partida para que los desarrolladores de cualquier plataforma puedan completar su total integración.

6.2.1.1. Integración de SARLAB en la tecnología web

La integración de SARLAB en la tecnología web, abordada en este apartado, hace que se pueda acceder a sus funcionalidades desde cualquier LMS de manera indirecta. Por tanto, esto constituye la base de la integración total de SARLAB en el LMS. La tecnología web se toma como marco de ejecución de la interfaz de usuario y de los clientes DIGEXLAB y CONEXLAB. La página de ejecución se denominará a partir de ahora *página de acceso a la experiencia*, e incluirá todas las funcionalidades propuestas por el modelo, y que son necesarias para la realización de una experiencia concreta. Para ello, las funcionalidades se integrarán en una aplicación en forma de applet de Java. De esta forma, y dado que todos los LMS permiten invocar llamadas a páginas web externas, se consigue la integración de forma indirecta de todas las funcionalidades de SARLAB en el LMS.

La estrategia propuesta dejaría en manos de la comunidad de usuarios y desarrolladores adaptar el *look and feel* de la página de acceso a la experiencia al correspondiente LMS. Es importante destacar que, al cargar en el navegador la página de acceso, el cliente DIGEXLAB necesita información para localizar el servidor SARLAB (*socket* del servidor), y para validar el acceso (usuario y contraseña). Finalmente, necesita saber a cuál de las experiencias quiere acceder el usuario. Esta información debe estar incluida en la página de acceso para transferirla al applet como parámetro, y evitar así que el usuario deba introducirla de nuevo. El código mostrado en la figura 6.1

presenta la forma en que se deben incluir el applet y sus parámetros en una página web de acceso.

```
<applet
code="ControlNivel1.classApplet"
codebase="/sarlab/jarfiles/"
archive="ControlNivel1.jar"
width="200" height="100">

<param name="permissions" value="all-permissions"/>
<param name="ipserver" value="150.214.163.74"/>
<param name="portserver" value="443"/>
<param name="idExp" value="Nivel 1"/>);
<param name="passwd" value="2err67781f270505b90oo95e6cd361440"/>
<param name="user" value="admin@http://sarlab2.uhu.es/sarlab"/>
</applet>
```

Figura 6.1: Código del applet en el LMS.

La siguiente etapa para conseguir la integración de SARLAB en el LMS mediante la tecnología web, consiste en encontrar una aplicación web que sirva de base para desarrollar la funcionalidad de gestión de las reservas de los sistemas experimentales. Para solucionar esta cuestión, se propone la elaboración del gestor de reservas de sistemas experimentales a partir del gestor de aulas, usando el software libre denominado MRBS (*Meeting Room Booking System*). Este software puede ser modificado para adaptarlo a las necesidades de gestión de sistemas experimentales. Dispone además de módulos de gestión de acceso que controlan la validación mediante distintas estrategias como LDAP, SMTP, POP3, IMAP, NIS, etc. Además existen módulos que permiten su integración completa en algunos LMS, como por ejemplo Moodle.

A modo de ejemplo, en la figura 6.2, se presenta la interfaz web de usuario modificada para la reserva de sistemas experimentales. En ella, el alumno puede acceder a las distintas áreas de conocimiento y en cada área se le oferta un conjunto de sistemas experimentales para los que puede reservar una hora del mes y día que desee.

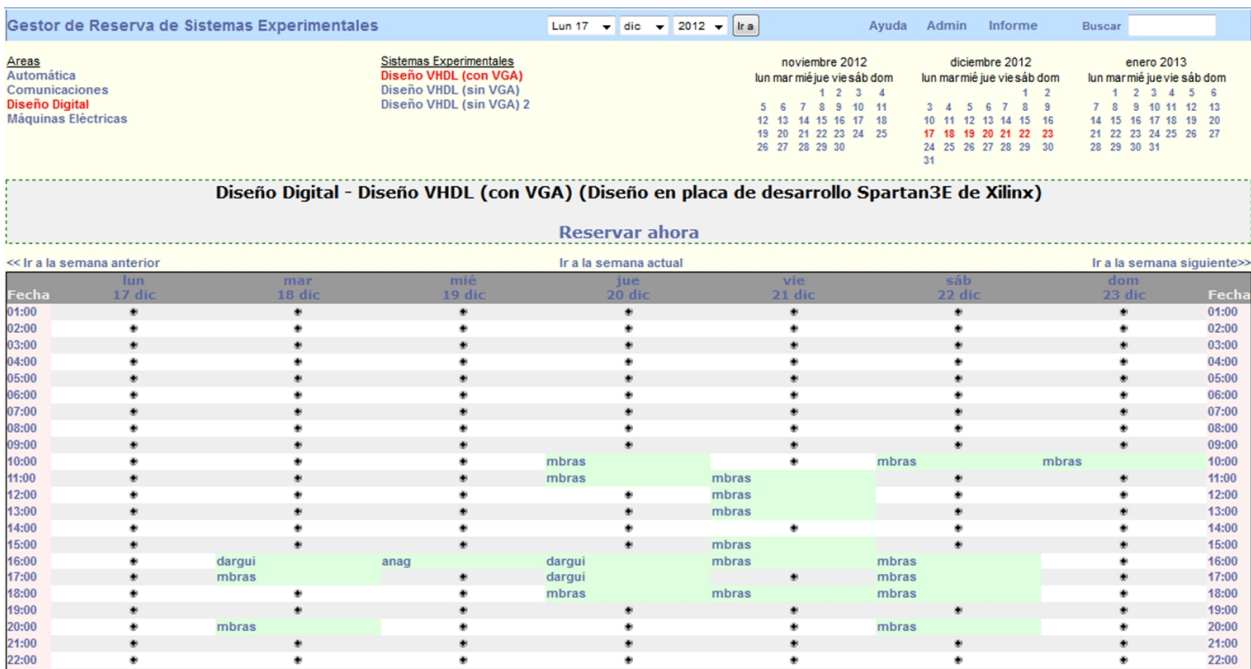


Figura 6.2: Interfaz del gestor de reservas.

Cada reserva de un sistema experimental bloquea el uso horario de éste para el resto de los alumnos, y además hace una entrada en la base de datos de reservas que será utilizada por DIGEXLAB para validar el acceso a SARLAB.

Cuando el alumno accede a una reserva, tiene acceso a todas las reservas y a realizar o borrar cualquiera de ellas, figura 6.3.

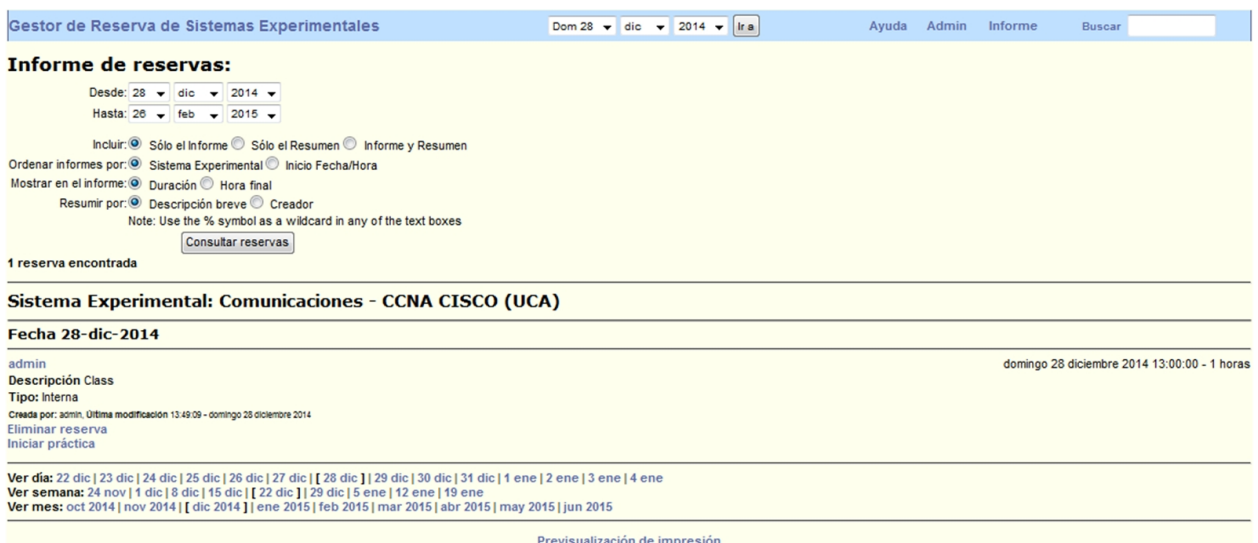


Figura 6.3: Listado de reservas de un usuario.

6. Implementación Funcional del Modelo basado en la Tecnología Disponible

Para administrar la gestión de reservas, además de permitir las altas, bajas y modificaciones de áreas de conocimiento y sistemas experimentales propios de la aplicación libre, se debe incluir la gestión de la información necesaria para la ejecución del applet. En la figura 6.4 se muestra la página de administración del gestor de reservas, a la que se puede acceder desde la línea superior del menú mostrado en la figura 6.3. Desde ella se pueden realizar las altas, bajas y modificaciones de las distintas áreas de conocimiento y sistemas experimentales ofertados, y se puede gestionar la información que se debe pasar al applet al crear la página web de acceso cuando el usuario pulsa la opción de iniciar práctica.

Gestor de Reserva de Sistemas Experimentales Lun 22 dic 2014 Ira Ayuda Admin Informe Buscar

Administración

Areas	Sistemas Experimentales en Comunicaciones	Experimentos
<ul style="list-style-type: none">Automática (Editar) (Borrar)Comunicaciones (Editar) (Borrar)Diseño Digital (Editar) (Borrar)Máquinas Eléctricas (Editar) (Borrar)	<ul style="list-style-type: none">CCNA CISCO (IESPPG) (Laboratorio remoto CCNA, 1) (Editar) (Borrar)CCNA CISCO (UCA) (Laboratorio remoto CCNA, 1) (Editar) (Borrar)CCNA CISCO (UHU) (Laboratorio remoto CCNA, 1) (Editar) (Borrar)	Sistema experimental: Bancada de motores Consultar
Agregar Area Nombre: <input type="text"/> Agregar Area	Agregar sistema experimental Nombre: <input type="text"/> Descripción: <input type="text"/> Capacidad (personas): <input type="text"/> Agregar sistema experimental	Agregar Experimento Denominación: <input type="text"/> Id experimento: <input type="text"/> IP SARLAB: <input type="text" value="150.214.163.74"/> Puerto SARLAB: <input type="text" value="443"/> URL experimento: <input type="text"/> Sistema experimental: <input type="text" value="Bancada de motores"/> Agregar experimento

Ver día: 16 dic | 17 dic | 18 dic | 19 dic | 20 dic | 21 dic | [22 dic] | 23 dic | 24 dic | 25 dic | 26 dic | 27 dic | 28 dic | 29 dic
Ver semana: 24 nov | 1 dic | 8 dic | 15 dic | [22 dic] | 29 dic | 5 ene | 12 ene | 19 ene
Ver mes: oct 2014 | nov 2014 | [dic 2014] | ene 2015 | feb 2015 | mar 2015 | abr 2015 | may 2015 | jun 2015

Figura 6.4: Página de administración del gestor de reservas.

Para finalizar la integración del gestor de reserva en la tecnología web es necesario resolver el problema de validación conjunta, que se estudiará en detalle en el apartado de la implementación de DIGEXLAB.

Esta solución tiene carácter general porque permite la integración de SARLAB en cualquier LMS, pero deja por resolver cuestiones como la validación conjunta de acceso o la unificación de la gestión del LMS, del gestor de reservas y de la integración de la

interfaz de usuario. Estas cuestiones deberán ser abordadas por los responsables de la integración completa de SARLAB en el LMS.

6.2.1.2. Integración completa de SARLAB en Moodle

A partir de la integración de SARLAB en el LMS mediante la tecnología web desarrollada en el apartado anterior, se propone en el presente apartado su evolución hasta la integración completa de SARLAB en un LMS en concreto: Moodle. En la elección del LMS del modelo propuesto se han tenido en cuenta los siguientes aspectos:

- ✓ Ser de código abierto.
- ✓ Ser ampliamente utilizado en Universidades.
- ✓ Disponer de módulo de gestión de reservas.
- ✓ Disponer de mecanismos de ampliación de sus funcionalidades para desarrollar el mecanismo de validación de DIGEXLAB.
- ✓ Haber sido usado anteriormente en la integración de experiencias de acceso remoto.

Moodle cumple ampliamente con los aspectos indicados. Además, en el momento de realizar esta Tesis, Moodle es un proyecto de los pioneros en el desarrollo de los LMS que se mantiene 100% de código abierto, frente a otros proyectos que están evolucionando a productos comerciales. Por otro lado, Moodle aparece entre los 10 primeros en la mayor parte de las estadísticas de valoración de los LMS que se pueden encontrar en Internet, y es una aplicación web que se puede ejecutar en cualquier sistema operativo que soporte lenguaje PHP, como Windows, Linux y la mayoría de los proveedores de hostings webs. Soporta además una gran variedad de bases de datos. En resumen, entre sus principales características se pueden citar las siguientes:

6. Implementación Funcional del Modelo basado en la Tecnología Disponible

- ✓ Es fácil de instalar y actualizar.
- ✓ Es adaptable y flexible y dispone de API de desarrollo (en código abierto).
- ✓ Es compatible con la mayoría de los sistemas operativos.
- ✓ Es intuitivo y por tanto facilita su uso por parte de alumnos y profesores.
- ✓ Hay más de 80 millones de usuarios distribuidos en cuarenta y siete mil sitios web, y se ha traducido a 75 idiomas, lo que convierte a esta plataforma en la más usada, en opinión de muchos.
- ✓ Dispone de varios gestores de reservas, aparte del MRBS. Entre ellos, destaca uno realizado por la UNED, específico para gestionar experiencias remotas.
- ✓ Dispone también en su API de funciones que permiten acceder a los mecanismos de autenticación que facilitan la validación conjunta.

Además de todos los aspectos positivos mencionados en el párrafo anterior, en el momento de desarrollar esta Tesis, el profesor de la UNED Dr. D. Luis de la Torre Cubillo estaba realizando la reconversión de la plataforma Emersion a Moodle, y estaba trabajando en el desarrollo de módulos que permitiesen la integración de experiencias con acceso remoto desde Moodle. En efecto, la colaboración con el profesor de la Torre ha permitido la integración total de SARLAB en Moodle de la forma descrita a continuación.

El prof. de la Torre ha desarrollado un módulo para la integración de EJS en Moodle (EJSApp) y otro para la integración de un gestor de reservas (EJS booking) que colaboran entre ellos. El EJSApp integra aplicaciones desarrolladas en EJS para la realización de experiencias remotas y simulaciones. El módulo EJS booking gestiona específicamente las experiencias de laboratorio y colabora con EJSApp para asistir al diseñador en el proceso de creación de una experiencia. Nótese que la simple

introducción de una actividad EJSApp conlleva modificaciones en el gestor de reservas para que el alumno pueda reservar el uso de la correspondiente experiencia. Además, su cuidada elaboración permite al profesor seleccionar los alumnos que tienen acceso a cada experiencia de una forma muy cómoda. La larga experiencia de los desarrolladores de estas aplicaciones y el trabajo de colaboración entre ellos han permitido la inclusión de SARLAB. Ésta se limita a parametrizar las comunicaciones en la página de configuración del módulo EJS App, figura 6.5. En esta página se especifican los sockets de todos los servidores SARLAB desde los que se ofertan experiencias en el LMS (Moodle) correspondiente. Además, se facilita una clave para encriptar las comunicaciones que se realicen entre el servidor Moodle y DIGEXLAB al validar el acceso a una experiencia.



Figura 6.5: Página de configuración del EJSApp donde se incluyen los parámetros de SARLAB.

Una vez configurado SARLAB en el EJSApp, la oferta desde Moodle de una nueva experiencia remota controlada por SARLAB se limita a añadir un recurso EJSApp,

6. Implementación Funcional del Modelo basado en la Tecnología Disponible

figura 6.6. A continuación, se parametriza la experiencia indicando si su acceso se controla o no mediante SARLAB y, en caso afirmativo, también se parametriza qué servidor SARLAB controla su acceso, si se permite el acceso colaborativo o no y finalmente cual es el identificador de la experiencia, figura 6.7.

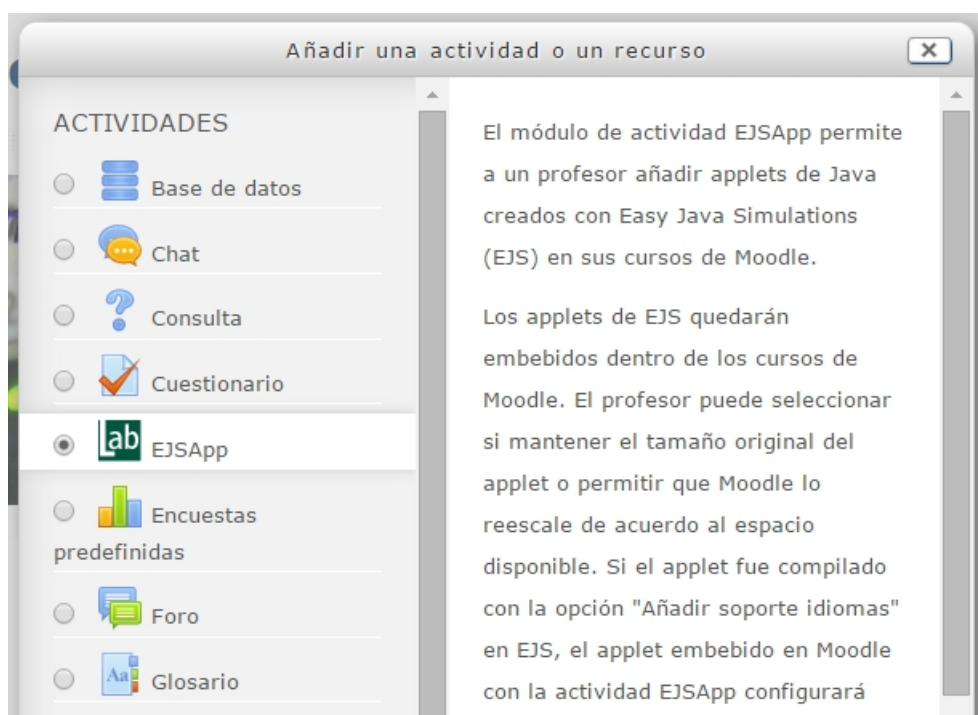


Figura 6.6: Inclusión de una nueva experiencia remota en Moodle controlada por SARLAB.

Como se ha indicado anteriormente, el módulo EJSApp está relacionado con el de gestión de reservas de forma que la definición de una nueva experiencia lleva asociada la correspondiente alta de forma automática en éste último. Asimismo, la inclusión del gestor de reservas en un curso permite la gestión de todas las experiencias definidas en éste de manera automática. Además, EJS booking dispone de mecanismos para la selección de los usuarios a los que se les oferta cada experiencia de entre los alumnos matriculados en un curso. Por otro lado, EJSApp se comunica con SARLAB para obtener el conjunto de identificadores de experiencias y verifica si todos los componentes de una experiencia son accesibles. Además, en el caso de detectar un fallo

en las comunicaciones, envía un correo electrónico al administrador de la experiencia con un aviso de la incidencia. Estos dos módulos (EJSApp y EJS Booking) se pueden descargar del repositorio oficial de Moodle en las siguientes direcciones:

- ✓ EJSApp: https://moodle.org/plugins/view.php?plugin=mod_ejsapp
- ✓ EJS Booking: https://moodle.org/plugins/view.php?plugin=mod_ejsappbooking

Lab Exp Rem Español - Internacional (es) Mis cursos

Personalizar variables del laboratorio EJS

Archivo .exp con el experimento que la aplicación EJS debe ejecutar al cargarse

Configuración del laboratorio remoto

Sistema experimental remoto? Sí

Usar Sarlab? Sí

Servidor Sarlab para este laboratorio? SARLAB UHU

Usar acceso colaborativo de Sarlab? No

Identificador de práctica en Sarlab? Experiencia nula, Demo, Demo AR1, Demo AR LAB DIG

Figura 6.7: Configuración del servidor SARLAB y el identificador de experiencias.

Con el procedimiento descrito se consigue la integración completa de SARLAB en Moodle. Esta es la solución más recomendable para facilitar el diseño de las experiencias de acceso remoto según el modelo propuesto en esta Tesis. Sin embargo, SARLAB, en su línea de ser una solución general del modelo, deja abierta la posibilidad de integración en otros LMS con la solución intermedia que se planteó en el apartado anterior.

Una vez desarrolladas las distintas alternativas propuestas por el modelo para integrar SARLAB en el LMS, queda resuelto el lazo de gestión experimental.

6.3. Esquema genérico del lazo de desarrollo experimental controlado por SARLAB

En el capítulo anterior se presentaron los dispositivos físicos que conforman el lazo de desarrollo experimental del modelo para las diferentes alternativas propuestas como solución a este lazo, dependiendo de las dificultades funcionales del sistema experimental. En este apartado se presenta el esquema físico que relaciona todos esos componentes para establecer el lazo de desarrollo experimental.

El lazo de desarrollo experimental se establece entre el extremo de la comunicación en el que se encuentra un alumno conectado a Internet a través de un PC, en el que se ejecuta la interfaz de usuario, y el extremo en el que se encuentra el sistema experimental convergente. Dado que el alumno se conecta a red a través de Internet y el sistema experimental convergente está conectado a la intranet de la institución que oferta la actividad, el lazo de desarrollo experimental se debe cerrar a través de SARLAB, que actúa de frontera entre las dos redes, figura 6.8. Este esquema permite además que los sistemas experimentales controlados por SARLAB estén repartidos por todas las instalaciones de la institución conectadas a una misma VLAN. En la figura 6.8 se presenta además el subsistema de alimentación que controla la alimentación del sistema experimental convergente, así como un pequeño esquema que presenta la implementación física del lazo con su diagrama de relación, establecido en el modelo conceptual general presentado en el capítulo 3.

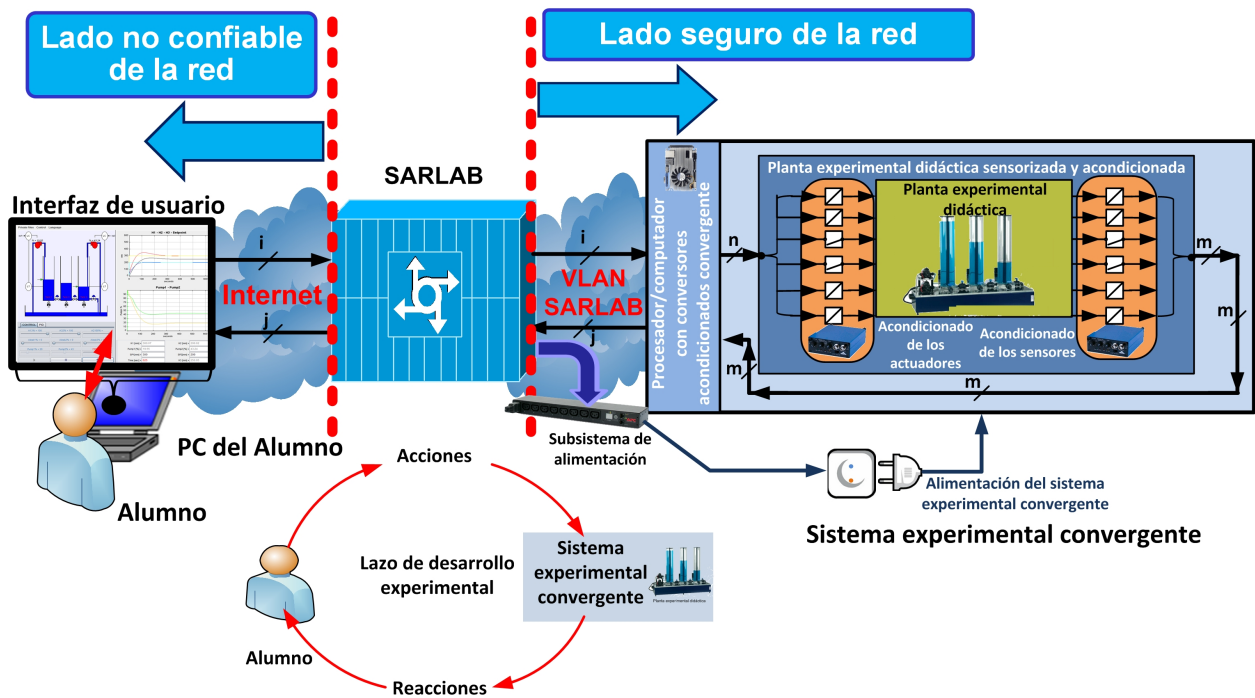


Figura 6.8: Esquema físico genérico del lazo de desarrollo experimental.

Según se establece en la figura 6.8, el diseño de una experiencia controlada por SARLAB precisa del desarrollo del sistema de convergencia genérico, de la creación de la correspondiente interfaz de usuario, de la definición en el servidor SARLAB de las conexiones necesarias para establecer el lazo de desarrollo experimental y de la configuración del subsistema de alimentación. De esta forma, se dispone de un canal de comunicación abierto entre el usuario y el sistema experimental. Teniendo en cuenta que cada sistema experimental plantea unas necesidades de procesamiento propias de seguridad, control, etc., es necesario decidir si ese procesamiento hay que implementarlo en el lado seguro de la red (intranet) o se puede trasladar a su lado no confiable (Internet). Esto determinará el sentido de la información que fluya por el canal abierto.

6.4. Unificación de las comunicaciones

En este punto, el problema de las comunicaciones está totalmente resuelto con respecto a la conectividad, pero no lo está con respecto al intercambio de información entre cliente y servidor. El desarrollo de la estructura cliente-servidor es una cuestión más que debe resolver el modelo tecnológico general y que, en primera instancia, llevaría a la implementación a bajo nivel de un protocolo de aplicación de intercambio de información. Esta solución sería inabordable por diseñadores con poca experiencia en el desarrollo de aplicaciones de red, lo que contradice doblemente la filosofía de desarrollo del modelo propuesto en esta Tesis: por un lado, el modelo promueve la asistencia en el diseño de experiencias para profesores no expertos en comunicaciones y por otro aboga por la programación al más alto nivel. En efecto, ese protocolo, entre otras cosas, debería soportar accesos en concurrencia para facilitar el trabajo colaborativo de los alumnos, lo que lo convierte en un problema todavía más complejo. Además, el desarrollo de protocolos fuera de un estándar dificulta el mantenimiento y la verificación de su funcionamiento. Así, en un escenario en el que pueden existir múltiples sistemas conectados a una red, a través de la cual todos intercambian información, se impone de manera natural la unificación de los protocolos de comunicaciones. En este sentido, aunque el protocolo TCP/IP unifica las comunicaciones a nivel de transporte, es necesario unificarlas también a nivel de aplicación. Para ello, se propone en la implementación del modelo tecnológico general de esta Tesis la utilización del estándar abierto de uso en comunicaciones industriales ModBus.

La unificación de las comunicaciones mediante el uso de ModBus no es un condicionante de funcionamiento de SARLAB. Esto es, SARLAB puede funcionar en un escenario en el que no esté unificado el protocolo de aplicación de comunicaciones del laboratorio, o en el que se utilice cualquier otro protocolo para realizar la unificación. Ahora bien, al ser la solución propuesta para el desarrollo de la estructura cliente-

servidor de este modelo, el desarrollador que opte por otra opción, no podrá utilizar las aplicaciones expuestas a lo largo de este capítulo.

El protocolo Modbus es un protocolo maestro-esclavo. Por tanto, las comunicaciones se inician en el maestro, que se encarga de dar órdenes a los esclavos. Los esclavos se limitan a cumplir las órdenes recibidas desde los maestros. Este modo de funcionamiento obliga a situar los esclavos en los sistemas experimentales y el maestro en la interfaz de usuario, figura 6.9.

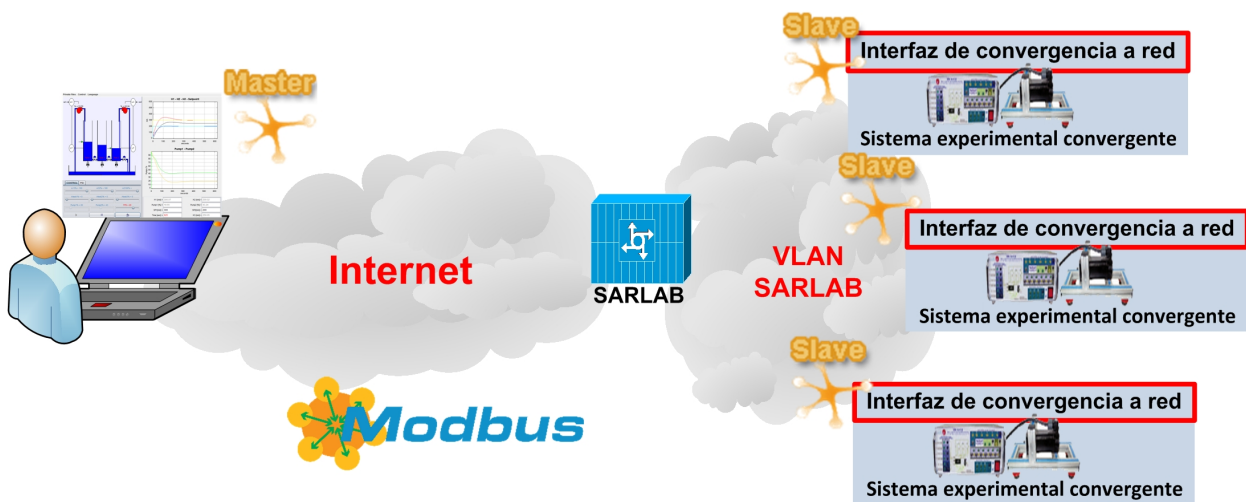


Figura 6.9: Estructura de comunicaciones unificada mediante protocolo ModBus.

El uso de ModBus en los laboratorios gestionados por SARLAB plantea las siguientes ventajas:

- ✓ En un escenario unificado de comunicaciones *todos pueden hablar con todos* en el modelo maestro-esclavo, característica impensable en un laboratorio en el que múltiples diseñadores desarrollen estructuras cliente-servidor de forma independiente.
- ✓ Se pueden utilizar aplicaciones de comunicaciones M2M (*machine to machine*) para realizar las conexiones de los componentes del laboratorio (control topológico).

6. Implementación Funcional del Modelo basado en la Tecnología Disponible

- ✓ Se facilita el diseño de experiencias, ya que el método de comunicaciones es idéntico en todos los casos.
- ✓ Se permite la utilización de componentes industriales como PLCs, controladores digitales, etc., que soporten el estándar de comunicaciones Modbus.
- ✓ Se permite el uso de monitores estándar para acceder a los sistemas experimentales convergentes y verificar su funcionamiento.
- ✓ Existen simuladores que permiten la depuración de aplicaciones.
- ✓ El desarrollo de la aplicación de comunicaciones se realiza a alto nivel.

La capa de aplicación que proporciona Modbus permite cumplir con las necesidades de intercambio de información entre la interfaz de usuario y el sistema experimental convergente, de manera casi transparente al uso de la red, ya que el diseñador se limita a realizar llamadas desde el máster a métodos que realizan operaciones de lectura o escritura en la imagen de proceso de los esclavos, sin tener que realizar una gestión directa del establecimiento, desarrollo y fin de las comunicaciones. Esta capa de software oculta los detalles de la tecnología de comunicaciones, lo que facilita el desarrollo de aplicaciones a usuarios que no tengan conocimiento sobre el desarrollo de aplicaciones en red.

6.5. Herramienta para el desarrollo rápido de la interfaz de usuario

Una vez estudiada la estructura del lazo de desarrollo experimental, se aborda en este apartado la elección de la herramienta de diseño de la interfaz de usuario de las experiencias de acceso remoto indicada en el capítulo 4. Para ello, se seguirá

considerando el criterio de usar aplicaciones de código abierto que comprendan las funcionalidades necesarias y su adaptación a los requerimientos específicos del modelo.

La integración de SARLAB en el LMS condiciona el uso de Java como lenguaje para generar los applets a incluir en las páginas web. Esta primera condición la cumplen muchas aplicaciones que pueden ser elegidas como herramienta de desarrollo rápido, por ejemplo, OpenXava, EJS, Processing, MyOpenLab, etc.

Sin embargo, hay otras condiciones que debe cumplir esta herramienta. Por ejemplo, debe facilitar la creación de aplicaciones gráficas, lo que implica que incluya las siguientes utilidades:

- ✓ Componentes para diseñar formularios como botones, entradas de datos, deslizadores, áreas de texto, etc.
- ✓ Componentes para la representación de datos y para su tratamiento matemático.
- ✓ Componentes de presentación de datos multimedia como gráficos, video, sonido, etc.
- ✓ Componentes de representación de elementos virtuales en 2D y 3D.
- ✓ Componentes para manejar recursos de realidad aumentada.

EJS incluye todos esos componentes, además de disponer de un entorno de integración de sus desarrollos en la plataforma Moodle, el módulo EJSApp, que, como se ha comentado en el apartado 2, ha sido creado por desarrolladores de la UNED.

EJS es una herramienta de código abierto que fue desarrollada para realizar simulaciones matemáticas por computador. Posteriormente, fue usada en el proyecto Automat@Labs para unificar la interfaz de usuario de las experiencias remotas.

6. Implementación Funcional del Modelo basado en la Tecnología Disponible

Asimismo, EJS puede ser usada en el modelo en dos vertientes distintas: para el diseño de sistemas experimentales convergentes y/o para el desarrollo de la interfaz de usuario de la experiencia.

Esta herramienta presenta dos modos de funcionamiento: diseño y ejecución. En el modo diseño, la herramienta permite desarrollar las aplicaciones definiendo dos bloques denominados *Modelo* y *Vista*. El bloque *Vista* define la interfaz gráfica con la que interactúa el alumno para controlar e interpretar la evolución de la experiencia. Se crea arrastrando y soltando los componentes gráficos necesarios, como ventanas, paneles, botones, entradas de datos, etc., figura 6.10.

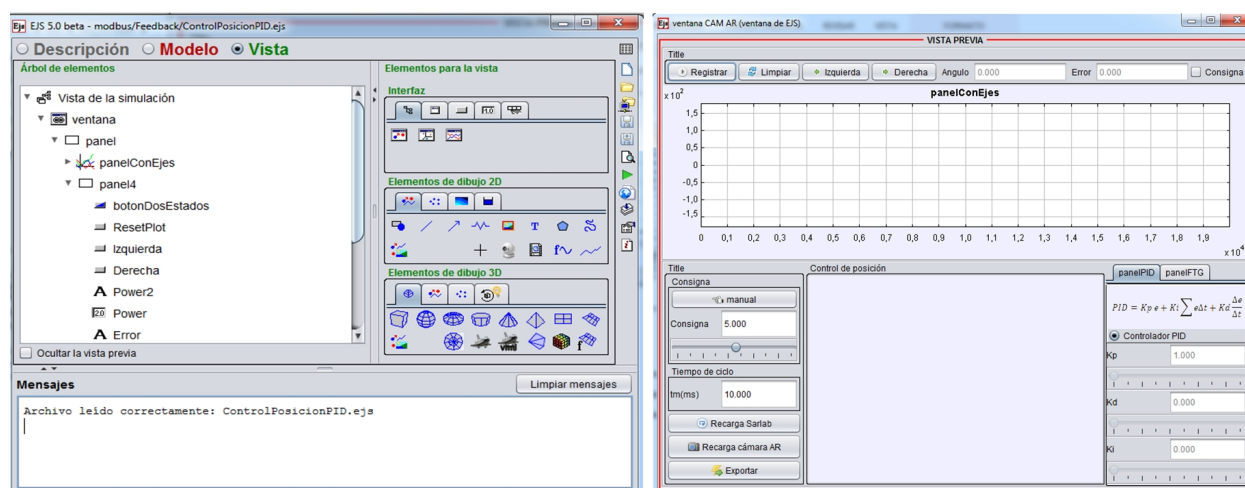


Figura 6.10: Interfaz de EJS. A la izquierda se muestra la interfaz de diseño y a la derecha el resultado.

Cada componente gráfico de la interfaz dispone de la posibilidad de capturar y gestionar las acciones realizadas sobre él por el usuario al mover el ratón, pulsar el teclado, etc. De esta forma, la interfaz puede reaccionar ante las acciones del usuario y comunicarlas al sistema experimental.

En el bloque *Modelo* se puede definir cualquier tipo de variable que haga referencia a una clase de Java. Además, dispone de un apartado donde pueden definirse los procedimientos necesarios para la inicialización de una experiencia y otro apartado,

denominado evolución, donde se define el procedimiento que se repite periódicamente en la ejecución de la experiencia. Todos estos procedimientos serán desarrollados en Java. EJS dispone de un gran número de funciones que ayudan a desarrollarlos y mejoran la organización del código con el uso de páginas etiquetables, figura 6.11.

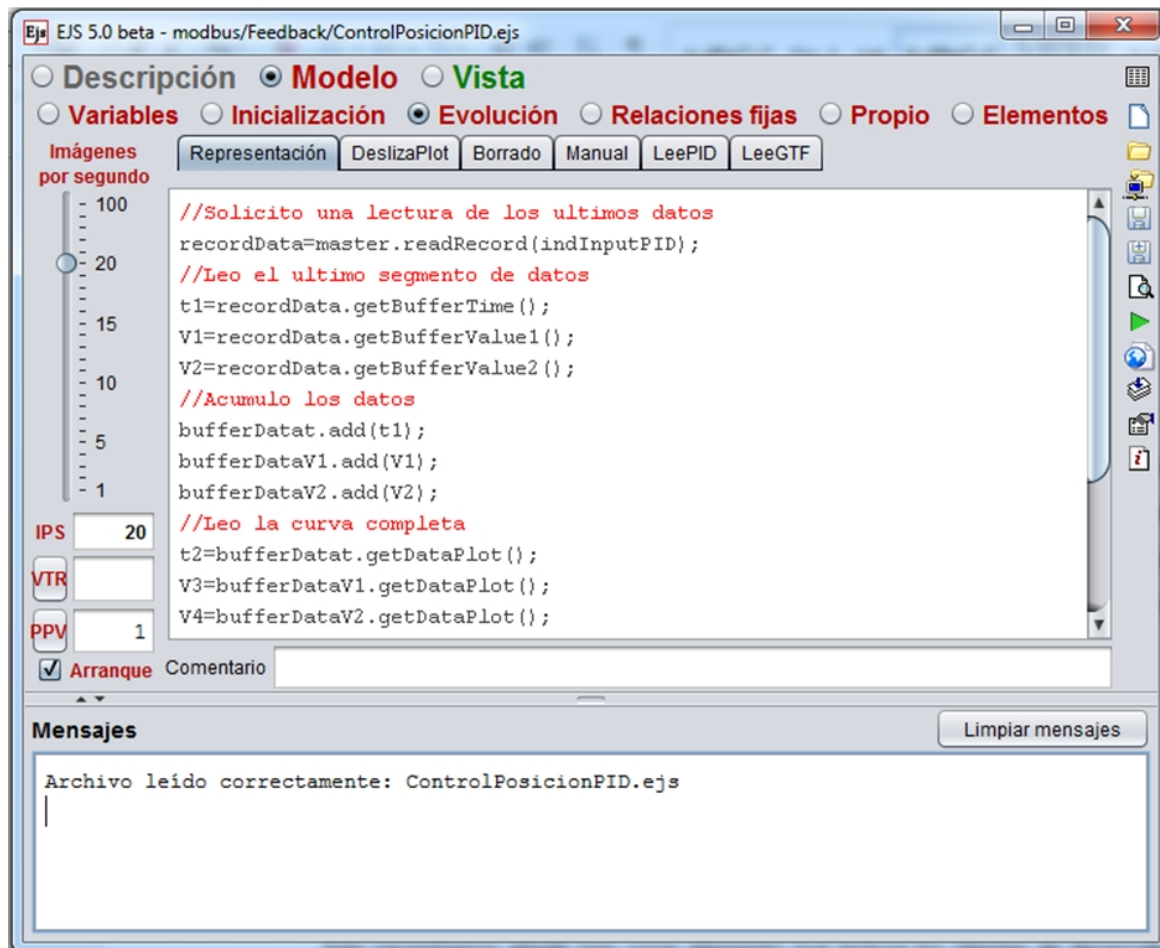


Figura 6.11: Interfaz de desarrollo de procedimientos de EJS.

EJS permite el empaquetamiento de la interfaz en un fichero Java ejecutable (jar) que incluye una clase applet de Java. Esto permite su integración en páginas web, o su integración en una actividad de Moodle a través de su módulo EJSApp. La ejecución en línea del fichero jar permite también utilizar la interfaz sin necesidad del navegador, lo que abre la posibilidad de usarla fuera de la tecnología web.

6. Implementación Funcional del Modelo basado en la Tecnología Disponible

EJS cuenta también con un importante número de funciones matemáticas que la convierte en un excelente motor matemático. De esta forma, el diseñador de la experiencia puede hacer uso de estas funciones matemáticas para procesar la información de las experiencias, y el usuario puede interpretar más fácilmente los resultados de la misma.

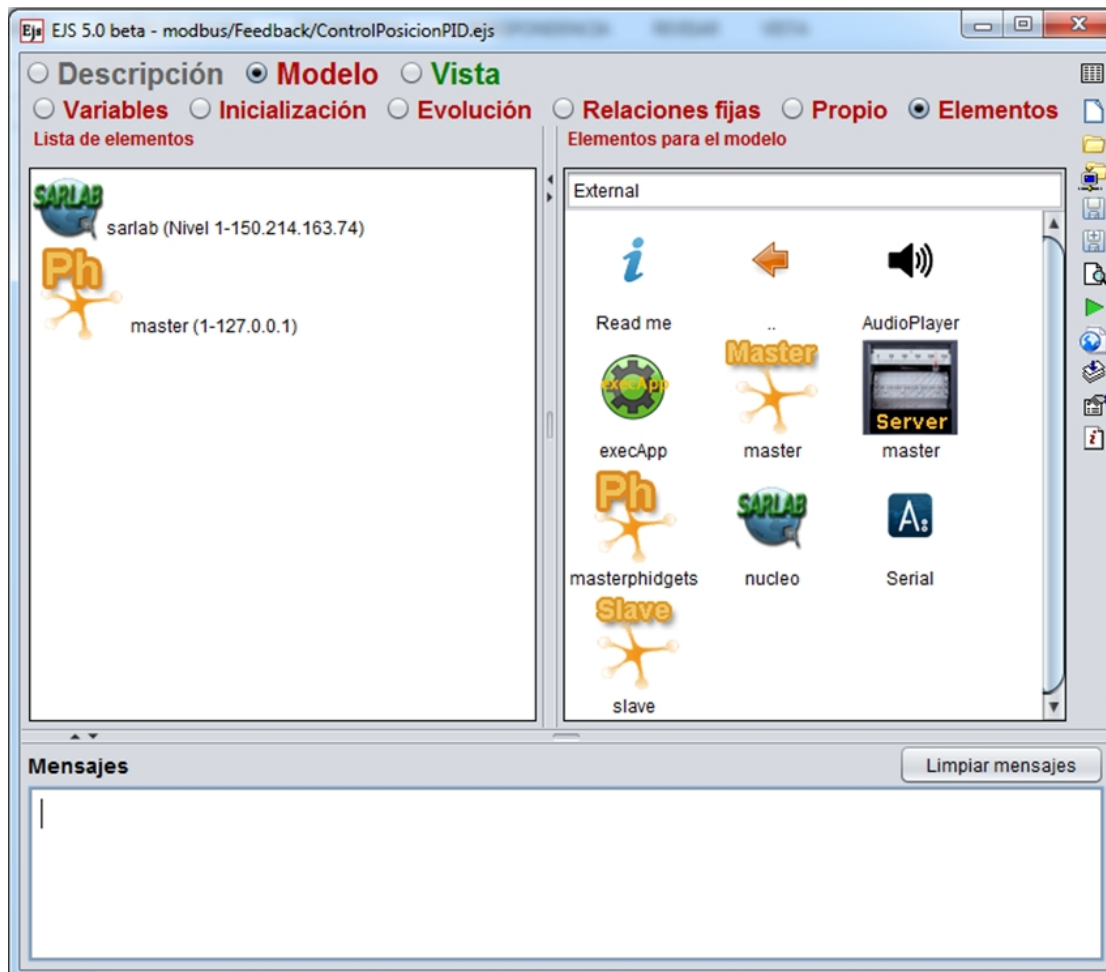


Figura 6.12: Interfaz de gestión de los elementos en EJS.

Asimismo, la herramienta permite crear funcionalidades nuevas añadiendo código propio escrito en Java. En este sentido, la característica fundamental en la elección de EJS como herramienta de diseño rápido es que dispone de mecanismos para integrar fácilmente los desarrollos realizados en Java. Entre estos desarrollos se encuentran

componentes hardware nuevos que se denominan *elementos* y que se incluyen en las experiencias arrastrando y soltando. Esta característica permite incluir todos los desarrollos de la Tesis en EJS como elementos nuevos. Así, al empaquetar la interfaz de usuario, se estarán empaquetando también los clientes de SARLAB y todos los componentes necesarios para conseguir el acceso por red a los sistemas experimentales. Por tanto, usando el módulo EJSApp, la interfaz con el sistema experimental puede ser totalmente integrada en Moodle, y con ella SARLAB.

En la figura 6.12 se presenta la interfaz de gestión de los elementos. A la izquierda figuran los elementos incluidos en la interfaz de usuario y a la derecha algunos de los creados en el desarrollo del modelo propuesto en esta Tesis. La inclusión de uno nuevo se limita a arrastrarlo y soltarlo desde la derecha a la izquierda y a su parametrización.

Para conseguir que EJS pueda ser la herramienta de desarrollo rápido de la interfaz de usuario propuesta en el modelo, ha sido necesario elaborar un conjunto de elementos que permitan integrar las siguientes funcionalidades:

- ✓ Acceso a DIGEXLAB y CONEXLAB.
- ✓ Intercambio unificado de información de la interfaz con los sistemas experimentales.
- ✓ Gestión de la latencia de Internet en el envío de datos.
- ✓ Gestión y mantenimiento de las experiencias situadas en el laboratorio.

Estos elementos facilitan el trabajo del diseñador a nivel de comunicaciones porque su función se limita a incluir en la interfaz de usuario el elemento SARLAB encargado del establecimiento de los túneles, y otros elementos necesarios para establecer las comunicaciones con el sistema experimental, figura 6.13. Esto es, para realizar la

6. Implementación Funcional del Modelo basado en la Tecnología Disponible

conexión entre la interfaz de usuario y el sistema experimental convergente solo será necesario:

- ✓ Arrastrar y soltar el elemento SARLAB y configurarlo con la dirección IP, el puerto del servidor SARLAB y el identificador de experiencia a la que se quiere acceder.
- ✓ Arrastrar y soltar el módulo maestro de Modbus, y configurarlo para acceder al correspondiente punto de entrada del túnel que conectará en el otro extremo con el sistema experimental convergente.

El uso de ModBus para unificar las comunicaciones apunta la existencia de un esclavo ModBus en el sistema experimental, que se limita a mantener actualizada la imagen del proceso en dicho sistema. En esta imagen del proceso, el esclavo almacena el valor de los sensores en cada instante y aquellos otros valores que condicionan el estado de los actuadores. Por tanto, para conocer el valor de un sensor solo es necesario invocar el método de lectura de la imagen del proceso correspondiente desde el máster, y para modificar el estado de un actuador solo hay que escribir en una determinada posición de la imagen del proceso el nuevo valor de excitación del mismo. Por tanto, la unificación del protocolo de aplicación, o más concretamente, el uso de Modbus, ha permitido ocultar al usuario todas las cuestiones relacionadas con la programación de la red de comunicaciones.

En resumen, el uso unificado de las comunicaciones en la interfaz de usuario simplifica el diseño de la experiencia, tan solo incluyendo los elementos SARLAB y el máster Modbus, ambos convenientemente configurados. Además, cada experiencia necesitará un diseño particular para la interfaz de usuario, que proporcione la información necesaria para que el alumno pueda percibir la evolución de la experiencia y los componentes gráficos que permitan modificarla, como por ejemplo botones que activen señales, deslizadores que modifiquen las consignas de los controles, líneas de

entradas de valores que condicionen la evolución de algún componente de la experiencia, etc.



Figura 6.13: Elección de los elementos de establecimiento de las comunicaciones en EJS.

6.6. Desarrollo de CONEXLAB

En este apartado se propone la solución para el desarrollo de las funcionalidades del modelo tecnológico general atribuidas a CONEXLAB sobre el protocolo TCP/IP. Esta funcionalidad es la encargada de establecer las conexiones entre el PC del alumno y el sistema experimental convergente. El modelo tecnológico general proponía como solución óptima el uso de túneles para establecer las conexiones y por tanto CONEXLAB debe adoptar la estructura de servicio de túneles establecidos sobre TCP/IP.

El procedimiento para desarrollar esta funcionalidad será el que se ha venido siguiendo hasta ahora: buscar siempre un desarrollo de software abierto que se aproxime lo más posible al objetivo marcado y hacer las adecuaciones necesarias. Esto conduce, en el caso de CONEXLAB, a utilizar las aplicaciones realizadas en Java para implementar el protocolo estándar de túneles. En efecto, Java incorpora de forma nativa clases que permiten el desarrollo de los dos protocolos de túneles utilizados sobre

TCP/IP, los protocolos SSL y SSH. Estos protocolos incorporan, además de la posibilidad de encapsular tráfico en forma de túneles, la capacidad de cifrar las comunicaciones, lo que le proporciona un alto nivel de seguridad a las comunicaciones entre el usuario y el laboratorio.

Como solución general al problema, cualquiera de estos dos protocolos proporcionaría la funcionalidad principal de CONEXLAB, ya que ambos son capaces de realizar las redirecciones del tráfico de paquetes entre un cliente que esté conectado a una red TCP/IP y el correspondiente servidor conectado a esa misma red. Además, una vez alcanzado el servidor, los paquetes pueden continuar camino a través de cualquiera de las redes internas a él conectadas para alcanzar el equipo destino. Por tanto, el uso de alguno de estos protocolos proporciona tanto la aplicación cliente como la aplicación servidora, y lo único que resta es redefinir los métodos que no se ajusten estrictamente a las necesidades del modelo, y añadir aquellos otros que sean necesarios y no existan. Estos protocolos disponen de métodos en los que no hay más que especificar el *socket* de entrada en el equipo donde esté el cliente y el *socket* de destino con el que se cursará el tráfico de salida por el servidor. Cualquier tráfico de paquetes que se dirija en el cliente a estos *sockets* de entrada se traslada al servidor de forma encriptada a través de Internet. Una vez en el servidor, éste redirige el tráfico al *socket* destino en alguna de sus redes internas, ya descriptado.

El aprovechamiento de la capacidad de cifrado de estos protocolos hace posible que todo el tráfico entre las distintas funcionalidades (CONEXLAB, DIGEXLAB, etc.) sea cifrado, al igual que el tráfico de evolución de las experiencias. Para que todo el tráfico sea cifrado de forma unificada, es necesaria una ligera modificación en el esquema final de modelo tecnológico general, que se representa en la figura 6.14. En esta figura se observa que las únicas modificaciones que se han realizado con respecto de la figura 4.21, de la cual procede, son la inclusión de los protocolos SSL o SSH en la arquitectura de CONEXLAB, y la extensión de la capa de cifrado a todas las comunicaciones.

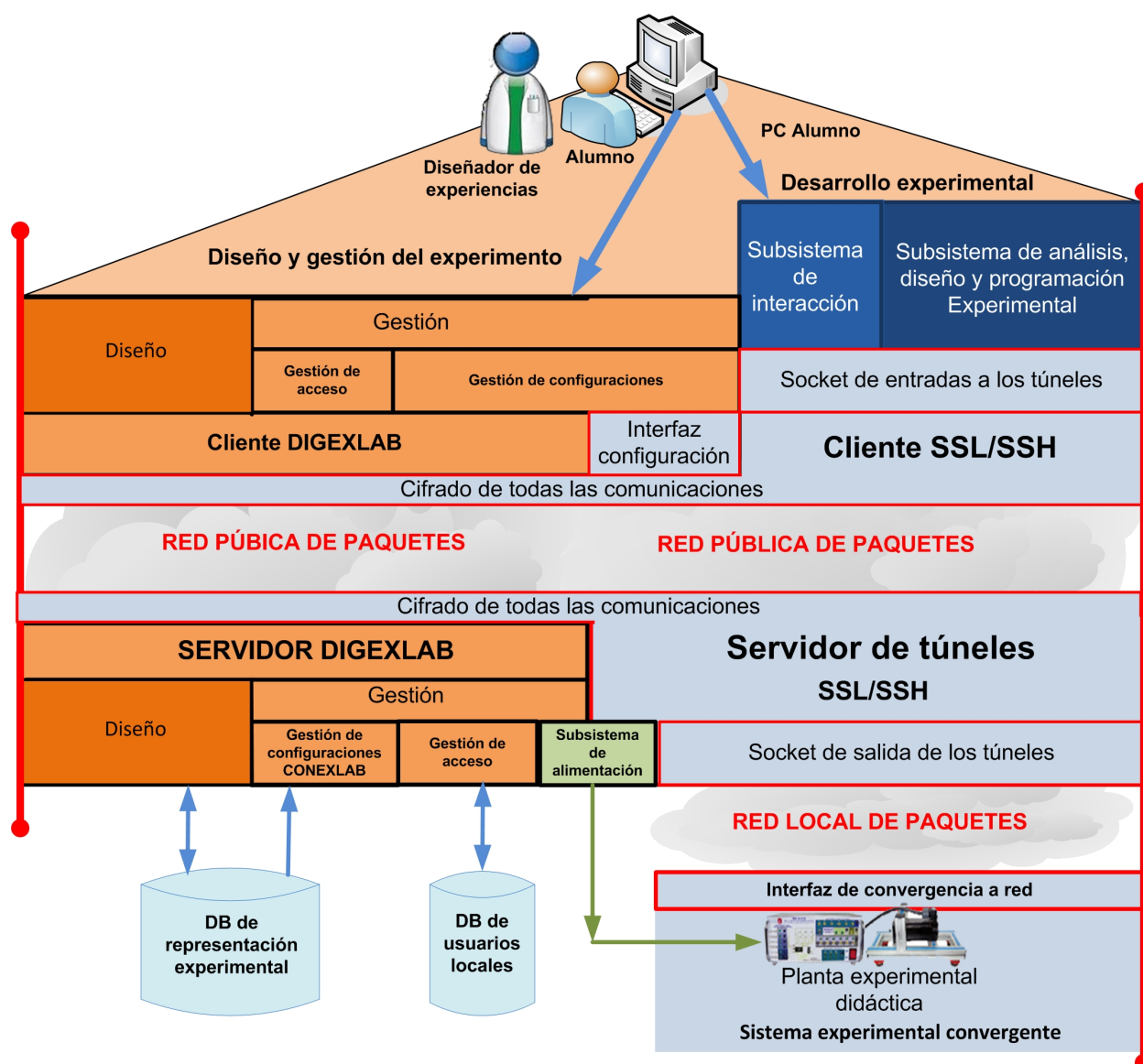


Figura 6.14: Modelo tecnológico general que incluye la tecnología de túneles con protocolo SSL/SSH.

El uso de los protocolos SSL o SSH aporta la funcionalidad fundamental de CONEXLAB de creación de túneles, tanto a nivel de cliente como de servidor. Además, el uso de estos protocolos estándares en la implementación del modelo aporta la estabilidad y fiabilidad de un protocolo ya depurado y estable. Aunque se han construido prototipos con los dos protocolos, finalmente se ha optado por SSH para la implementación del modelo. Este protocolo dispone de un abundante número de proyectos de software libre como por ejemplo OpenSSH, Ganymed-ssh-2, JCraft, Hierynomus etc., que aportan gran cantidad de información y librerías para facilitar la

implementación del protocolo estándar, tanto del cliente como del servidor, totalmente en lenguaje Java.

El problema de diseño queda pues limitado a estudiar las modificaciones a realizar en el cliente y el servidor SSH para incluir las funcionalidades de DIGEXLAB y CONEXLAB, sin comprometer la estabilidad de este protocolo. En la figura 6.15 se ha representado el diagrama SDL simplificado del protocolo SSH a nivel de cliente y servidor, y los estados y procesos fundamentales. En ella se observa cómo el servidor está en estado de espera de conexión y, una vez que se establece una conexión, crea un hilo para atender la solicitud de servicio del cliente y vuelve a esperar nuevas conexiones. Por su parte el cliente se limita a iniciar la conexión que, una vez establecida, inicia el procedimiento de negociación de cifrado de las comunicaciones con el servidor. Si este proceso tiene éxito, se inicia el establecimiento de una sesión SSH con el servidor. En caso de no poder iniciar el cifrado, la conexión con el servidor se cierra. En el servidor, el hilo iniciado se encarga de la negociación del cifrado y, si es exitosa, de poner en marcha el proceso de inicio de sesión SSH con el cliente.

Una vez establecido el cifrado de las comunicaciones, se inicia el proceso de establecimiento de una sesión SSH entre el cliente y el servidor. En la figura 6.16 se ha representado el diagrama SDL simplificado del proceso de establecimiento de una sesión entre un cliente y un servidor SSH. En el servidor, el hilo de ejecución que atiende la conexión del cliente está en estado de espera de usuario y contraseña. Cuando se reciben estos datos, se procede a la validación y se informa al cliente de su resultado. Si la validación es positiva, se inicia el gestor de sesión. En caso contrario se cierra la conexión con el cliente. Por su parte, el cliente se encarga de enviar usuario y contraseña. Cuando lo hace, pasa al estado de espera de validación. Si la validación es positiva, se inicia el gestor de sesiones. Tanto en el cliente como en el servidor, el gestor de sesiones se ejecuta hasta que se recibe un evento que indique el fin de la sesión.

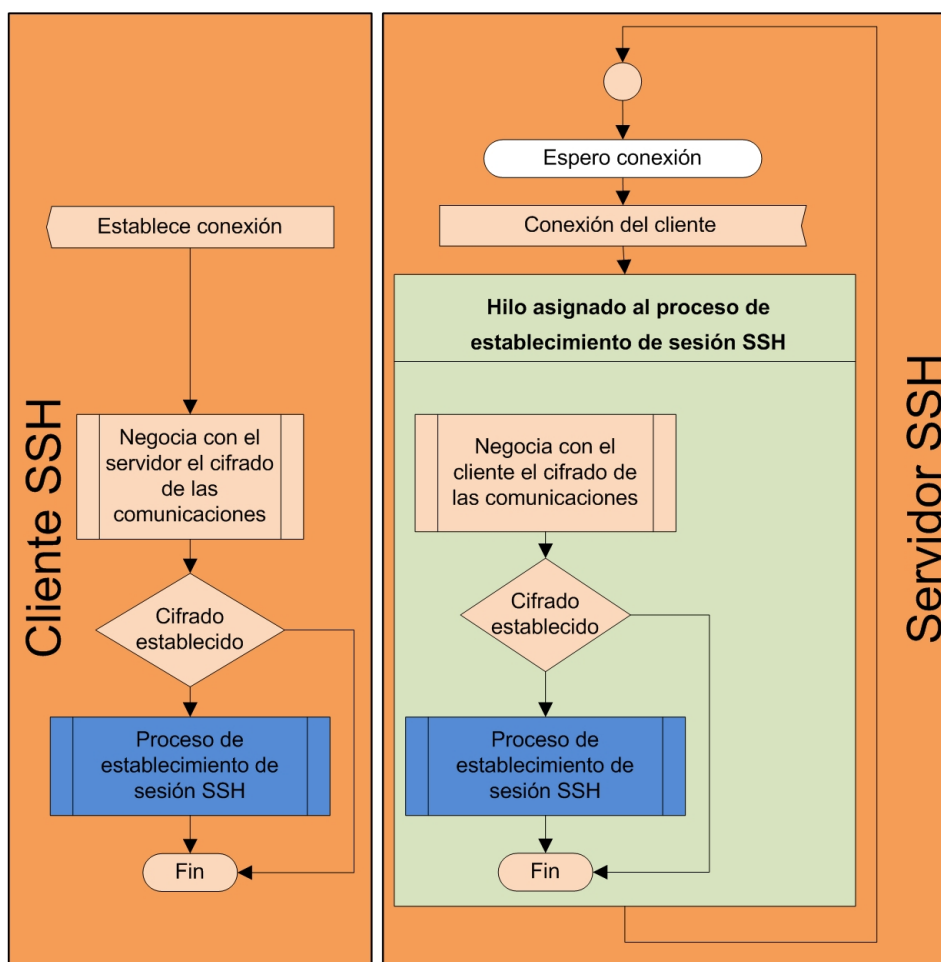


Figura 6.15: Diagrama SDL simplificado del protocolo SSH.

Una vez validada la sesión entre el cliente y el servidor, queda establecido un canal de comunicaciones único por el que se pueden definir las redirecciones necesarias para el desarrollo de una experiencia, o se puede intercambiar información entre cliente y servidor. Como se observa en el diagrama de relación de CONEXLAB presentado en el capítulo 4, el primer paso para iniciar las funciones de CONEXLAB es la configuración de las redirecciones precisas para establecer las conexiones necesarias que permitirán el desarrollo de una experiencia concreta. Estas redirecciones constituyen la funcionalidad principal de CONEXLAB, y deben incluirse en el proceso de inicio de sesión SSH, mediante el gestor de sesiones.

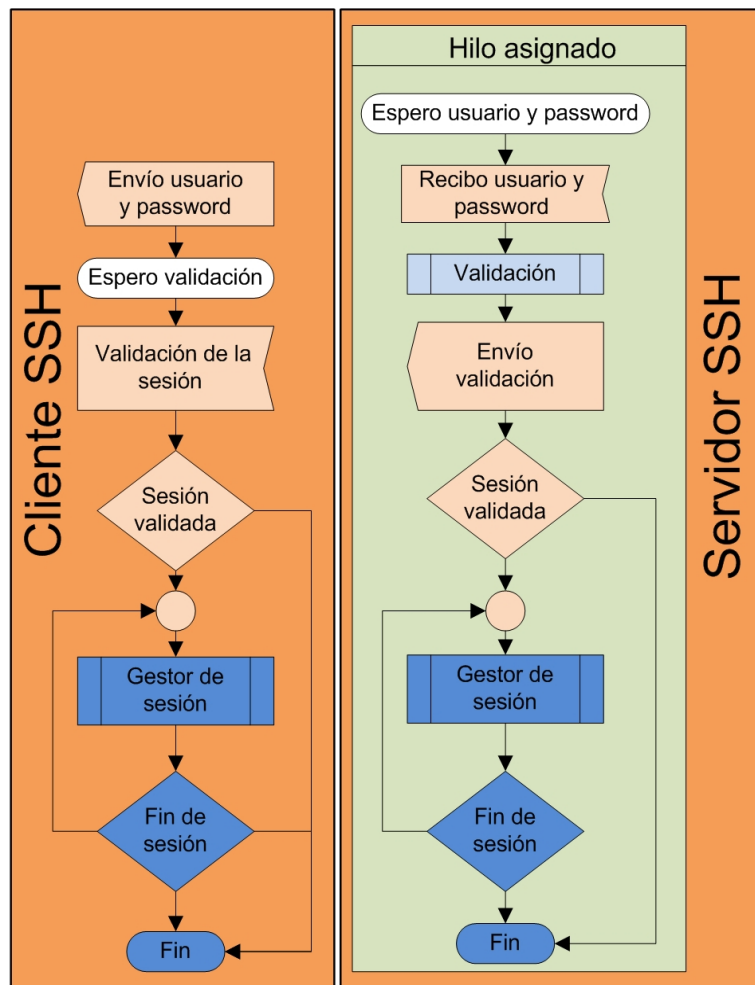


Figura 6.16: Diagrama SDL simplificado del proceso de establecimiento de sesión SSH.

La figura 6.17 presenta la nueva estructura del gestor de sesiones, en la que se han introducido las funcionalidades de DIGEXLAB que gestionan la configuración de la experiencia y que se estudiarán en profundidad en el siguiente apartado. Además se han introducido en esta estructura todas las funcionalidades de CONEXLAB presentadas a lo largo del capítulo 4. La inclusión de las funcionalidades de CONEXLAB y DIGEXLAB convierte el servidor SSH en un servidor SARLAB. En la figura 6.17 se muestra que el cliente, una vez obtenida la configuración de la experiencia, procede a establecer las redirecciones necesarias para la realización de la misma. A partir de ahí, se limita a gestionar los eventos que se producen desde la red o desde el usuario hasta que se dan las condiciones de fin de sesión. Por su parte, el servidor inicia la alimentación eléctrica de la experiencia a través del subsistema de

6. Implementación Funcional del Modelo basado en la Tecnología Disponible

alimentación, y procede a verificar el fin del tiempo de vida de la experiencia. Gestiona además los eventos que se producen desde la red hasta que se dan las condiciones de fin de sesión.

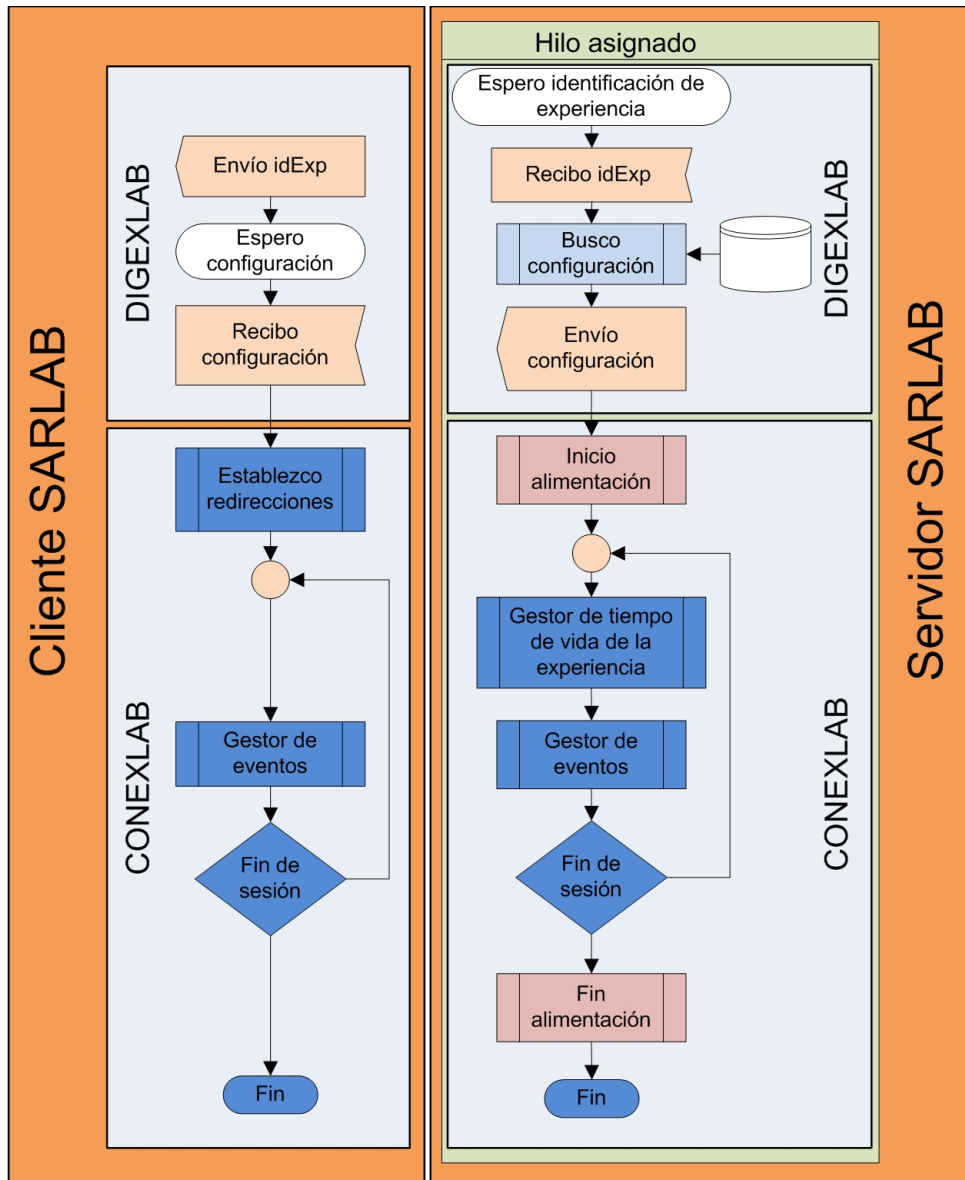


Figura 6.17: Diagrama SDL simplificado del gestor de sesiones de SARLAB.

Para el desarrollo de CONEXLAB hay que crear por tanto una clase Java que permita modelar la estructura de la matriz de conexiones de las experiencias del modelo tecnológico general, y que le será proporcionada a CONEXLAB por DIGEXLAB.

Además, hay que crear un método en el cliente que reciba como parámetro una instancia de esta clase y genere sus correspondientes redirecciones a los túneles que definen la experiencia. Este método será invocado desde el proceso de gestión de sesiones, en el procedimiento de establecimiento de redirecciones que puede observarse en el diagrama de la figura 6.17.

6.7. Desarrollo de DIGEXLAB

En la definición de DIGEXLAB del modelo tecnológico general se pueden diferenciar claramente dos funcionalidades: la de diseño y la de gestión. La funcionalidad de diseño comprende las tareas de creación y mantenimiento de las experiencias que, como se comentó en el capítulo anterior, se traduce en la gestión de la base de datos de definición de experiencias. La otra funcionalidad comprende las tareas de gestión del inicio de la experiencia, entre las que se encuentra la configuración adecuada de CONEXLAB y la validación del acceso.

6.7.1. Funcionalidad de gestión de DIGEXLAB

La inicialización de CONEXLAB consiste en configurar las redirecciones necesarias para establecer las conexiones entre el PC del alumno y el sistema experimental para la realización de una experiencia. Previamente debe haberse realizado la validación de acceso. Ambas son tareas de gestión de DIGEXLAB, que se estudian a continuación, así como las modificaciones necesarias sobre el proceso de inicio de sesiones SSH de la figura 6.17, para incluir las funcionalidades de gestión del inicio de sesiones de DIGEXLAB, convirtiendo así el servidor SSH en el servidor SARLAB.

6.7.1.1. Configuración inicial de CONEXLAB

En la figura 6.17 se presenta el diagrama SDL simplificado del gestor de sesiones. En una primera etapa el gestor realiza el proceso de inicialización. Este proceso parte del cliente, que manda el identificador de la experiencia al servidor y pasa al estado de espera de configuración. El servidor, que estaba a la espera del identificador, cuando lo recibe, busca en la base de datos de definición de experiencias la configuración que se corresponde a este identificador y se la envía al cliente. Cuando el cliente recibe la configuración de la experiencia, inicia CONEXLAB ejecutando el procedimiento de definición de las redirecciones correspondientes a la configuración recibida, figura 6.17. Finalmente hay que indicar que en ninguno de los diagramas SDL se han incluido los controles de tiempo excedido para facilitar la interpretación de los algoritmos.

Por otro lado, la base de datos de definición de experiencias se ha estructurado en un conjunto de ficheros: un fichero directorio y un fichero de definición de conexiones para cada una de las experiencias remotas gestionadas. Todos los ficheros del conjunto tienen formato XML. El fichero directorio contiene un registro por cada experiencia, con el nombre de su fichero de definición de conexiones y su identificador de experiencia. Además, en cada registro se encuentra/n el/los usuario/s propietario/s de la experiencia. En la figura 6.18 se muestra una parte del contenido de este fichero.

```
<Experience>
  <idExperience>Demo</idExperience>
  <fileName>EXP0001.xml</fileName>
  <owneUser>manolo</owneUser>
  <owneUser>marta</owneUser>
  <owneUser>luis</owneUser>
  <owneUser>pepe</owneUser>
</Experience>
<Experience>
  <idExperience>Demo AR1</idExperience>
  <fileName>EXP0002.xml</fileName>
  <owneUser>luis</owneUser>
</Experience>
```

Figura 6.18: Detalle del fichero de experiencias.

En el fichero XML de cada experiencia se almacenan, además de la definición de conexiones, datos generales de gestión de la experiencia que no son utilizados por SARLAB, tales como el autor, el título, el/los cursos de utilización de la experiencia, el tipo de acceso colaborativo, los derechos de uso de la experiencia, etc. En la figura 6.19 se muestra un ejemplo de fichero de configuración XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<experiment>
  <general>
    <author>Andres Márquez</author>
    <title>Diseño Digital con VHDL (I)</title>
    <course>Diseño de Sistemas Empotrados</course>
    <auth>Moodle</auth>
    <status>1</status>
    <multiplexar>1</multiplexar>
    <administratorRights>marcoa</administratorRights>
    <administratorRights>mjias</administratorRights>
    <designRights>pepe</designRights>
    <designRights>juan</designRights>
    <useRights>user1</useRights>
    <useRights>user2</useRights>
  </general>
  <forwardingL>
    <IORemote>
      <nodosIORemote description="Camara AXIS 2">
        <ip>192.168.1.41</ip>
        <port>80</port>
      </nodosIORemote>
      <nodosIORemote description="Phidgets InterfaceKit 8/8/8">
        <ip>192.168.1.103</ip>
        <port>5001</port>
      </nodosIORemote>
    </IORemote>
    <IOLocal>
      <nodosIOLocal>
        <ip>127.0.0.1</ip>
        <port>8083</port>
      </nodosIOLocal>
      <nodosIOLocal>
        <ip>127.0.0.1</ip>
        <port>5001</port>
      </nodosIOLocal>
    </IOLocal>
  </forwardingL>
</experiment>
```

Figura 6.19: Fichero XML de configuración de una experiencia.

Por tanto, la implementación de la base de datos de definición de experiencias se realiza mediante estos ficheros de tipo XML. Esto se debe a que SARLAB actúa en un ambiente de programación distribuida, y la información de la base de datos debe ser compartida con otras aplicaciones, como por ejemplo el módulo EJSapp. Este intercambio de información se facilita con el uso de este formato estándar.

6.7.1.2. Validación de acceso

El protocolo SSH valida el acceso utilizando un método de identificación que compara los valores de usuario y contraseña introducidos por el usuario con los valores correspondientes almacenados en el servidor. Si existe correspondencia, se autoriza el inicio de la sesión. Este protocolo se corresponde sólo con una de las tres opciones de validación establecidas para el modelo tecnológico general, figura 4.19. Es necesario por tanto, incluir las otras dos en el protocolo, con lo que el servidor correspondiente pasa de ser servidor SSH a servidor SARLAB. Para englobar las tres opciones, se redefine la estrategia de validación SSH creando una clase que se encargará de gestionar la validación de usuario y contraseña desde un fichero local en formato Unix, en el que cada campo está separado por el delimitador ":". En ese fichero se almacenan el usuario y la contraseña, el nivel de usuario y los datos de identificación personal. La clase encargada de gestionar este fichero aporta métodos para gestionar altas, bajas, modificaciones, búsquedas, validaciones, etc. Con respecto a las otras dos opciones de validación (de acceso colaborativo y desde el LMS) se deben añadir dos nuevas clases que se encarguen de aportar sendos métodos de autorización de acceso en el servidor SSH.

La clase de gestión de acceso colaborativo se encarga de asociar a cada sesión un juego de claves (usuario, contraseña) de tal forma que el intento de acceder con este par usuario-contraseña, genere de forma automática la validación del usuario en la experiencia asociada con la correspondiente sesión colaborativa. El fin de la experiencia

colaborativa debe estar asociado al fin de la experiencia de inicio de la sesión. Para ello, esta clase tiene entre sus atributos una lista de todas las sesiones abiertas de las que se deben conocer, además, sus correspondientes usuarios y contraseñas y la hora de final de la experiencia. Como la clase anterior, ésta dispone también de métodos para gestionar altas, bajas, etc.

La tercera opción es la validación del acceso desde un LMS. En este caso el acceso a una experiencia se inicia desde una sesión abierta en el LMS. Por tanto, este proceso se inicia en el navegador del alumno y deben participar en él los dos servidores implicados: el servidor LMS y el servidor SARLAB. En esta opción, SARLAB y el LMS deben colaborar para validar el usuario. Así, una vez validado el usuario por el LMS, se crea un identificador que contiene el nombre de usuario y el servidor LMS que lo ha validado separados por el signo @ (<usuario>@<servidor>). La figura 6.20 presenta una visión global de este proceso en la que se muestra, de forma simplificada, el diagrama de interacción entre todos los participantes en el mismo.

En efecto, en la figura 6.20 se observa que, cuando un usuario que tiene iniciada una sesión en el LMS selecciona sobre el navegador el acceso a una experiencia, el servidor LMS recibe la solicitud y procede a la verificación de las condiciones de acceso (existencia de reserva, servidor autorizado, etc.). Si no se dan las condiciones de acceso, el servidor LMS informa al usuario con una notificación web en la que figuran las razones del rechazo. En el caso de que el acceso sea autorizado, el servidor LMS le manda al cliente (navegador del usuario) una página web que incluye, entre sus componentes, un applet generado con EJS que contiene la interfaz de usuario y los clientes SARLAB (CONEXLAB y DIGEXLAB). El applet contiene además como parámetros la dirección IP y el puerto del servidor SARLAB, el identificador de la experiencia, usuario, servidor LMS, algunos parámetros para definir el nivel de acceso al servidor SARLAB y una llave de sesión (identificador aleatorio único y temporal asociado al usuario y a su sesión). Una vez cargada esta página en el navegador, se

6. Implementación Funcional del Modelo basado en la Tecnología Disponible

inicia la ejecución del applet que pone en marcha el proceso de validación del cliente DIGEXLAB. Este cliente conecta con el servidor DIGEXLAB utilizando la dirección IP y el puerto recibidos como parámetros, e inicia el proceso de validación en el servidor DIGEXLAB. Este último solicita al servidor LMS autorización del intento de acceso. Para ello llama a una página dinámica que recibe como parámetro la llave de sesión. Si la llave no existe en la base de datos de sesiones, el servidor LMS contesta construyendo la denegación del acceso en la página dinámica. Si la llave existe, la página dinámica se construye validando el acceso. Cuando el servidor DIGEXLAB recibe la validación del acceso, informa al cliente del éxito del proceso. Si el proceso fracasa, el servidor corta la comunicación con el cliente.

Una vez concluido el proceso de validación de acceso desde el LMS, en la figura 6.20 se muestra el resto del proceso de acceso a una experiencia. En efecto, una vez que el cliente recibe la confirmación de acceso, se inicia la fase de configuración de CONEXLAB. Para ello, el cliente DIGEXLAB manda el identificador de experiencia al servidor DIGEXLAB, que lo utilizará para buscar la configuración de la matriz de conexiones de la experiencia y se la devuelve al cliente. El cliente DIGEXLAB termina la fase de inicio de la experiencia llamando al cliente CONEXLAB y transfiriéndole la matriz de conexiones. El cliente CONEXLAB crea los sockets de entrada y le solicita al servidor CONEXLAB las redirecciones indicadas en la matriz de conexiones. El servidor crea las conexiones y avisa al cliente para que inicie la interfaz de usuario.

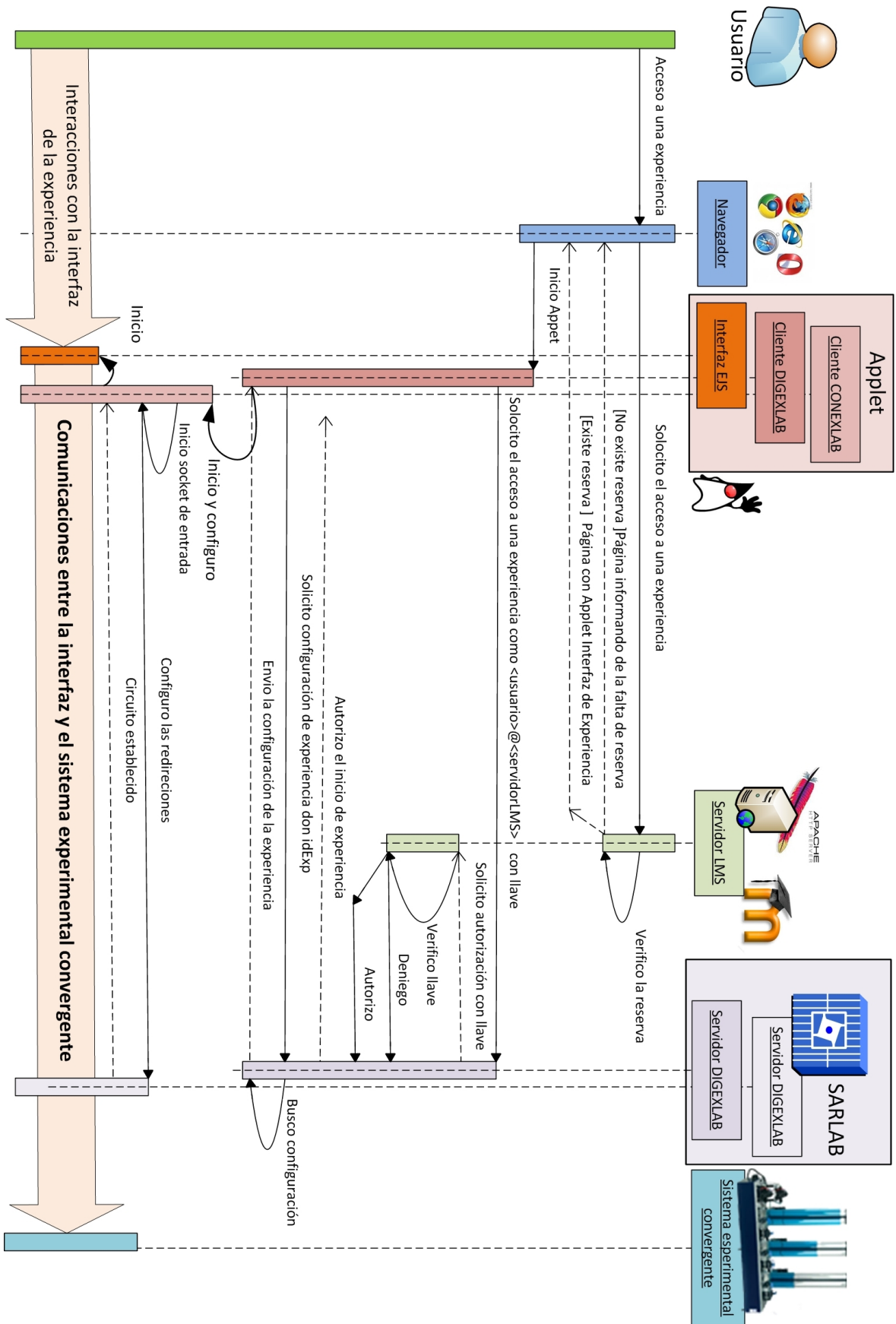


Figura 6.20: Diagrama de interacción de la validación y el acceso a una experiencia.

Finalmente, hay que indicar que, para garantizar la seguridad en la validación de acceso, DIGEXLAB dispone de una lista de servidores LMS autorizados (LMSList). Para hacer más flexible la configuración de los parámetros de validación de cualquier LMS, el LMSList incluye un parámetro que autoriza o no a validar procesos de diseño de experiencias desde el servidor LMS, otro parámetro que especifica el URL de llamada a la página dinámica, y otro parámetro que especifica el tipo de LMS (Moodle, Sakai etc.). En la figura 6.21 se muestra un ejemplo de la estructura del fichero LMSList.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<tabla>
  <LMSList>
    <entry>
      <key>http://sarlab2.uhu.es/sarlab</key>
      <value>
        <url>http://sarlab2.uhu.es/sarlab/mod/ejsapp/sarlab.php</url>
        <managementAccess>>true</managementAccess>
        <methodValidation>Moodle</methodValidation>
      </value>
    </entry>
    <entry>
      <key>TEST</key>
      <value>
        <url>http://sarlab2.uhu.es/LabExpRem/mod/ejsapp/test.php</url>
        <managementAccess>>true</managementAccess>
        <methodValidation>Moodle</methodValidation>
      </value>
    </entry>
  </LMSList>
</tabla>
```

Figura 6.21: Estructura del fichero LMSList.

6.7.2. Funcionalidad de diseño de DIGEXLAB

La función de diseño de experiencias de DIGEXLAB se limita a gestionar la base de datos de definición de experiencias, lo que se traduce en realizar procesos de altas, bajas, modificaciones, búsquedas, listado, etc., sobre las estructuras de datos almacenadas. Como se comentó en el apartado anterior, la base de datos de definición de experiencias está constituida por un conjunto de ficheros en formato XML, y en

consecuencia las tareas de diseño se traducen en la realización de estos procesos sobre este conjunto de ficheros. Esto es, crear una experiencia se corresponde con crear el correspondiente fichero de configuración XML, y hacer la correspondiente entrada en el fichero directorio. Borrar una experiencia es borrar el fichero XML y su correspondiente entrada en el fichero directorio. Modificar supone reescribir los correspondientes valores del fichero de configuración y/o fichero directorio. Estos procesos se realizan a través de la red, por lo que hay que crear una estructura cliente-servidor que lo permita. El servidor es el que accede a los ficheros XML y asume las funcionalidades de gestión de la base de datos de definición de experiencias. Debe además soportar las peticiones de realización de estos procesos producidas desde los clientes a través de la red. La estructura del servidor es básicamente la de un intérprete de comandos para la gestión de ficheros. El servidor espera conexiones desde los clientes y asigna un hilo de ejecución para atender cada una de estas conexiones, quedando nuevamente en estado de espera. El hilo de ejecución que atiende al cliente espera como primer parámetro un comando que especifique el proceso solicitado por el cliente. Una vez definido el servicio demandado, se produce el intercambio de información necesaria para realizarlo.

El lenguaje Java dispone de librerías para gestionar con gran comodidad el almacenamiento y la carga de ficheros XML en sus correspondientes estructuras de datos. Para adaptarlas a las necesidades concretas del servidor, se ha creado una clase que realiza todas las funciones de mantenimiento, incluido el bloqueo de ficheros durante la edición. Para ello, al iniciarse, carga en memoria la estructura de directorios de la experiencia. Dispone también de métodos para cargar configuraciones en memoria desde los ficheros de configuración XML y viceversa, para realizar las actividades de mantenimiento en el fichero directorio, para borrar, crear o modificar ficheros de configuración XML, etc.

6. Implementación Funcional del Modelo basado en la Tecnología Disponible

Al igual que las comunicaciones correspondientes a la evolución de una experiencia, las de diseño se establecen a través de túneles, considerando el servidor como un componente convergente más situado en la red del laboratorio, por lo que su acceso solo se permite a través de SARLAB. Para distinguir estos accesos de los propios de la evolución de la experiencia, se ha reservado un identificador de la experiencia (idExp). La validación de acceso al diseño requiere disponer del perfil adecuado y una cuenta de validación local, o que el servidor LMS tenga nivel de diseñador en la lista de servidores Moodle autorizados (ListLMS).

El servidor de diseño completa la estructura de DIGEXLAB y culmina el desarrollo a nivel de servidor del modelo de la figura 6.14. Sin embargo, a nivel de desarrollo hay que abordar el del cliente de diseño. Esto hace necesaria la creación de un elemento EJS adicional que implemente el cliente DIGEXLAB para gestionar el diseño de experiencias. Este elemento hace posible el diseño de experiencias desde EJS y por ende desde el LMS a través del EJSApp.

La ejecución del elemento DIGEXLAB proporciona al diseñador de experiencias una interfaz gráfica para asistirle en todas las tareas de gestión de experiencias. Además, en esa herramienta se han integrado las tareas de mantenimiento del servidor de diseño, como la configuración de las conexiones de una experiencia, la gestión de los usuarios locales, la gestión de la lista servidores LMS autorizados o la gestión del subsistema de alimentación. En el panel principal de la interfaz del cliente DIGEXLAB se diferencian dos áreas: la de configuración de conexiones y la de configuración de la alimentación, figura 6.22. En el área de configuración de conexiones (*Configuración de las conexiones SARLAB* en la figura) se muestra una lista de los identificadores de experiencias definidas (idExp), y un conjunto de botones para la definición de una nueva experiencia o para editar o borrar cualquiera de ellas.

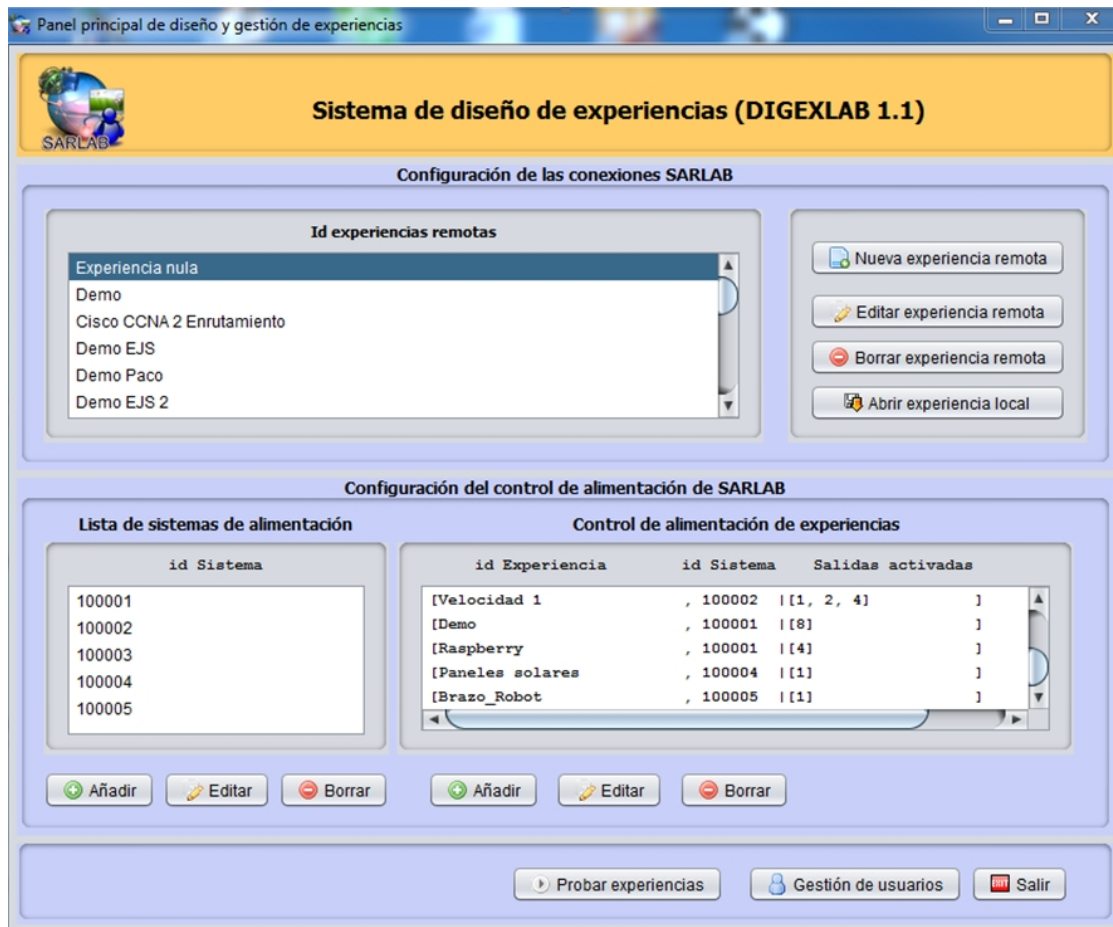


Figura 6.22: Panel principal de la interfaz del cliente DIGEXLAB.

Si se pulsa el botón de nueva experiencia o el de edición, se abre el panel de diseño de conexiones, figura 6.23, donde se definen los parámetros de configuración de las experiencias, tanto los parámetros generales (autor, título, curso, derechos, etc.) como aquellos que especifican las conexiones a través de los sockets de entrada y salida de las conexiones. Todos estos parámetros se almacenan en el correspondiente fichero de configuración XML, de forma local o remota.

Este panel ofrece la opción de añadir, borrar o editar las conexiones que aparecen en la lista. Al pulsar el botón de añadir o editar se abre el panel de conexiones, donde se especifican los puntos de entrada y salida de una conexión, figura 6.24.

6. Implementación Funcional del Modelo basado en la Tecnología Disponible

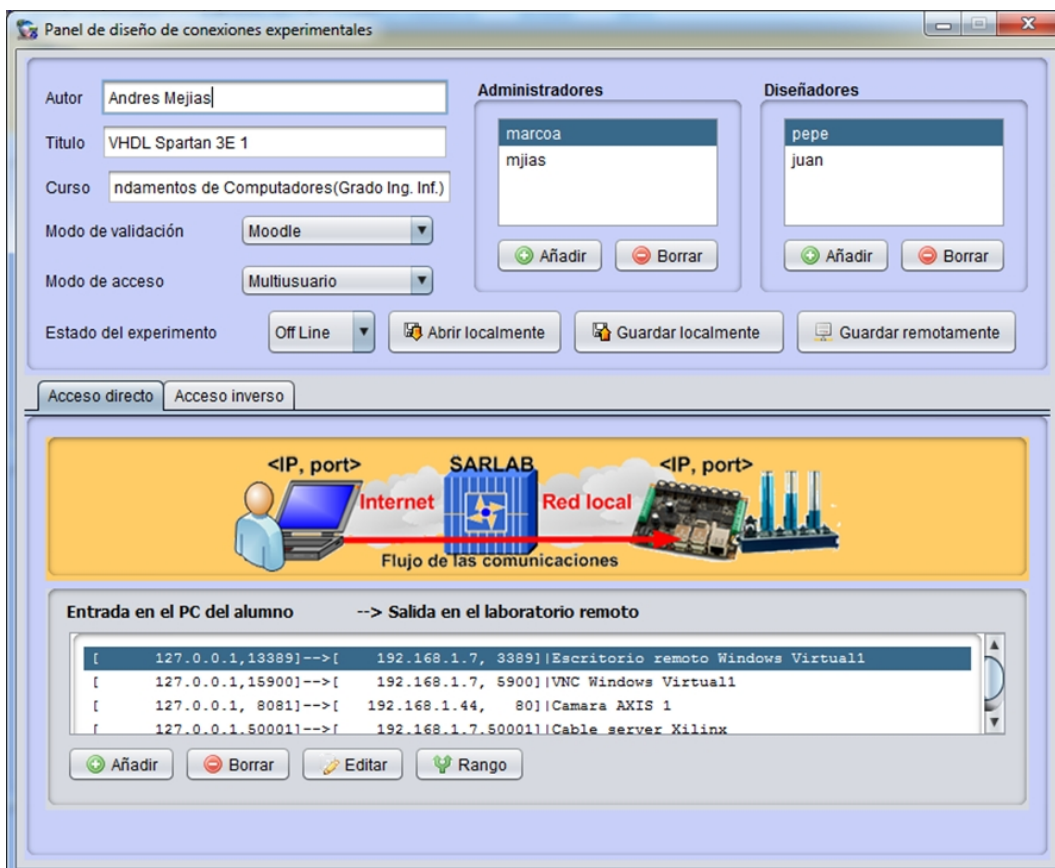


Figura 6.23: Panel de diseño de conexiones de DIGEXLAB.



Figura 6.24: Panel de conexiones de DIGEXLAB.

En el área de control del subsistema de alimentación (*Configuración del control de alimentación de SARLAB*, en la figura 6.22) se muestra una lista de los identificadores de los dispositivos de control de alimentación disponibles en la red del laboratorio, y las

experiencias asignadas de cada uno. En este área se pueden definir nuevos dispositivos de control de alimentación o editar los existentes. Si se pulsa el botón de añadir, se abre un cuadro diálogo con todos los sistemas experimentales convergentes disponibles en la red del laboratorio, para que el diseñador seleccione la experiencia a controlar desde el nuevo dispositivo. Ese cuadro diálogo abre otro en el que se muestran los dispositivos definidos y el diseñador hace las conexiones correspondientes, figura 6.25. Además, el diseñador puede definir las salidas que debe activar el nuevo dispositivo al entrar un usuario en la experiencia asociada, y desconectarlas al producirse la salida.

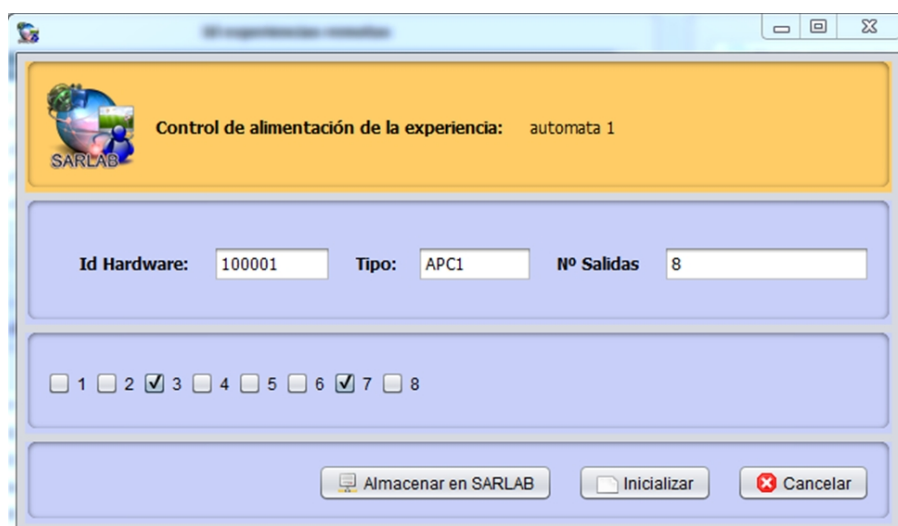


Figura 6.25: Panel de asignación de salidas de alimentación.

Con todo lo expuesto están asistidas las funciones de diseño de las experiencias. Además de estas funciones de administración y diseño de la experiencia, es necesario administrar en el cliente DIGEXLAB los perfiles de usuario que pueden acceder al cliente de diseño.

6.7.3. Funcionalidades del sistema de convergencia genérico

Establecida EJS como herramienta de desarrollo rápido para la creación de interfaces de usuario, y desarrollados todos los aspectos funcionales de SARLAB, se aborda en

este apartado la asistencia al diseñador de experiencias en la creación de los sistemas de convergencia genéricos.

En el capítulo anterior se propusieron dos soluciones escalables a la arquitectura física de los sistemas de convergencia genéricos. Una primera denominada solución monolítica, para sistemas sencillos basada en el uso de la plataforma Arduino, y una segunda denominada solución modular para sistemas más complejos, basada en el uso de tarjetas microcomputadoras i86 o ARM, junto a tarjetas de adquisición de datos conectadas a través del bus USB.

Según los requerimientos derivados del modelo, las aplicaciones a añadir en estas arquitecturas deben aportar las tres funcionalidades siguientes:

- ✓ Conectividad del sistema de convergencia genérico con la interfaz de usuario.
- ✓ Control de la evolución de la experiencia.
- ✓ Seguridad de la experiencia.

La primera se encarga de todas las cuestiones relacionadas con el intercambio de información entre la interfaz de usuario y el sistema de convergencia genérico a nivel de aplicación (a nivel de transporte, las comunicaciones están resueltas por SARLAB). La funcionalidad de control de la evolución de la experiencia se encarga del tratamiento de la información experimental en el lado seguro de la red, es decir, de los valores de los sensores y de los actuadores implicados. La funcionalidad de seguridad de la experiencia se encarga de la misma en los dispositivos físicos situados en el laboratorio. Así, esta funcionalidad impide que el usuario dañe ningún elemento de la planta experimental didáctica. Las funcionalidades de seguridad y control de evolución de la experiencia son propias de cada naturaleza experimental, por lo que no es posible generalizarlas ni ampliar su desarrollo funcional. Sin embargo, es necesario hacer notar que la complejidad de estas dos funcionalidades es la que marca el uso de una de las

dos arquitecturas propuestas para el sistema de convergencia genérico. En este sentido, la primera opción basada en Arduino solo se utilizará en los casos en que las necesidades de seguridad y control sean muy básicas o inexistentes, y por tanto Arduino se limitará a proporcionar conectividad entre la interfaz de usuario y el sistema experimental.

Con respecto de la funcionalidad de conectividad, dado que en el modelo se propone un escenario unificado a nivel de aplicación y para ello se elige el uso de Modbus, conseguir resolver ésta se limita a añadir un esclavo Modbus en cualquiera de las arquitecturas propuestas, bien sea en la plataforma Arduino o en la tarjeta microcomputadora i86 o ARM. La fácil solución de la funcionalidad de conectividad es por tanto consecuencia de la utilización de Modbus, que con el uso del esclavo indicado y el máster en la interfaz de usuario, permite el intercambio de información entre cualquiera de los elementos que intervienen en el desarrollo de la experiencia.

Para el diseño de una experiencia es necesario implementar las funcionalidades del sistema de convergencia genérico de seguridad y control de la evolución. Aunque ambas dependen de la naturaleza experimental y por tanto su implementación no es generalizable, lo que se propone es un modo general de asistencia al diseñador en su desarrollo a través de la herramienta de desarrollo rápido EJS.

6.7.3.1. Uso de EJS como herramienta de desarrollo rápido de las funcionalidades del sistema de convergencia genérico

Tanto la arquitectura basada en i86 como la basada en ARM, pueden soportar el uso del sistema operativo Linux y la programación en Java. Por tanto, en este caso se puede usar EJS como herramienta de desarrollo rápido. Además, el acceso a los dispositivos de adquisición y actuación desde EJS solo requiere de los correspondientes controladores (*drivers*), figura 6.26. Para la implementación de estos controladores en EJS se ha creado un elemento por cada dispositivo al que sea necesario acceder, figura 6.26. Así, el acceso

6. Implementación Funcional del Modelo basado en la Tecnología Disponible

a los dispositivos desde EJS se limita, para el diseñador, a arrastrar, soltar y configurar los elementos correspondientes a los dispositivos presentes en la experiencia. De la misma forma, para conseguir la conectividad desde red, se ha creado el elemento esclavo Modbus que también hay que arrastrar, soltar y configurar.

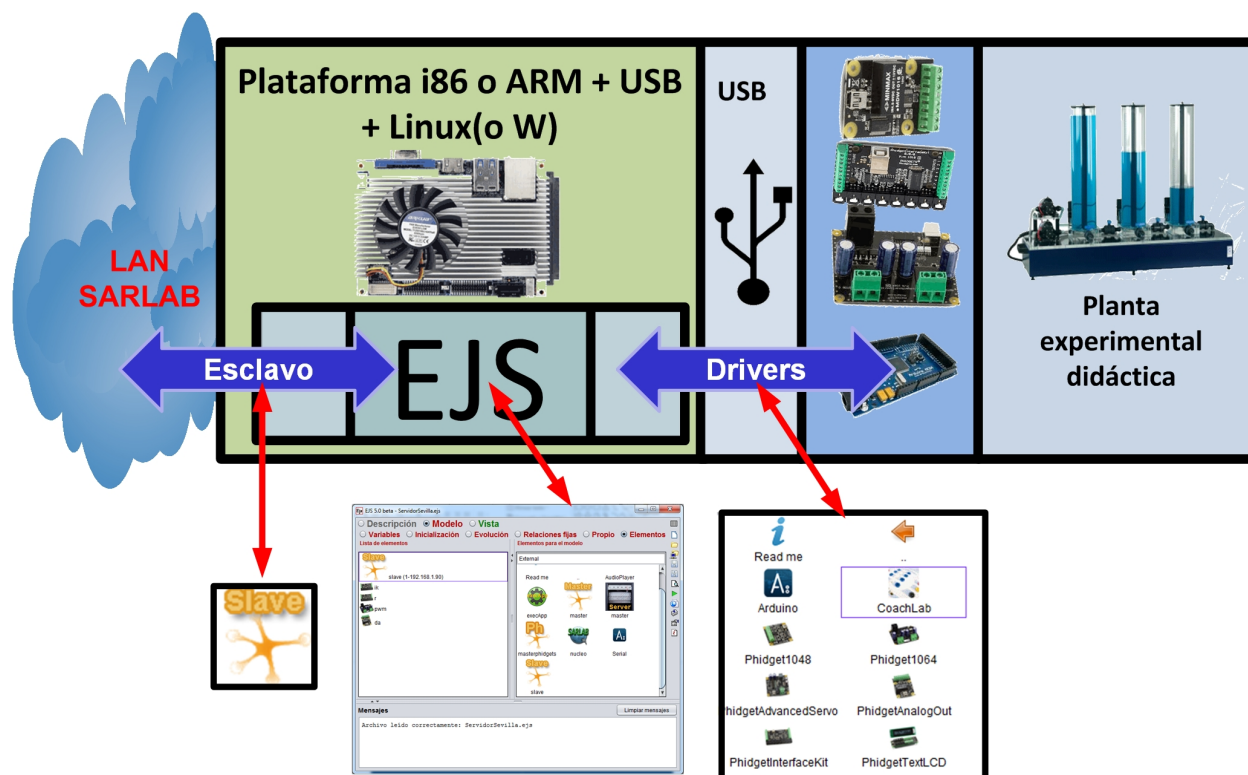


Figura 6.26: Esquema de uso de EJS como herramienta de desarrollo del sistema de convergencia genérico.

El desarrollo de estos elementos ha constituido una parte importante de esta Tesis, en la que el autor de la misma ha colaborado estrechamente con el Dr. Andrés Mejías Borrero.

Con estos elementos se ha logrado la conectividad del sistema experimental desde la red, y el acceso a la información de los sensores y actuadores que contiene. Además, la herramienta EJS asiste al diseñador en la creación de las funcionalidades de seguridad y control de la evolución de la experiencia.

6.7.4. Consideraciones sobre la eficiencia de Internet

En la evolución del modelo tecnológico general hacia la implementación física, que se ha detallado en estos dos últimos capítulos, se han hecho las consideraciones pertinentes sobre el rendimiento de las comunicaciones en el ámbito de las redes de campus, en las que se sitúan los sistemas experimentales. Estas redes presentan un diseño jerárquico en el que es factible el control de la calidad de servicios y en el que es posible disponer de un ancho de banda alto (mínimo 100 Mbps. por cable y 54 Mbps. mediante Wifi), con un número de paquetes perdidos muy bajo y una latencia que se puede acotar por debajo de las decenas de milisegundo. Sin embargo, no se ha considerado el rendimiento de Internet como red de comunicaciones. En efecto, en el ámbito de Internet no es posible acotar ni los paquetes perdidos, ni el ancho de banda de una comunicación, ni tan siquiera saber si la latencia va a ser constante o variable. En definitiva, en la actualidad no es posible medir ni la fiabilidad ni la velocidad de Internet, porque si bien estos datos de rendimiento son conocidos para cada uno de los operadores, son muy pocos los publicados y los que tienen carácter general a nivel de toda la red.

En este sentido, una vez evaluados algunos proyectos existentes que pretenden desarrollar indicadores de la eficiencia de las comunicaciones a través de Internet, [ITR 2015], o que miden de forma aproximada el rendimiento de Internet, centrándose en las conexiones entre un grupo de sistemas distribuidos en varios lugares del mundo, [Khamis 2003], en esta Tesis se ha optado por considerar únicamente que existe una relación directa entre el retraso, las pérdidas de paquetes y la velocidad de transmisión, pero que no existe un indicador numérico que cuantifique globalmente la relación entre ellos y que permita por tanto calcular el rendimiento de la red.

Además, SARLAB no puede mejorar las condiciones de la red, pero sí podría evaluarla e informar al usuario sobre las condiciones en las que trabaja en cada

momento. Sin embargo, hay, como mínimo, dos razones por las que se ha decidido no hacerlo:

- ✓ La evolución continua de la tecnología de Internet hace que los anchos de banda de los usuarios estén en continuo aumento, y la competencia entre los operadores de comunicaciones hace que los avances en ancho de banda sean enormes. Por ejemplo, en la actualidad se ofertan a usuarios anchos de banda de conexión a Internet que hace unos años eran propios únicamente de las redes locales. Esta tendencia hace suponer que esta evolución nos llevará a un escenario con anchos de banda cada vez más elevados, latencias cada vez más bajas y menor número de paquetes perdidos. De hecho, en España y en los países de su entorno, el rendimiento de la red es suficientemente bueno actualmente para el funcionamiento óptimo del modelo propuesto en esta Tesis.
- ✓ En los casos y países en los que el rendimiento de la red no garantice el funcionamiento óptimo del modelo, debe ser el diseñador de la experiencia o el usuario de la misma el que decida sobre la conveniencia de su diseño y/o realización.

En definitiva el escenario hacia el que evoluciona Internet tiende al aumento continuo de la eficiencia. No obstante, y dada la incertidumbre actual del rendimiento de la red, el diseñador de la experiencia debe ser consciente de ello y decidir, por ejemplo, donde colocar los algoritmos de control y seguridad necesarios en cada sistema experimental, si en el lado seguro de la red (laboratorio) o en el lado no confiable de la red (usuario). En este segundo caso, debe arbitrar los mecanismos necesarios para superar la latencia, la posible pérdida de paquetes o la posible limitación en el ancho de banda de la instalación. En este sentido, se ha creado un elemento de EJS que minimiza los efectos de la latencia de la red para la lectura de datos. Este elemento permite leer a través de la

red con la velocidad que requiera el sistema experimental y adecúa la transmisión de esos datos a las condiciones de latencia que presenta la red en cada momento. El elemento crea un buffer circular en el que se almacenan las adquisiciones de datos tomadas del sistema experimental. Cuando se produce una lectura desde la interfaz de usuario, el elemento envía todos los datos almacenados desde la lectura inmediatamente anterior.

6.8. Control topológico de un sistema experimental

El diseño y construcción de la planta experimental didáctica es la parte de la implementación del modelo que más esfuerzo de desarrollo y coste requiere. Por tanto, sería interesante poder usar una misma planta experimental con distintos fines didácticos, o el uso de los distintos componentes de cada una en varios sistemas experimentales. Este reto que a priori parece difícil, se consigue para determinados sistemas experimentales a través de control topológico desarrollado en el modelo de esta Tesis. Por ejemplo, considérese una planta experimental que representa una célula de fabricación flexible. Esta planta tiene motores, sensores, y elementos que permiten el control de su evolución. A este sistema experimental se le podrían dar varios enfoques didácticos. Por ejemplo, podría ser usado para el estudio de sensores, para el desarrollo del control automático del proceso o para el aprendizaje del lenguaje de programación de los dispositivos, entre otras opciones. La implementación de todos estos enfoques consiste simplemente en la decisión de las señales requeridas en la interfaz de usuario de cada uno y los elementos de procesamiento correspondientes, como por ejemplo PLC, FPGA, MCU, etc. Este uso versátil del sistema hace necesario el cableado de las conexiones de todos los elementos del sistema experimental y la accesibilidad desde red a toda la información relevante para todas las experiencias. La elección de uno de los enfoques consiste únicamente en habilitar las conexiones correspondientes y tomar los datos de las variables relevantes.

6. Implementación Funcional del Modelo basado en la Tecnología Disponible

El escenario de comunicaciones unificadas a través de Modbus propuesto en el modelo de esta Tesis facilita sobremanera el desarrollo del control topológico. En efecto, el uso de Modbus permite que cualquier elemento conectado a la red del laboratorio sea accesible desde la red interna con retardos del orden de milisegundos. Además, SARLAB hace accesible cualquiera de esos elementos desde Internet. Desde este lado, como se ha indicado anteriormente, no es posible acotar la latencia.

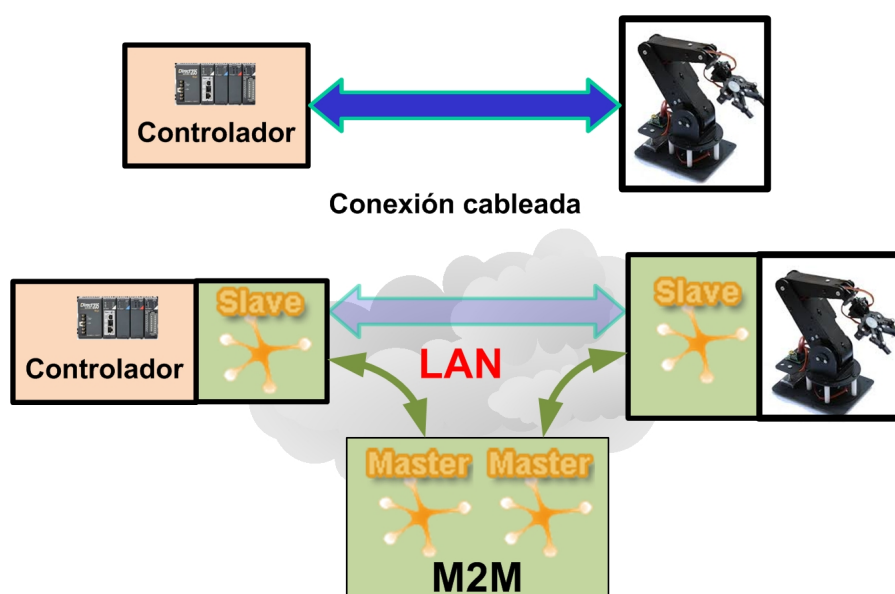


Figura 6.27: Esquema básico del control topológico.

El control del conexionado de todos los componentes presentes en un laboratorio se consigue mediante los elementos esclavo y máster de Modbus, figura 6.27. En la parte superior de la figura se presenta la conexión directa entre el controlador y el equipo controlado. En la parte inferior de la figura se realiza esta conexión mediante el control topológico propuesto. La conexión física se sustituye por una conexión virtual a través de una aplicación M2M que conecta el controlador con el equipo controlado mediante elementos maestro y esclavo de Modbus. La aplicación M2M es la encargada de realizar las conexiones necesarias para el intercambio de información en los dos sentidos. Para ello, dispone de dos másteres. En un sentido de la comunicación, uno de los másteres

lee en el esclavo del controlador la información sobre salidas de actuaciones a realizar en el sistema y el otro máster las traslada al esclavo del dispositivo controlado. En el otro sentido, uno de los másteres lee los sensores del equipo y el otro los traslada como entradas al esclavo del controlador.

Para usar distintos tipos de controladores, sería necesario hacerlos convergentes, accesibles desde red y conectarlos a través de la correspondiente aplicación M2M. Esto proporciona un mecanismo para habilitar o no las conexiones físicas entre los distintos componentes.

En resumen, el control topológico se consigue haciendo accesible toda la información de sensores y actuadores de los componentes de las distintas plantas experimentales a través de Modbus, figura 6.28, y conectando y haciendo convergentes desde la red las entradas y las salidas de los diferentes sistemas de control, como por ejemplo MCU, FPGA, PLC, PC+DAQ, etc. Dada la simplicidad que plantea la convergencia de estos dispositivos de control, ésta se puede realizar utilizando la arquitectura basada en Arduino de los sistemas de convergencia genéricos, añadiéndole un esclavo Modbus que mantendrá en su imagen el valor de las entradas y salidas de los controladores. La última etapa para conseguir el control topológico es la creación de una aplicación M2M desarrollada a partir de EJS, con los másteres Modbus necesarios para interconectar los dispositivos necesarios en cada caso.

En la figura 6.28 se presenta una planta experimental didáctica convergente con tres controladores distintos: un MCU, un PLC y una FPGA. El diseñador decide qué usar en cada caso, mediante la configuración del elemento M2M. La existencia del esclavo Modbus asociado a la interfaz de usuario permite que el alumno pueda controlar la evolución de la experiencia usando cualquiera de los controladores.

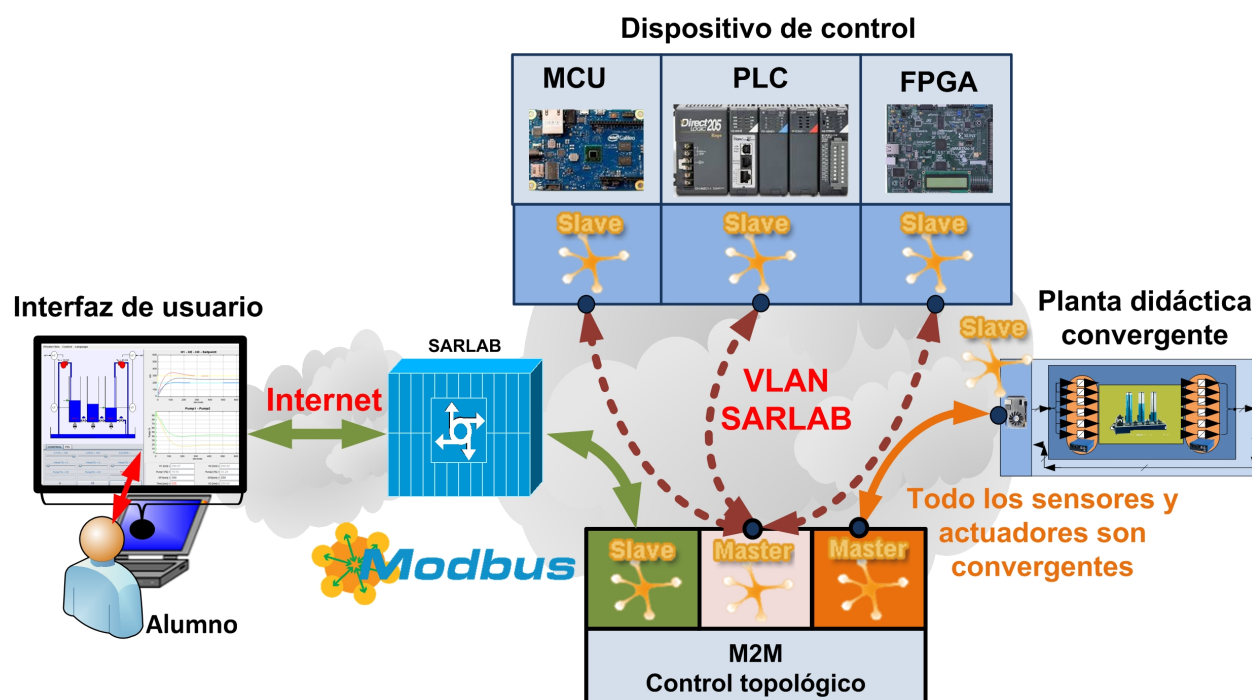


Figura 6.28: Representación del control topológico de un sistema experimental.

6.9. Conclusiones

Una vez desarrollado en el capítulo anterior el esquema general de implementación física del modelo tecnológico general, se han presentado en este capítulo las aplicaciones desarrolladas para dotar al modelo de todas las funcionalidades impuestas. Así, se describen dos estrategias de integración de SARLAB en el LMS, su integración como applet de Java y su integración completa en Moodle. Se cierra con ello el lazo de gestión experimental.

Asimismo, se desarrollan las aplicaciones encargadas de cerrar el lazo de desarrollo experimental a través del uso del protocolo de comunicaciones unificado Modbus y la elección de EJS como herramienta de desarrollo rápido. El uso de Modbus simplifica el desarrollo de las aplicaciones necesarias para cerrar el lazo de desarrollo experimental, y facilita asimismo la implementación del control topológico entre los distintos

componentes del laboratorio. Por otro lado, EJS proporciona las herramientas necesarias para el desarrollo de las interfaces de usuario conteniendo toda la información necesaria para la integración en Moodle, de sí misma y de las funcionalidades que permiten el acceso remoto a la experiencia. El diseño de nuevas experiencias se asiste a partir de EJS con el desarrollo de elementos que hacen transparente para el usuario las aplicaciones relacionadas con las comunicaciones de los distintos dispositivos.

Finalmente, se ha presentado en este capítulo el desarrollo de las funcionalidades de SARLAB, CONEXLAB y DIGEXLAB, y las diferentes funciones realizadas por cada una, así como las relaciones necesarias entre ellas a la hora de comenzar y terminar una experiencia.

Una vez presentados los aspectos físicos y funcionales de la implementación del modelo tecnológico general en la tecnología existente, se ha culminado la evolución del modelo conceptual general presentado en el capítulo 3 hacia un sistema real actualmente en explotación en varias universidades, y que ofrece prácticas remotas de distinta naturaleza experimental, lo cual se expone en el capítulo 7.

7. Laboratorios Remotos diseñados según el Modelo Propuesto

El modelo conceptual general propuesto en el capítulo 3 ha evolucionado en los capítulos 5 y 6 a una implementación física operativa que resuelve el problema de acceso remoto a experiencias para su realización. Asimismo, esta herramienta proporciona un sistema de asistencia al diseño de este tipo de experiencias desde su concepción hasta su puesta en explotación.

En este capítulo se presentan diferentes experiencias e incluso plataformas completas de acceso remoto en las que se ha usado el modelo propuesto en esta Tesis. En este sentido, el grado de aplicación del modelo no es el mismo en todas ellas, lo que pone de manifiesto su carácter abierto. Esto es, hay experiencias en las que se ha usado SARLAB como gestor de las comunicaciones en sistemas experimentales convergentes desarrollados a partir de herramientas propias o simplemente distintas de la que propone este modelo, y hay otras en las que el modelo propuesto se ha utilizado en todas las etapas del diseño y la implementación de la experiencia, como es el caso de las desarrolladas en la Universidad de Huelva. En este capítulo se muestra por tanto que el modelo es susceptible de usar con distintos grados de implementación.

Más que una enumeración de todas las experiencias que actualmente están en la red, y cuyo acceso remoto se ha conseguido mediante el uso de SARLAB, lo que se pretende es dar muestras concretas de estas experiencias, con las que además se constatan las

características fundamentales del modelo. En este sentido, lo primero que puede apreciarse es que todas las experiencias propuestas están integradas en un LMS, que garantiza el funcionamiento del lazo de gestión experimental. En segundo lugar, todas ellas utilizan SARLAB para la gestión de las comunicaciones y, por último, la naturaleza de las experiencias es muy diversa (esto es posible gracias al sistema de convergencia genérico), con lo que se atienden los tres aspectos fundamentales del acceso remoto a experiencias: el didáctico, el de comunicaciones y el experimental.

Se presentarán las experiencias más representativas desde el punto de vista del uso del modelo en sus distintos grados de implementación: sólo gestión de comunicaciones, gestión de comunicaciones e implementación del sistema de convergencia genérico o diseño de la experiencia desde su concepción hasta su explotación en acceso remoto de la forma que se describe en el apartado siguiente.

7.1. Procedimiento de diseño

El modelo implica un método sistemático de desarrollo de experiencias, que se puede resumir en los siguientes puntos:

- ✓ *Diseño didáctico.* Esta primera fase es común a todo tipo de experiencias de laboratorio, ya sean presenciales o con acceso remoto. En ella, el diseñador de la experiencia planifica los objetivos de la misma y a partir de ellos, su desarrollo para que el alumno adquiriera las capacidades y competencias asociadas con esos fines didácticos. En esta fase se construye la planta experimental didáctica alrededor de la cual girará la actividad del alumno.
- ✓ *Diseño del sistema de convergencia genérico.* Para conseguir el acceso desde red a la experiencia, es necesario hacerla convergente. Es decir, es necesario acceder a sensores, actuadores y controladores de la planta experimental didáctica desde

la red interna en la que está ubicada. La planta experimental y el sistema de convergencia genérico constituyen el sistema experimental convergente, que será accesible desde red, pero de forma local. El modelo propuesto en la Tesis proporciona un método sistemático y asistido para diseñar el sistema de convergencia genérico utilizando la herramienta de desarrollo rápido EJS y los elementos construidos con ese fin, aunque la convergencia en el modelo también podrá alcanzarse utilizando cualquier otro tipo de estrategia.

- ✓ *Decidir la información a proporcionar al alumno.* En esta fase el diseñador debe decidir la información necesaria para que el alumno pueda seguir la evolución de la experiencia y controlar la misma. Para ello dispone, entre otras herramientas, del flujo de vídeo y/o audio que se puede enriquecer con componentes de realidad aumentada.
- ✓ *Desarrollo de la interfaz de usuario.* En esta fase se utiliza EJS para el desarrollo de la interfaz de usuario, en la que se incluyen las herramientas usadas para proveer la información a proporcionar al alumno. La interfaz de usuario también incluirá los elementos que permitan seguir la evolución de la experiencia e interactuar con ella.
- ✓ *Diseño de comunicaciones y control de alimentación.* En esta fase se configuran las comunicaciones entre la interfaz de usuario y el sistema experimental convergente. Se configura asimismo el subsistema de alimentación de la experiencia. En ambos casos, la herramienta de diseño es DIGEXLAB, mediante la que se definirán los parámetros característicos de las conexiones necesarias para el desarrollo experimental (puntos de entrada y salida de los túneles), y se asociará a la experiencia el/los terminal/es de alimentación necesarios. En este punto, la experiencia es accesible desde cualquier punto de Internet, pero precisa de la validación local del acceso desde SARLAB. Asimismo, en este

punto está cerrado el lazo de desarrollo experimental y falta la implementación del de gestión experimental.

- ✓ *Integración de la experiencia en el LMS.* En esta etapa, se implementa y se cierra el lazo de gestión experimental. Además se habilita el acceso a experiencias con la validación en el LMS, que queda integrada con la de SARLAB. Esto se consigue mediante el módulo EJSApp, que integra el fichero .jar creado con EJS, en el LMS. Como quedó establecido en el modelo conceptual general, el LMS asume la gestión de todos los aspectos didácticos de la experiencia. Asimismo, el módulo EJSBooking, también integrado en el LMS, se encarga de controlar la reserva temporal del sistema experimental.

7.2. Experiencias en explotación usando el modelo

En este apartado se van a presentar una serie de experiencias que están actualmente en explotación con acceso remoto desde Internet, y que usan en su implementación el modelo presentado en esta Tesis. El apartado está organizado en tres bloques. En el primero se presenta UNILABs, [UNILabs 2015], que es una plataforma liderada por la UNED, en la que participan numerosas Universidades que comparten sus recursos experimentales a través de red. En esta plataforma, la herramienta que gestiona las comunicaciones es SARLAB, y se ofertan actualmente prácticas fundamentalmente de automática y robótica. En el segundo bloque se presentan prácticas ofertadas desde distintas Universidades y Centros Tecnológicos en las que también es SARLAB la herramienta que gestiona las comunicaciones. En este caso, los sistemas experimentales son de muy diversa índole. Por último, en el tercer bloque se presentan las experiencias desarrolladas y ofertadas desde la Universidad de Huelva, en la que la implementación del modelo es completa.

7.2.1. UNILabs

En UNILABS participan las siguientes Universidades: Universidad Nacional de Educación a Distancia, Universidad de Huelva, Universidad Complutense de Madrid, Universidad de Almería, Universidad Miguel Hernández, Universidad Loyola Andalucía, Universidad de León, Universidad de Alicante, Universidad de Murcia y Universidad Federal de Santa Catarina. La plataforma está liderada por la UNED, y el sistema que gestiona las comunicaciones en todas las experiencias es SARLAB.

Entre las prácticas ofertadas se encuentran las que se exponen a continuación:

7.2.1.1. Control de un cuatrirrotor de 3 grados de libertad

Esta práctica remota es ofertada por la Universidad Complutense de Madrid (UCM), y puede observarse en la figura 7.1. La planta es un sistema no lineal formado por una estructura rígida, que pivota sobre una articulación rotatoria de 3 grados de libertad (cabeceo, alabeo y guiñada), debido a la propulsión generada por 4 hélices acopladas a 4 motores. Los ángulos máximos de giro en cabeceo y alabeo están limitados a $\pm 40^\circ$. La potencia suministrada a cada motor permite controlar de forma independiente la velocidad de giro de cada hélice, mientras que la orientación de la estructura rígida se mide a través de 3 encoders ópticos acoplados a cada eje. Con este laboratorio pueden realizarse, entre otras, las siguientes tareas y actividades:

- ✓ Identificación del sistema.
- ✓ Diseño de controladores lineales: PID, realimentación de estado.
- ✓ Diseño de controladores no lineales: linealización por realimentación, por modos deslizantes.

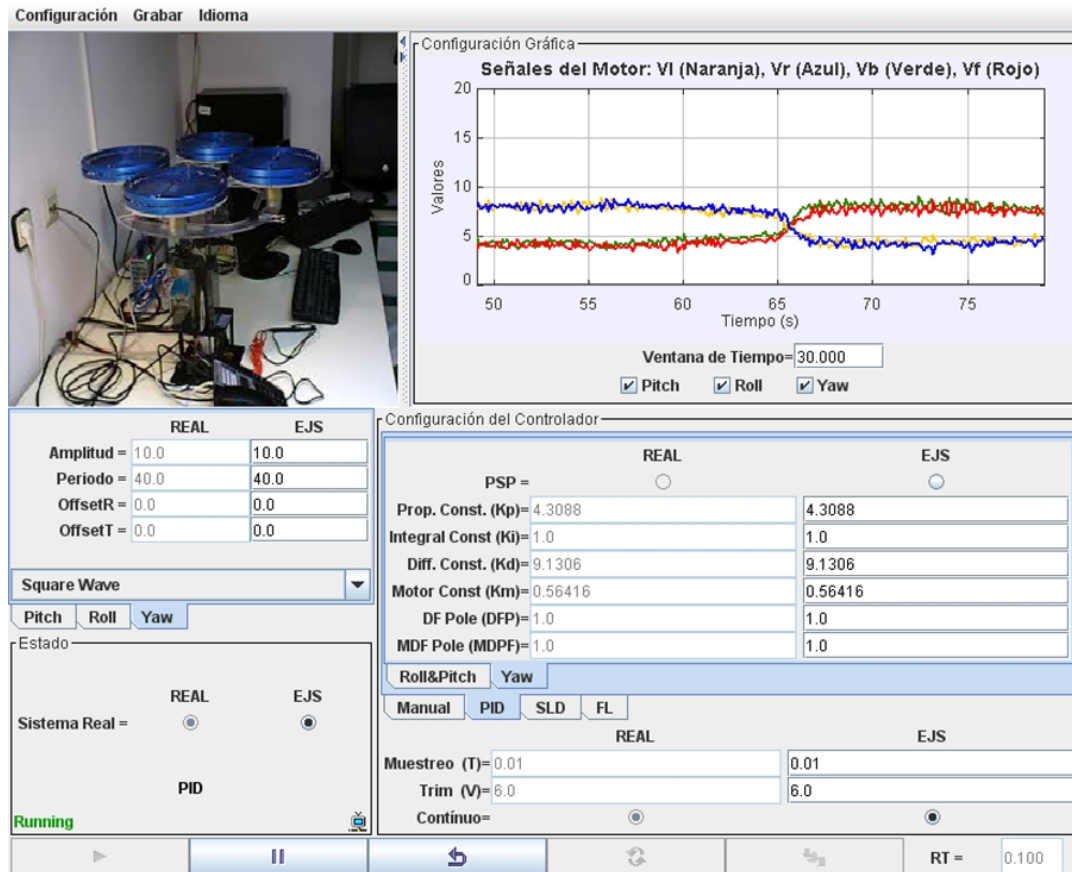


Figura 7.1: Control de un cuatrirrotor de 3 grados de libertad.

7.2.1.2. Control de robots móviles (sistema multi-agentes)

Este sistema experimental puede observarse en la figura 7.2. El laboratorio está basado en una plataforma multi-robots constituida por pequeños robots móviles con ruedas (Moway). Estos robots están formados por cinco módulos fundamentales: el sistema de procesamiento, el sistema motriz, el sistema de sensores e indicadores, el sistema de alimentación y el bus de expansión, mediante el cual se puede conectar un módulo de comunicaciones inalámbricas por radiofrecuencia (RF), que permite a los robots comunicarse entre ellos. Con este laboratorio pueden realizarse, entre otras, las siguientes tareas y actividades:

- ✓ Control de posición de un robot diferencial.

- ✓ Implementación de algoritmos de esquivas de obstáculos.
- ✓ Control de formación de un sistema multi-robots en un ambiente colaborativo.

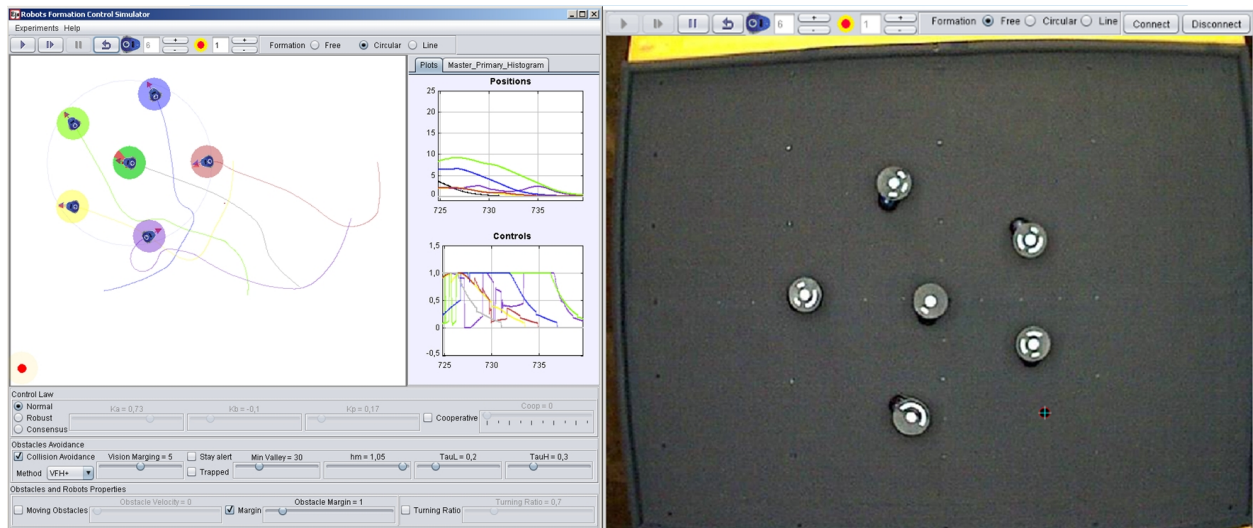


Figura 7.2: Control de robots móviles.

7.2.1.3. Control de posición y supresión de vibraciones de un brazo flexible

La figura 7.3 muestra este sistema experimental consistente en un brazo metálico flexible, que está libre por un extremo y unido por la base a un motor eléctrico. El problema de control principal es el posicionamiento del extremo del brazo flexible, al tiempo que se intentan suprimir o reducir las vibraciones que aparecen debido a la elasticidad del brazo. Los sensores incorporados en la planta permiten obtener medidas de la posición, velocidad del motor, y el ángulo de deformación del brazo flexible, medidos en la base de éste. La variable de control es la tensión de entrada al motor. Con este laboratorio pueden realizarse, entre otras, las siguientes tareas y actividades:

- ✓ Control de posición con un controlador PID.
- ✓ Control de posición por realimentación de estados y control LQR.

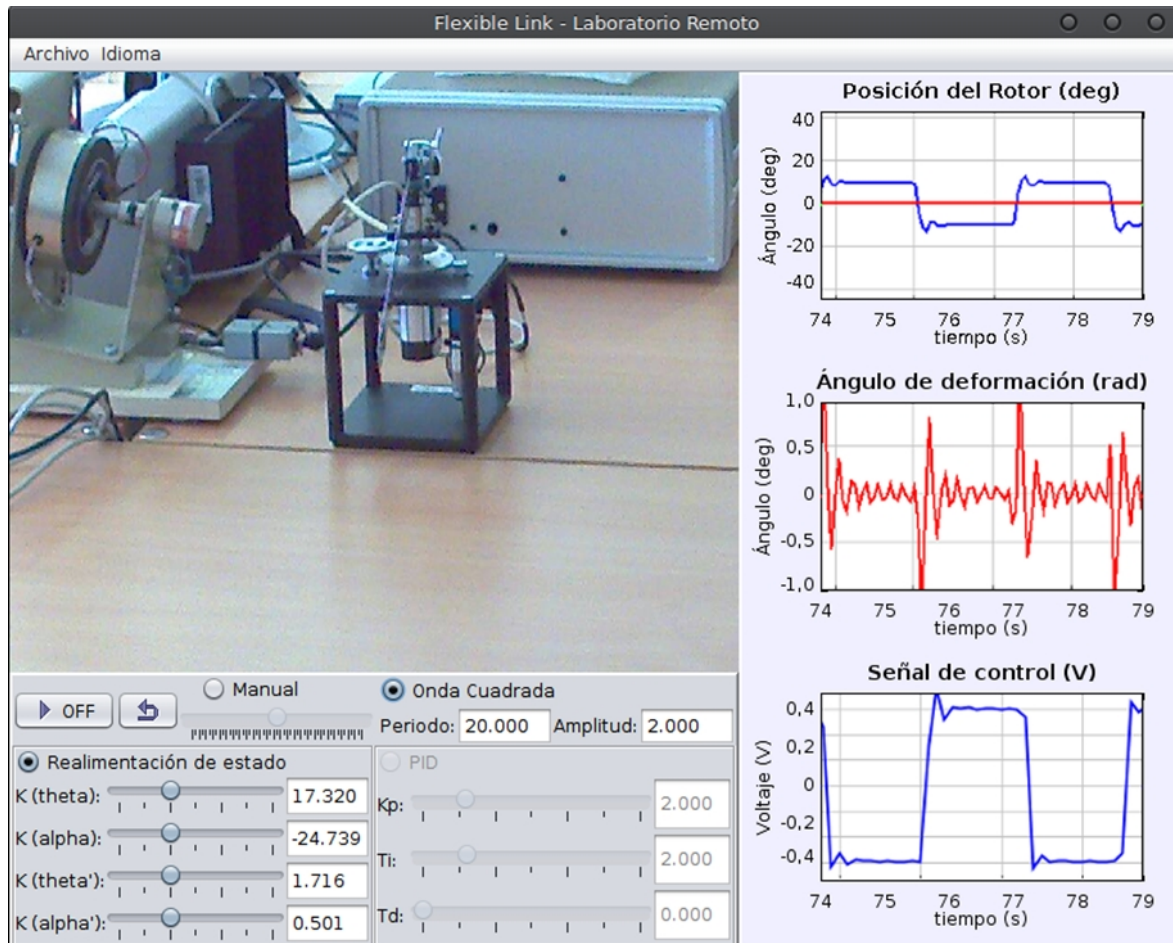


Figura 7.3: Control de posición y supresión de vibraciones de un brazo flexible.

7.2.2. Prácticas ofertadas por otras Universidades y Centros Tecnológicos

En este apartado se presenta una serie de prácticas ofertadas por Universidades distintas de la Universidad de Huelva en las que SARLAB es la herramienta que gestiona las comunicaciones. Es importante destacar en ellas la variedad de naturalezas experimentales que abarcan. Las tres primeras prácticas presentadas son ofertadas por la UNED. La última de ellas está ofertada por la Universidad de Cádiz, a través de su Centro de Innovación y Tecnología. En este caso, la naturaleza experimental son las comunicaciones.

7.2.2.1. Efecto fotoeléctrico

Se trata de un experimento (figura 7.4) que pone de manifiesto el carácter corpuscular de la radiación electromagnética y está dirigido a los estudiantes que se introducen por primera vez en el estudio de la Física Cuántica.

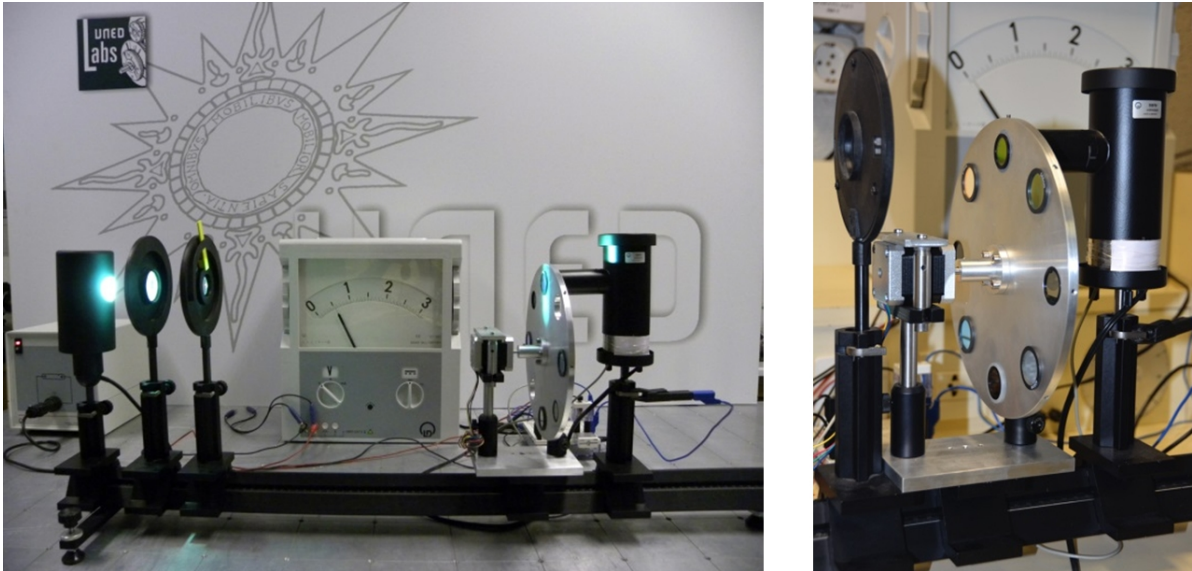


Figura 7.4: Fotografía general del montaje experimental real utilizado en el laboratorio remoto(izquierda). Detalle del disco giratorio motorizado para el intercambio de filtro interferencial a la entrada de la célula fotoeléctrica (derecha).

7.2.2.2. Difracción de la luz

La difracción de la luz es un fenómeno que pone de manifiesto el carácter ondulatorio de la luz y para su estudio suelen emplearse modelos físicos con distintos grados de complejidad matemática. El experimento aquí presentado (figura 7.5) emplea el modelo más sencillo, la denominada aproximación de Fraunhofer, y está dirigido a los estudiantes que se inician en el estudio de la óptica física.

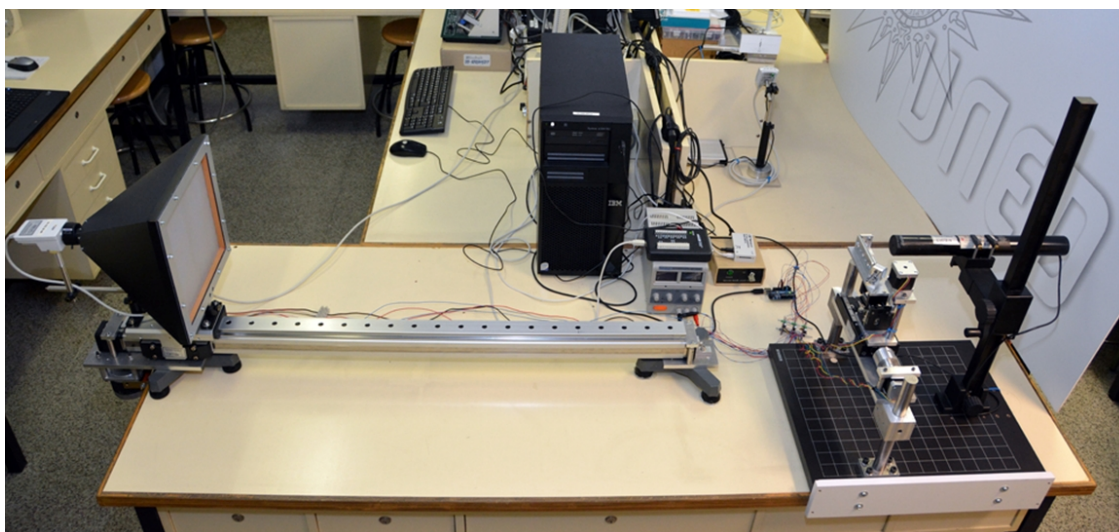


Figura 7.5: Sistema experimental para estudiar la difracción de la luz.

7.2.2.3. La luz en medios isótropos

Uno de los primeros fenómenos luminosos que abordó la ciencia es el de la reflexión y refracción de la luz en la superficie de separación de dos medios dieléctricos. Cuando un haz de luz pasa de un medio a otro, una parte del rayo se refleja hacia arriba y continúa en el primer medio mientras que otra se refracta, penetrando en el segundo medio. El sistema experimental se puede observar en la figura 7.6.

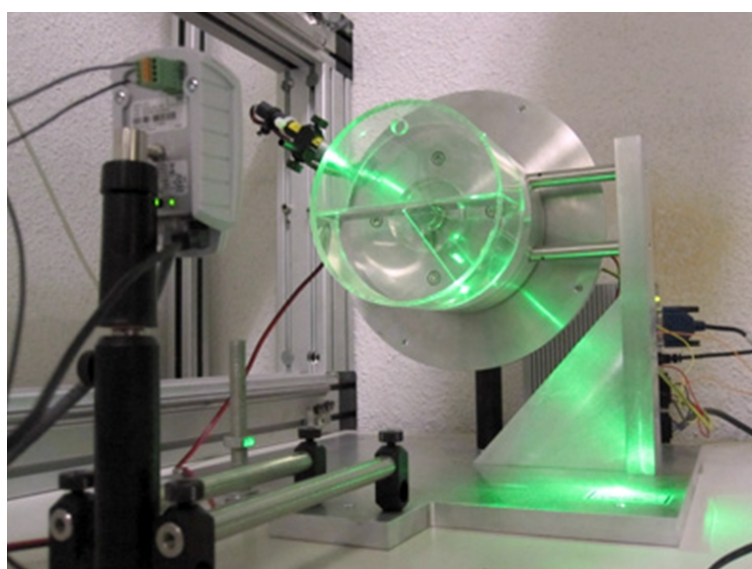


Figura 7.6: Sistema experimental que estudia la luz en medios isótropos.

7.2.2.4. Configuración del bucle de abonado para integración de voz y datos

En esta experiencia de laboratorio remoto el alumno realiza los procedimientos de puesta en servicio y configuración de un *Vigor Access IP DSLAM A24S* para prestar servicio a abonados de voz, datos y TV en un escenario de proveedor de comunicaciones (figura 7.7).

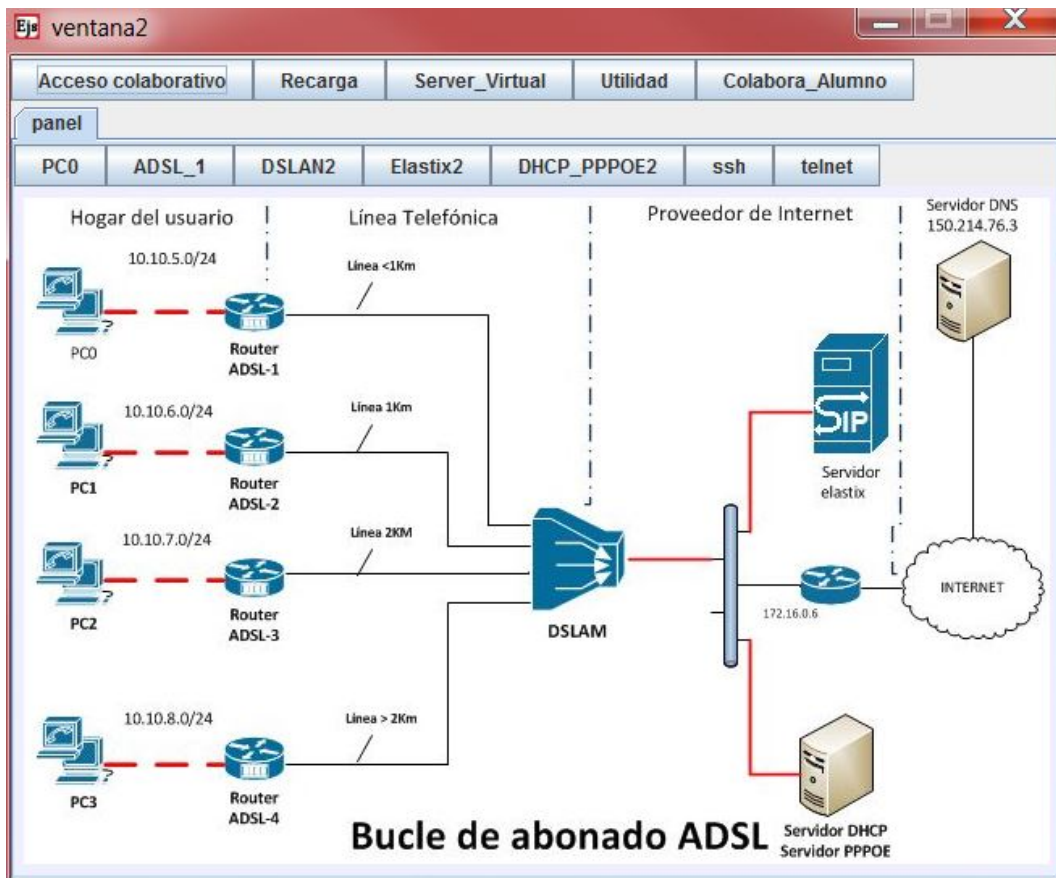


Figura 7.7: Sistema experimental de la práctica configuración del bucle de abonado para integración de voz y datos.

7.2.3. Práctica ofertadas por la Universidad de Huelva

Finalmente, en este apartado se presentan algunas de las prácticas ofertadas por la Universidad de Huelva desde el Laboratorio de Experimentación Remota. En este caso

la implementación del modelo ha sido completa, es decir, para su diseño y construcción se ha seguido el procedimiento indicado en el apartado 1 de este capítulo, pero además se cumple que:

- ✓ El sistema que gestiona las comunicaciones es, como en los apartados anteriores, SARLAB.
- ✓ La interfaz de usuario está integrada completamente en Moodle (esto también ocurre en las prácticas presentadas en los apartados anteriores). Para que esto se pueda hacer, la interfaz de usuario debe estar diseñada con EJS y el elemento SARLAB debe haber sido añadido.
- ✓ El sistema de convergencia genérico se ha diseñado según las opciones planteadas en el modelo de esta Tesis y de la forma asistida que el modelo proporciona, es decir, incluyendo los elementos de convergencia correspondientes a cada sistema experimental en EJS.

Asimismo, las prácticas presentadas en este apartado muestran el carácter general del modelo por la diversidad de áreas experimentales representadas.

7.2.3.1. Automatización con PLCs

Con esta práctica (figura 7.8) el alumno puede programar un autómata industrial *Schneider Electric* a distancia, para controlar adecuadamente una planta de laboratorio a escala que representa una línea embotelladora. El interfaz de usuario de esta práctica utiliza un sistema de realidad aumentada para enriquecer la visión del sistema experimental que observa el alumno. En este caso la convergencia se ha obtenido mediante una tarjeta microcomputadora de tipo i86 y una interfaz Phidget de adquisición de datos (empleando el elemento correspondiente de EJS).

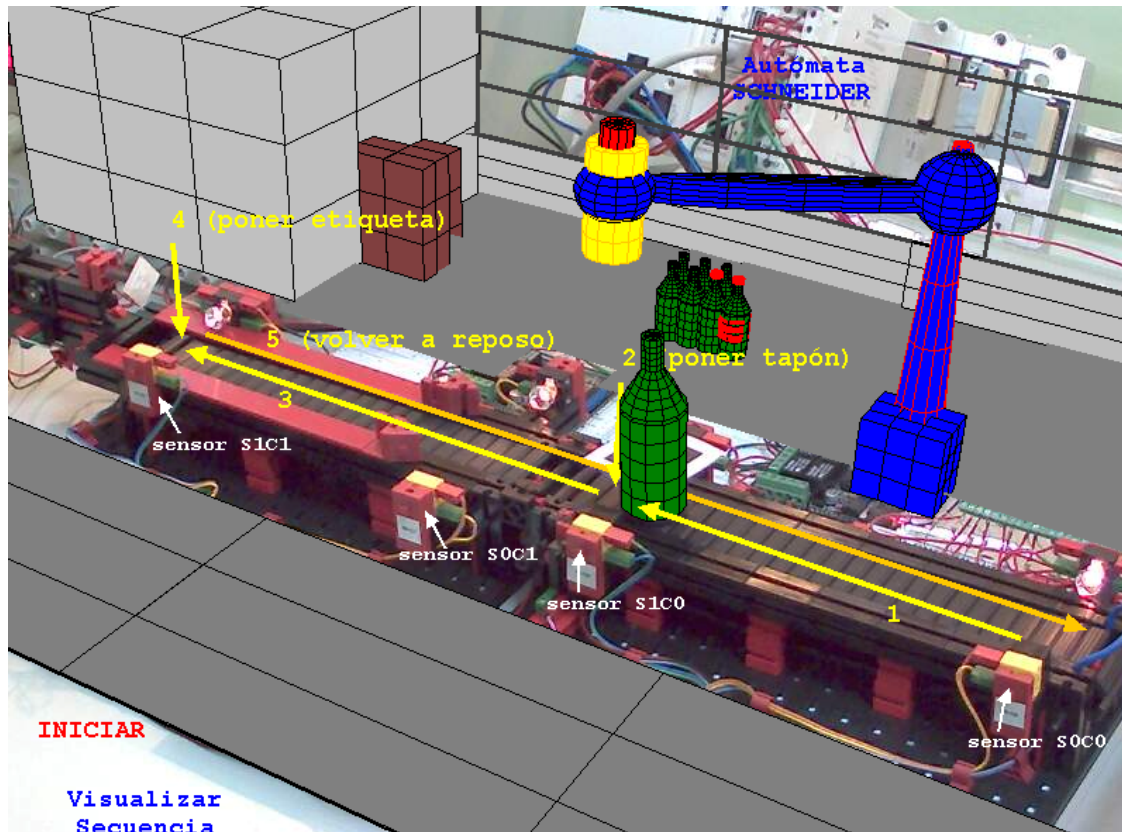


Figura 7.8: Sistema experimental de la práctica automatización con PLCs.

7.2.3.2. Diseño VHDL

Esta actividad permite realizar prácticas de diseño digital basado en un lenguaje de descripción de hardware (VHDL). Estas prácticas se realizan en tarjetas de desarrollo basadas en dispositivos FPGA de la empresa Xilinx. Los pulsadores e interruptores virtuales que pueden observarse en la figura 7.9 están realizados con técnicas de realidad aumentada y tienen la misma posición, tamaño y perspectiva que los reales (mayor percepción de interacción física con la tarjeta de desarrollo). En este caso la convergencia se ha obtenido a partir de una tarjeta Arduino Ethernet.

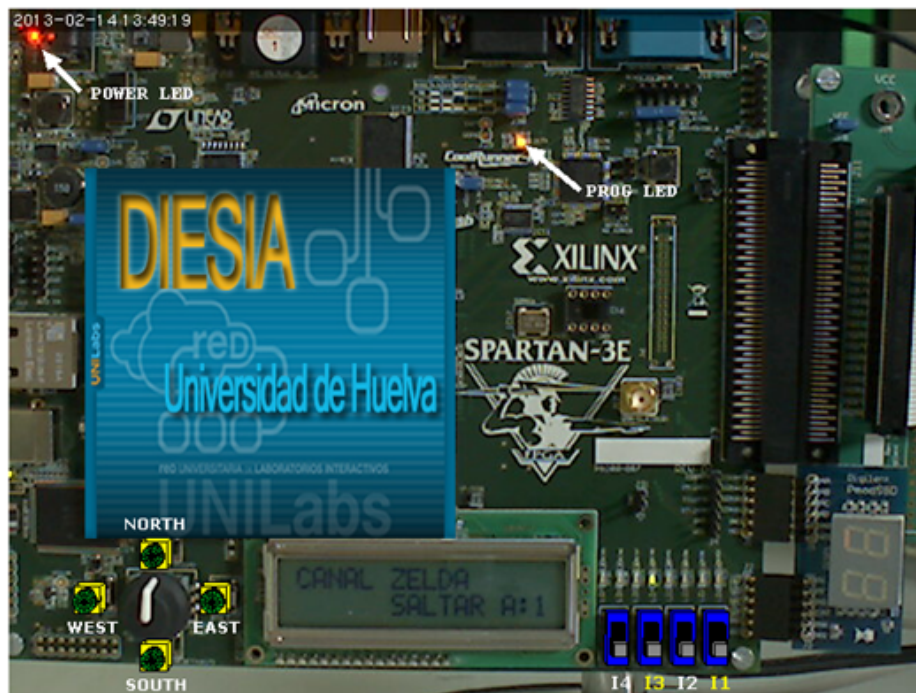


Figura 7.9: Sistema experimental de la práctica de diseño VHDL.

7.2.3.3. Curvas características de placas fotovoltaicas

Esta práctica (figura 7.10) permite obtener la curva característica de una placa fotovoltaica o de varias de ellas conectadas en serie y/o en paralelo. El sistema permite además variar las condiciones de radiación incidente. De esta forma, el alumno podrá analizar los efectos de estos parámetros en la curva característica de placas fotovoltaicas. En este caso la convergencia se ha obtenido con tarjetas Arduino Mega y el control de alimentación se realiza a través de una tarjeta microcomputadora Raspberry Pi.

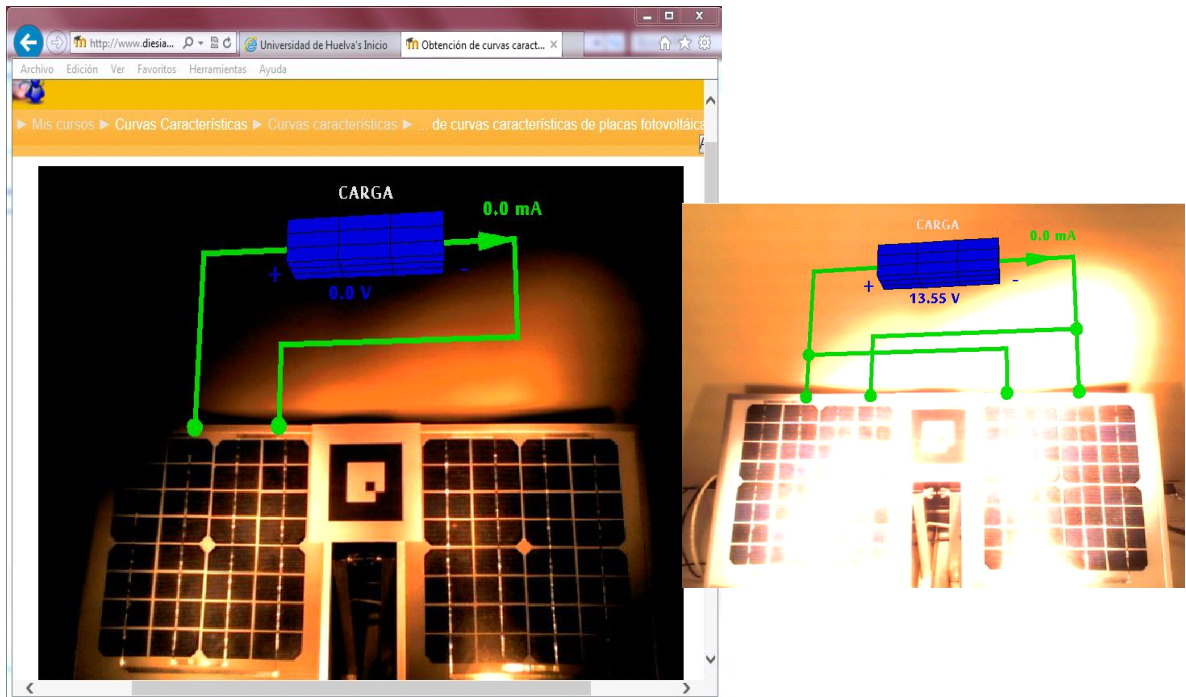


Figura 7.10: Sistema experimental de la práctica de placas fotovoltaicas. El alumno percibe los cambios de irradiancia que aplica de forma remota a las placas.

7.2.3.4. Modelado de una máquina síncrona

Con esta práctica (figura 7.11) se pretende determinar el valor de los parámetros del circuito equivalente de una máquina síncrona según el modelo lineal o de Behn-Eschenburg y según el procedimiento de Potier. A continuación se compararán los resultados obtenidos por cada uno de ellos con los reales que proporcionará la máquina al conectarse a ella una carga determinada. En este caso la convergencia se ha obtenido con una tarjeta basada en un procesador ARM.

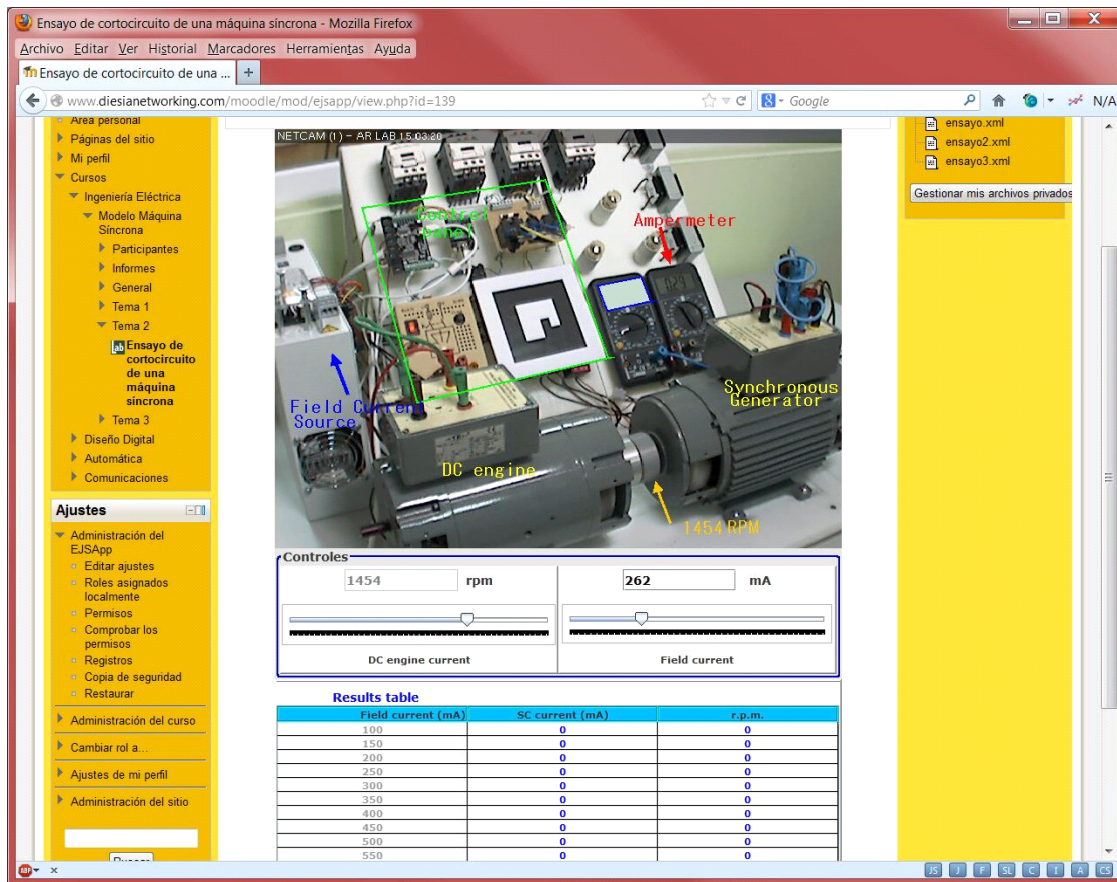


Figura 7.11: Sistema experimental de la práctica de modelado de una máquina síncrona.

7.3. Conclusiones

Aunque estas prácticas son una representación de las muchas existentes, queda patente la gran variedad de naturalezas experimentales presentadas, lo que pone de manifiesto el carácter general del modelo y su aplicabilidad a las distintas áreas de conocimiento. Además, estos ejemplos demuestran que se puede realizar tanto una implementación completa del modelo como parcial, usando en este último caso SARLAB para gestionar las comunicaciones.

8. Conclusiones y Desarrollos Futuros

Este último capítulo que cierra la memoria de esta Tesis está orientado a recoger todas las conclusiones derivadas de los trabajos de investigación realizados y las aportaciones principales, lo cual conformará la primera sección del capítulo; así como las líneas de trabajo futuras que se prevé pueden surgir de la Tesis, que serán el objeto de la segunda sección.

8.1. Conclusiones

En las titulaciones técnicas y de ciencias en general, la realización de prácticas de laboratorio es una necesidad incuestionable. Asimismo, el espíritu del Marco Europeo de la Enseñanza Superior, en el que se promueve la formación permanente y la racionalización de los recursos humanos y financieros en las Universidades, es un entorno muy propicio para el uso de técnicas de acceso remoto a esas experiencias de laboratorio, con el objeto de maximizar su aprovechamiento y el beneficio que los alumnos obtienen al realizar las prácticas. Por otra parte, el estado de desarrollo actual de las tecnologías relacionadas con las comunicaciones hace viable la realización de las prácticas a través de Internet. Pero lejos de centrar el problema en un aspecto relacionado sólo con las comunicaciones, el acceso remoto a experiencias de laboratorio

tiene que comprender los tres aspectos fundamentales del problema: el didáctico, el de comunicaciones y el experimental propiamente dicho.

En esta Tesis, se propone una solución global a este problema que se ha denominado modelo general conceptual y que se ha desarrollado en el capítulo tres. En los siguientes apartados se ha ido completando de forma gradual este modelo general para obtener finalmente una herramienta real que pueda explotarse en aquellas instituciones que requieran del uso de prácticas de laboratorio remoto. El proceso de diseño seguido se puede calificar como descendente, porque se ha ido siempre de lo general a lo particular, planteando en todo momento tanto las cuestiones de análisis como las de síntesis. En este sentido, el enfoque en el transcurso de la Tesis no ha sido el desarrollo de una arquitectura concreta como primer objetivo, sino que se ha priorizado la generalización de los desarrollos propuestos, introduciendo las abstracciones necesarias para modelar de forma general todas las cuestiones planteadas. A partir del modelo conceptual general, que es el primero que se propone y el más generalista, se han ido introduciendo los factores tecnológicos que hacen evolucionar ese primer modelo hacia una estructura cada vez más real, y que culmina en un modelo físico y explotable que además es escalable y completo. La eficacia de este modelo físico ha quedado demostrada por su exitosa implementación en diferentes universidades, con prácticas de laboratorio de diversa naturaleza, que están en explotación en este momento y que han sido todas ellas desarrolladas según el modelo propuesto. En algunos casos se ha implementado simplemente SARLAB para solucionar las cuestiones relacionadas con las comunicaciones, y en otros se ha implementado el modelo completo, desde el diseño de la experiencia por parte del profesor, hasta la realización de la misma por parte de los alumnos. En cualquier caso, el sistema de convergencia genérico (ya sea el propuesto en el modelo u otro basado en el uso de otras herramientas como Matlab, LabView, etc.) convierte el problema de control remoto de una experiencia en un problema de gestión de las comunicaciones, de la que se encarga en el modelo propuesto el servidor

SARLAB, gestionando el intercambio de información entre el usuario y el sistema experimental de forma completamente independiente a la naturaleza experimental de este último. El sistema de convergencia genérico facilita asimismo la creación de herramientas que permiten asistir el proceso de diseño de la experiencia.

En esencia, la arquitectura de implementación propuesta proporciona una herramienta que tiene las siguientes características fundamentales: es completa, general, abierta, escalable, asistida y distribuida.

Es completa porque supone una solución para todas las cuestiones relacionadas con los tres aspectos fundamentales del problema: el de enseñanza-aprendizaje (LMS), el de comunicaciones (SARLAB) y el experimental (sistema de convergencia genérico).

Es general porque el sistema de convergencia genérico permite el tratamiento de las experiencias de forma independiente a su naturaleza experimental, por lo que el modelo es aplicable a cualquier área de conocimiento.

Tiene carácter abierto porque, si bien el modelo siempre debe incluir SARLAB como el componente responsable de la gestión de las comunicaciones, deja abierta la posibilidad de uso de cualquier LMS y permite la utilización de cualquier estrategia de diseño en la implementación del sistema de convergencia genérico. Así, el carácter abierto del modelo es consecuencia de que, en cada una de las etapas de concreción del modelo conceptual general hasta la herramienta física, se ha hecho un análisis cuidadoso de todas las alternativas posibles para maximizar el grado de libertad en el diseño de la experiencia. Esto hace posible tanto que diseñadores con gran experiencia puedan utilizar sus propios desarrollos e integrarlos con SARLAB, como que diseñadores con menos experiencia queden asistidos en todos los pasos del diseño de la experiencia, a partir de las herramientas propuestas en el modelo.

Es escalable gracias a las posibilidades presentadas como solución al sistema de convergencia genérico. Se proponen varias plataformas físicas que permiten una implementación adaptada a la dificultad del sistema y de su acceso desde red. Así, se propone el uso desde pequeños MCU hasta potentes procesadores/computadores basados en microprocesadores con varios núcleos. El objetivo prioritario de facilitar el desarrollo de experiencias a diseñadores con bajos conocimientos en tecnologías de las comunicaciones, ha propiciado la creación de herramientas que lo asistan en todas las fases del desarrollo. Así, el modelo comprende el uso de EJSApp (desarrollado por la UNED) para la integración en Moodle, DIGEXLAB para el diseño de las comunicaciones y los distintos elementos de EJS creados en el desarrollo de la Tesis, los cuales permiten el uso de esta aplicación como herramienta de desarrollo rápido de la interfaz de usuario, entre otros.

Finalmente, el modelo es distribuido como consecuencia de la estructura de desarrollo seguida, en la que se delega la gestión del aprendizaje en el LMS. Esto ha propiciado la posibilidad de distribuir experiencias entre varias instituciones, centralizando su acceso en un mismo LMS, lo que constituye la creación de redes de laboratorios distribuidos, como la presentada en el capítulo anterior.

En las redes y experiencias individuales en que el modelo se encuentra en explotación, se pone de manifiesto su carácter general y abierto. De hecho, existen experiencias de una amplia variedad de áreas de conocimiento, en muchas de las cuales se desarrolla una estrategia de convergencia propia. Así por ejemplo, en las implementaciones realizadas por la UNED, se utiliza una estrategia de convergencia propia basada en LabView, que ha sido desarrollada a lo largo de varios años, y que está totalmente asistida y optimizada para las áreas de conocimiento en las que se utiliza. En las implementaciones realizadas por la Universidad Complutense de Madrid (UCM), el sistema de convergencia genérico está basado en arquitecturas PLC. En la implementación realizada por el Círculo de Innovación y Tecnología (CIT) de la

Universidad de Cádiz, y en el que el área experimental es el de redes, ha sido necesario el desarrollo de elementos de gestión de los dispositivos de red. Finalmente, en las implementaciones realizadas por la UHU, las experiencias se han diseñado aplicando el modelo íntegramente.

8.2. Desarrollos Futuros

A partir de lo expuesto a lo largo de esta Tesis, pueden desprenderse diferentes líneas futuras de desarrollo, que se encuadran en distintos aspectos del modelo propuesto. El primero de ellos es la creación de una distribución que permita la instalación de SARLAB de forma amigable para aquellos usuarios con un nivel medio de conocimientos en el ámbito de las comunicaciones.

La inclusión del modelo en plataformas que ofertan experiencias de laboratorio con acceso remoto, tales como UNILabs, hace pensar en la posibilidad de que pueden existir sistemas experimentales ofertados desde diferentes plataformas. Esto es un nuevo problema, que requiere la ampliación de la capacidad de comunicación entre el LMS y SARLAB, y que constituye una línea de desarrollo adicional.

Aunque todas las etapas del diseño de experiencias de laboratorio con acceso remoto y la gestión de las mismas están asistidas, esa asistencia se realiza a partir de varias aplicaciones (EJS, DIGEXLAB, EJSApp, etc.). La unificación de todas ellas en una única aplicación, que integre todas sus funcionalidades, es una línea de desarrollo futura que facilitaría el proceso de diseño completo, ofreciendo una interfaz gráfica de usuario intuitiva que englobara todas las facetas a cubrir.

La experiencia acumulada en los tres años de desarrollo de esta Tesis pone de manifiesto la necesidad de facilitar la gestión de todos los aspectos relacionados con la configuración de red de los dispositivos convergentes. Para ello se propone la creación

de mecanismos *plug and play* para la detección automática de la presencia de componentes convergentes y la configuración de los mismos.

La calidad de servicio en una red pública como Internet es una cuestión fundamental que limita el tipo de experiencias que pueden ser realizadas de forma remota, fundamentalmente aquellas que plantean cerrar el lazo de control a través de esta red. Los recientes avances tecnológicos que pretenden garantizar la calidad de servicio extremo a extremo propicia el estudio de este aspecto en el modelo propuesto como otra línea de desarrollo futuro.

Referencias

- Abdulwahed, M., Nagy, Z.K. 2011. The TriLab, a novel ICT based triple access mode laboratory education model. *Computers & Education*, Vol. 56 pp. 262-274.
- Aktan, B.; Bohus, C.; Crawl, L.A. and Shor, M.H. 1996. Distance Learning Applied to Control Engineering Laboratories. *IEEE Transactions on Education*, Vol. 39, No. 3 pp. 320-326.
- Alamo, J.A. 2000. Remote Web-Enabled Laboratories. *MIT on the road*. Chicago.
- Alexander, D. G. & Smelser, R. E. 2003. Delivering an engineering laboratory course using the Internet, the post office, and a campus visit. *Journal of Engineering Education*, Vol. 92, No. 1 pp. 79-84.
- Aliane, N.; Fernández, J.; Martínez, A. and Ortiz, J. 2007. Un Laboratorio de Ingeniería de Control Basado en Internet. *Información Tecnológica*, Vol. 18, No. 6 pp. 19-26.
- Anderson, R. J., Spong, M. W. 1988. Bilateral control of teleoperator with time delay. *Proceedings of the IEEE conference on decision and control*, Vol. 1 pp. 167-173.
- Andria, G., Baccigalupi, A., Borsic, M., Carbone, P., Daponte, P., De Capua, C., Ferrero, A., Grimaldi, D., Liccardo, A., Locci, N., Lanzolla, A.M.L., Macii, D., Muscas, C., Peretto, L., Petri, D., Rapuano, S., Riccio, M., Salicone, S. and Stefani, F. Agosto 2007. Remote Didactic Laboratory G. Savastano, The Italian Experience for E-Learning at the Technical Universities in the Field of Electrical and Electronic Measurements: Overview on Didactic Experiments. *IEEE Transactions on Instrumentation and Measurement*, Vol. 56, No. 4 pp. 1135-1147.
- Andújar, J.M. y Mateo, T. 2010. Diseño de Laboratorios Virtuales y/o Remotos. Un Caso Práctico. *Revista Iberoamericana de Automática e Informática Industrial*, Vol. 7, No. 1, pp. 64-72.
- Andújar, J.M., Mejías, A. y Márquez, M. 2011. Augmented Reality for the Improvement of Remote Laboratories: An Augmented Remote Laboratory. *IEEE Transactions on Education*, Vol. 54, No. 3, pp. 492-500.
- Azorín, J.M.; Payá, L.; Jiménez, L.M.; García, N. and Sabater, J.M. 2004. Remote Laboratory for Automation Education. *IFAC Workshop on Internet Based Control Education, Grenoble*.
- Barrios, A.; Panche, S.; Duquea, M.; Grisales, V. H.; Prieto, F.; Villac, J. L.; Chevrel, P.; Canu, M. . A multi-user remote academic laboratory system. *Computers & Education*, Volumen 62 pp. 111-122.

- Benmohamed, H.; Lelevé, A.; Prévot, P. 2005. Generic framework for remote laboratory integration. *ITHET 6th Annual International Conference T1A-1*, pp. 1-6.
- Berntzen, R.; Strandman, J.O.; Fjeldly, T.A. and Shur, M.S. 2001. Advanced Solutions for Performing Real Experiments over the Internet. *International Conference on Engineering Education, Oslo, Norway*.
- Blackboard, portal web*. <http://www.blackboard.com>. [Fecha de consulta: 2014].
- Bohus, C.; Crowl, L.A., Aktan, B. and Shor, M.H. 1996. Running Control Engineering Experiments Over the Internet. *Proceedings of the IFAC World Congress, San Francisco*, Vol. 6 pp. 25-33.
- Brightspace, portal web*. <http://www.brightspace.com>. [Fecha de consulta: 2014].
- Buitrago, J.A.; Giraldo, F.D. and Lamprea, J.A. 2011. Remote Access Lab for Mitsubishi RV-2AJ Robot. *Robotics Symposium, 2011 IEEE IX Latin American and IEEE Colombian Conference on Automatic Control and Industry Applications (LARC)*, pp. 1-7.
- Buiu, C.; Moanta, N. . Using Web services for Designing a Remote Laboratory for Motion Control of Mobile Robots. *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*.
- Calvo, I., Zulueta, E., Oterino, F., & Lopez-Guede, J. M. 2009. A remote laboratory for a basic course on control engineering. *International Journal of Online Engineering*, Vol. 5, No. 3 pp. 8-13.
- Campos, C.; Lopes, J. and Nascimento, C. 2007. Remote Ccontrol of Real Experiments via Internet. *Proceedings of International Congress of Mechanical Engineering, Brasilia*.
- Carusi, F.; Casini, M.; Prattichizzo, D. and Vicino, A. 2004. Remote Control of a Lego Mobile Robot through the Web. *Actas del Taller de Educación en Control basada en Internet, Grenoble, Francia*.
- Casini, M., Prattichizzo, D. and Vicino, A. 2004. The Automatic Control Telelab: A Web based Technology for Distance Learning. *IEEE Control Systems Magazine*, Vol. 24, No. 3 pp. 36-44.
- Casini, M.; Prattichizzo, D. and Vicino, A. 2003. The automatic control telelab: a user-friendly interface for distance learning. *IEEE Transactions on Education*, Vol. 46, No. 2 pp. 252-257.
- Casini, M.; Prattichizzo, D. and Vicino, A. 2003. E-Learning by Remote Laboratories: a new tool for control education. *6th IFAC Conference on Advances in Control Education*.
- Chang, T. and Hung, D. 2000. Web-Based Distance Experiments: Design and Implementation. *IEEE International Conference on Engineering Education, Taipei, Taiwan*.

- Chen, S.H.; Chen, R.; Ramakrishnan, V.; Hu, S.Y.; Zhuang, Y.; Ko, C.C. and Chen, B.M. 1999. Development of Remote Laboratory Experimentation through Internet. *Proceeding IEEE Hong Kong Symp. Robot. Control*, Vol. 1 pp. 756-760.
- Claroline, portal web. <http://www.claroline.net>. [Fecha de consulta: 2014].
- Corter, J.E.; Nickerson, J.V.; Esche, S.K.; Chassapis, C. 2004. Remote Versus Hands-On Labs: A Comparative Study. *Proceedings of 34th ASEE/IEEE Frontiers in Education Conference, October 20 – 23*.
- Dalton, B. 2001. Ph.D. Thesis. Techniques for Web Telerobotics. *Department of Mechanical and Materials Engineering University of Western Australia*.
- Deniz, D.Z.; Bulancak, A. and Özcan, G. 2003. A Novel Approach to Remote Laboratories. *ASEE/IEEE Frontiers in Education Conference*.
- DeWeerth, S.P. and Knight, C.D. 1996. A Shared Remote Testing Environment for Engineering Education. *Proc. Frontiers in Education Conference*, Vol. 1 pp. 1003-1006.
- DeWeerth, S.P. and Knight, C.D. 1997. A Distance Learning Laboratory for Engineering Education. *International Conference on Engineering Education*.
- Dixon, W.E.; Dawson, D.M.; Costic, B. T. and Queiroz, M.S. 2002. A MATLAB-Based Control Systems Laboratory Experience for Undergraduate Students: Toward. *IEEE Transactions on Education*, Vol. 45, No. 3 pp. 218-226.
- Dokeos, portal web. <http://www.dokeos.com>. [Fecha de consulta: 2014].
- Dormido, S. . Control learning: present and future. *Annual Reviews in Control*,, Volume 28, Issue 1 pp. 115-136.
- Dusi, M.; Crotti, M.; Gringoli, F. and Salgarelli, L. 2009. Tunnel Hunter: Detecting application-layer tunnels with statistical fingerprinting. *Computer Networks* , Vol. 53 pp. 81–97.
- EJS. *Easy Java Simulation*.. Esquembre, F. <http://fem.um.es/Ejs/>. [Fecha de consulta: Abril 2015].
- Ernest, E.W. and Kopplint, J.O. 1962. Modern Objectives and Methods for Laboratory Instruction. *IRE TRANSACTIONS ON EDUCATION*, September pp. 168-172.
- ERNST, E.W. 1983. A New Role for the Undergraduate Engineering Laboratory. *ERNST, E.W.*, Vol. 26, No.2 pp. 49-51.
- European Comission. *Bologna Process EHEA*. . European Comission <http://www.ehea.info/article-details.aspx?ArticleId=5>. [Fecha de consulta: Enero 2013].
- Exact-learning, portal web. <http://www.exact-learning.com>. [Fecha de consulta: 2014].

- Fabregas, E.; Farias, G.; Dormido-Canto, S.; Dormido, S.; Esquembre, F. 2011. Developing a remote laboratory for engineering education. *Computers & Education*, Vol. 57 pp. 1686–1697.
- Faltin, N.; Böhne, A.; Tuttas, J. and Wagner B. 2002. Distributed Team Learning in an Internet-Assisted Laboratory. *International Conference on Engineering Education*.
- Farias, G; De Keyser, R; Dormido, S; Esquembre, F. 2010. Developing Networked Control Labs: A Matlab and Easy Java Simulations Approach. *IEEE Transactions on Industrial Electronics*, Vol. 57, No. 10 pp. 3266-3275.
- Ferrero, A. and Piuri, V. . A Simulation Tool for Virtual Laboratory Experiments in a www Environment. *Proceedings of IEEE Instrumentation and Measurement Technology Conference 1988*, pp. 102-107.
- Fjeldly, T. A.; Shur, M. S.; Shen, H. and Ytterdal, T. 1999. Automated Internet Measurement Laboratory (AIM-Lab) for Engineering Education. *IEEE Frontiers in Education Conference, San Juan, Puerto Rico*.
- Fjeldly, T.A.; Strandman, J.O. and Berntzen, R. 2002. Lab-on Web – A Comprehensive Electronic Device Laboratory on a Chip Accessible via Internet. *International Conference on Engineering Education, Manchester, U.K.*
- Frangu, L. and Chiculita, C. 2010. Remote Laboratory Allowing Full-Range Student-Designed Control Algorithm. *International Conference on Computational Intelligence and Software Engineering*.
- García Zubía, J. and Alves, G. 2011. *Using Remote Labs in Education*. Servicio de publicaciones de la Universidad de Deusto.
- García Zubía, J., Sáenz Ruiz de Velasco, J. 2005. Diseño de laboratorios remotos virtuales: WebLab. *XI Jornadas de la Enseñanza Universitaria de la informática JENUUI 2005 Universidad Europea de Madrid*.
- García-Zubia, J.; Orduña, P.; Irurzun, J.; Angulo, I. and Hernández-Jayo, U. 2009. Integración del laboratorio remoto WebLab-Deusto en Moodle. *In MoodleMoot Euskadi 2009*.
- Gilberta, D.; Yearwortha, M.; Oliverb, L. 2014. Systems Approach to the Development and Application of Technical Metrics to Systems Engineering Projects. *Procedia Computer Science*, Vol. 28 pp. 71 – 80.
- Gillet, D.; Salzmann, C.; Longchamp, R. and Bonvin, D. 1997. Telepresence: an Opportunity to Develop Real-World Experimentation in Education. *European Control Conference*.
- Goldberg, K., Gentner, S., Sutter, C., & Wiegley, J. 1995. Desktop Teleoperation via the Word Wide Web. *IEEE Internationa Conference on Robotics and Automation*, pp. 654-659.

- Goldberg, K., Gentner, S., Sutter, C., & Wiegley, J. 1999. The Mercury Project: A Feasibility Study for Internet Robots. *IEEE Robotics & Automation Magazine*, Vol. 7, No. 1 pp. 35-40.
- Golijanek-Jedrzejczyk, A.; Jedrzejczyk, D.; Referowski, L. and Swisulski, D. 2002. Educational Laboratory Stand for Remote Measurement Sensors' Dynamic Characteristics. *MKM Conference*.
- Guimarães, E., Maffeis, A., Pereira, J., Russo, B., Cardozo, E. and Bergerman, M. 2003. A virtual laboratory for Mobile Robot experiments. *IEEE Transaction on Education*, Vol. 46, No. 1 pp. 37-42.
- Hahn, H.H. 2000. Remote Laboratories for Control Education. *IEEE Conference on Decision and Control*, Vol. 1 pp. 895-900.
- Henderson, S. ; Feiner, S. 2011. Exploring the Benefits of Augmented Reality Documentation for Maintenance and Repair. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 17, No. 10 pp. 1355 - 1368.
- Henry, J. 1999. Using the World Wide Web for Teaching Control Systems Design. *Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design*.
- Henry, J. and Knight, C. 2000. Improving Laboratories with Internet Controlled Equipment and Student Support. *2000 ASEE Southeast Section Conference*.
- HPI. Hasso Plattner Institute. Distributed Control Lab. <http://www.dcl.hpi.uni-potsdam.de/research/dcl/>. [Fecha de consulta: Noviembre 2014].
- Hsu, S.; Alhalabi, B.A. and Ilyas, M. 2000. A Java-Based Remote Laboratory for Distance Learning. *International Conference on Engineering Education*.
- Hua, J.; Cui, Y.; Yang, Y. and Li, H. 2013. Analysis and prediction of jitter of internet one-way time-delay for teleoperation systems. *IEEE International Conference on Industrial Informatics*, Vol. 1 pp. 612-617.
- Indrusiak, L.S., Glesner M. and Reis, R. DEcember 2007. On the Evolution of Remote Laboratories for Prototyping Digital Electronic Systems. *IEEE Transaction on Industrial Electronics*, Vol. 54, No. 6 pp. 3069-3077.
- ITR. *Internet Traffic Report*. <http://www.internettrafficreport.com/>. [Fecha de consulta: Enero 2015].
- Jara, A. C., Candelas, A. F., Puente, T. S., & Torres, F. 2011. Hands-on experiences of undergraduate students in automatics and robotics using a virtual and remote laboratory. *Computers & Education*, Vol. 57 pp. 2451-2461.
- JillServer. <http://profesores.eie.ucv.cl/hvargas/jil/jil.html>. [Fecha de consulta: Diciembre 2014].

- Jiménez, L. M.; Puerto, R.; Reinoso, O.; Fernández, C. and Ñeco, R. 2005. Recolab: Laboratorio Remoto de Control Utilizando Matlab y Simulink. *Revista Iberoamericana de Automática e Informática*, Vol. 2, No. 2 pp. 64-72.
- Jochheim, A. and Röhrig, C. 1999. The Virtual Lab for Teleoperated Control of Real Experiments. *Conference on Decision and Control*.
- Jordán, R.A.; Martínez, A.; Tamura, E.; Martínez M.F. and Olarte, C.A. 2002. A Remote Laboratory for an Automatic Control Electromechanical System. *International Conference on Engineering Education, Manchester, U.K.*
- Juanping, Z. and Xianwen, G. 2009. Time-delay analysis and estimation of internet-based robot teleoperation system. *Control and Decision Conference, Chinese*, Vol. 1 pp. 4643 - 4646.
- Junge, T.F. and Schmid, C. 2000. Web-Based Remote Experimentation Using a Laboratory-Scale Optical Tracker. *American Control Conference, Chicago, Illinois*.
- Kahoraho, E. and Larrauri, J.I. 2002. A WebLab System for the Study of the Control and Protection of Electric Motors. *Proceedings of Telecommunication, Electronics and Control*, Vol. 1 pp. 7.
- Kent, G. and Cardt, H. 1961. Experiment in Laboratory Education. *IRE TRANSACTIONS ON EDUCATION*, June pp. 62-66.
- Khamis, A.M. 2003. Tesis PhD. Interacción Remota con Robots Móviles Basada en Internet. *Universidad Carlos III, Madrid*.
- Ko, C.C.; Chen, B.M. ; Jianping Chen ; Zhuang, Y. ; Chen Tan, K. 2002. Development of a web-based laboratory for control experiments on a coupled tank apparatus. *IEEE Transaction on Education*, Vol. 44, No. 1 pp. 76-86.
- Ko, C.C.; Chen, B.M.; Hu, S.; Ramakrishnan, V. 2002. A Web-Based Virtual Laboratory on a Frequency Modulation Experiment. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 31, No. 3 pp. 295-303.
- Labshare. <http://labshare.edu.au>. [Fecha de consulta: Marzo 2015].
- Laget, H.; Valle, F.; Tadeo, F. 2003. Web Based Rremote Control of an Electro-Pneumatic Process. *XXIV Jornadas de Automática, León*.
- Lasky, V.L.; Liu, D.K.; Murray, S.J. and Choy, K.K.L. 2005. A Remote PLC system for e-Learning. *Proceedings of the ASEE/AaeE 4th Global Colloquium on Engineering Education*.
- Lindsay, E.D. and Good, M.C. 2005. Effects of Laboratory Access Modes upon Learning Outcomes. *IEEE Transaction on Education*, Vol. 48, No. 4 pp. 619-631.
- LWB Laboratories without borders. <http://http://chem.engr.utc.edu/>. [Fecha de consulta: Enero 2015].

- Magin, D., & Kanapathipillai, S. 2000. Engineering students' understanding of the role of experimentation. *European Journal of Engineering Education*, Vol. 25, No. 4 pp. 351-358.
- Maiti, A.; Kist, A.A. and Maxwell, A.D. 2015. Real-Time Remote Access Laboratory with Distributed and Modular Design. *IEEE Transaction on Industrial Electronics*. Accepted for publication. DOI 10.1109/TIE.2014.2374572.
- Marner, M.R. ; Smith, R.T. ; Walsh, J.A. ; Thomas, B.H. 2014. Spatial User Interfaces for Large-Scale Projector-Based. *IEEE Computer Graphics and Applications*, Vol. 34, No. 6 pp. 74 - 82.
- Mejías, A. y Andújar, J.M. 2011. A Pilot Study of the Effectiveness of Augmented Reality to Enhance the Use of Remote Labs in Electrical Engineering Education. *Journal of Science Education and Technology*, pendiente de publicación, pp. n.d.
- Mejías, A.; Andújar, J.M. and Márquez, M.A. 2014. Digital Electronics Augmented Remote Laboratory: DEARLAB. *International Journal of Engineering Education*, Vol. 30, No. 4 pp. 950-963.
- Mercury Project*. Goldberg, K. and Mascha, M. <http://rieff.ieor.berkeley.edu/usc-archive/dept/raiders/>. [Fecha de consulta: Marzo 2015].
- Moodle, portal web*. <https://moodle.org>. [Fecha de consulta: 2014].
- Myakotnykh, E.S. ; Helvik, B.E. 2009. Analysis of evolution scenarios for end-to-end quality of services provisioning in the internet. *International Conference on Ultra Modern Telecommunications & Workshops*, Vol. 1 pp. 1-5.
- Nakano, H.; Matsuo, S. and Nakamura, Y. 2001. Distance Education Server for the Study of Remote Sensing. *International Conference on Information Technology Based Higher Education and Training, Kumamoto, JAPAN*.
- Nickerson, J.V., Corter, J.E., Esche, S.K., Chassapis, C 2007. A model for evaluating the effectiveness of remote engineering laboratories and simulations in education. *Computers & Education* , Vol. 49 pp. 708-725.
- Nikolic, S. ; Vial, P.J. ; Ros, M. ; Stirling, D. ; Ritz, C. 2014. Improving the Laboratory Learning Experience: A Process to Train and Manage Teaching Assistants. *IEEE Transactions on Education*, PP, 99 pp. 1.
- Odlyzko, A. 1999. The current state and likely evolution of the Internet. *Global Telecommunications Conference*, Vol. 3 pp. 1869-1876.
- Pérez, J. 2008. *Oportunidades y Desafíos de la Banda Ancha*. Grupo de Análisis y Prospectiva del Sector de las Telecomunicaciones.
- Perić, N. and Petrović, I. 2000. Virtual Laboratory for Automatic Control and Supervision-Challenges and Opportunities. *Proceedings of International Conference on Technology Transfer for Economic Development, Zagreb, Croatia*.

- Prashant K. J.; Yuxiang Gu, R. 2008. Broadcasting Engineering Laboratories — Audio/Video and Data — in Real-Time over the Internet. *Advances in Engineering Education Journal*, Vol. 1, No. 2 pp. 2-17.
- Ramos, C.A. and Ramírez, J.M. 2005. Virtual Tools for Remote Analysis and Simulation. *International Conference on Industrial Electronics and Control Applications*.
- Rasche, A.; Rabe, B.; Tröger, P. and Polze, A. 2004. Distributed Control Lab. *Proceedings of the International Workshop on e-learning and Virtual and Remote Laboratories*, Vol. 1 pp. 150-160.
- Raychaudhuri, D.; Mandayam, N.B. 2012. Frontiers of Wireless and Mobile Communications. *Proceedings of the IEEE Journals and Magazines*, Vol. 100, No. 4 pp. 824-840.
- Reguera, P. and Fuertes, J.J. 2004. Control Remoto de Posición con Java. *Jornadas de Automatica, Ciudad Real*.
- RFC 1930. *Estándar de comunicaciones*. Guidelines for creation, selection, and registration of an Autonomous System. [Fecha de consulta: Abril 2015].
- Rohrig, C. and Jochheim, A. 1999. The Virtual Lab for Controlling Real Experiments via Internet. *Proceedings of the IEEE International Symposium on Computer Aided Control System Design, Kohala Coast-Island of Hawaii*.
- S. Dormido, J. Sánchez, F. Esquembre, H. Vargas, S. Dormido-Canto, R. Dormido, N. Duro, G. Farias, M^a. A. Canto. 2007. The development of web-based virtual laboratories using Easy Java Simulations (Ejs). *Proceedings of the Remote Engineering and Virtual Instrumentation Conference*.
- Saad, M.; Saliyah-Hassane, H.; Hassan, H.; El-Guetioui, Z. and Cheriet, M. 2001. A Synchronous Remote Accessing Control Laboratory on the Internet. *International Conference on Engineering Education, Oslo, Norway*.
- SAHARA Labs. <http://sourceforge.net/projects/labshared-sahara/>. [Fecha de consulta: Febrero 2015].
- Sakaiproject, *portal web*. <https://sakaiproject.org>. [Fecha de consulta: 2014].
- Salzmann, C., Gillet, D., Latchman, H. A. & Crisalle, O. D. 1999. On-line Engineering Laboratories: Real-Time Control over the Internet. *Proceedings of the Annual Conference & Exposition*.
- Schmid, C. . VCLAB - The Virtual Control Engineering Laboratory. *Proceeding of Symposium on System Identification, 2000*, pp. 1-7.
- Sell, R. and Rüttemann, T. 2014. The International Cooperation on Remote Laboratories Conducted with Engineering Didactics. *Proceedings on International Conference on Remote Engineering and Virtual Instrumentation*, pp. 187-190.

- SLARToolkit - Silverlight and Windows Phone Augmented Reality Toolkit*. Schulte, R. <http://slartoolkit.codeplex.com/>. [Fecha de consulta: Marzo 2011].
- Smartta, C. and Ferreira, S. 2013. Applying Systems Engineering to Survey Research. *Procedia Computer Science*, Vol. 16 pp. 1102 – 1111.
- Stallings, W. 2011. *Comunicaciones y Redes de Computadores, 6ª edición*. Prentice Hall.
- Szymaniak, M.; Presotto, D.; Pierre, G. and Steen, M. 2008. Practical large-scale latency estimation. *Computer Networks*, Vol. 52 pp. 1343–1364.
- Szymaniak, M.; Presotto, D.; Pierre, G. and Steen, M. 2008. Practical large-scale latency estimation. *Computer Network*, Vol. 52 pp. 1343-1364.
- Tanenbaum, A.S. 1996. *Sistemas Operativos Distribuidos*. Prentice Hall.
- Tanenbaum, A.S. and Wetherall, D.J. 2014. *Redes de Computadoras*. Pearson.
- Taylor, K.; Trevelyan, J. 1995. Australia's Telerobot On The Web. *InterJournal Complex Systems*, Vol. 12 pp. 1-6.
- Tilbury, D., Messner, W. 1998. Controls Education on the WWW: Using MATLAB for Control Design, Simulation and Visualization. *Proceedings of the International Mechanical Engineering Congress and Exposition*, Vol. 1 pp. 65-70.
- Tuttas, J. and Wagner, B. 2001. Distributed OnLine Laboratories. *International Conference on Engineering Education*. Oslo, Noruega.
- UNILabs. UNED.. <http://unilabs.dia.uned.es/>. [Fecha de consulta: Abril 2015].
- UWA Telerobot. Taylor, K, and Dalton, B. <http://telerobot.mech.uwa.edu.au/Telerobot/index.html>. [Fecha de consulta: Febrero 2015].
- Vargas, H., Sánchez, J., Jara, C.A., Candelas F.A. Torres, F. and Dormido, S. 2011. A Network of Automatic Control Web-Based Laboratories. *IEEE Transactions on Learning Technologies*, Vol. 4, No. 3 pp. 197-208.
- Vargas, H., Sánchez, J., Jara, C.A., Candelas, F.A., Torres, F. and Dormido S. Julio 2011. A Network of Automatic Control Web-Based Laboratories. *IEEE Transactions on Learning Technologies*, Vol. 4, No. 3 pp. 197-208.
- Vargas, H.; Sanchez-Moreno, J.; Dormido, S.; Salzmann, C.; Gillet, D. and Esquembre, F. 2009. Web-Enabled Remote Scientific Environments. *Computing in Science and Engineering*, Vol. 11, No. 3 pp. 34-46.
- Vargas, H.; Sánchez, J.; Dormido, S.; Farias, G.; Duro, N.; Dormido-Canto, R.; Dormido-Canto, S. and Esquembre, F. 2006. Web-Based Learning Resources for Vocational Training for Automation Technicians. *7th IFAC Symposium on Advances in Control Education*.

WebCT, portal web. <http://www.webct.com>. [Fecha de consulta: 2014].

WebLab. Alamo, J.A.. <http://newsoffice.mit.edu/2001/weblabcomment>. [Fecha de consulta: Diciembre 2014].