



ugr

Universidad
de **Granada**

ESTUDIOS DE GRADO EN INGENIERÍA DE TECNOLOGÍAS DE
TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

Servicio de proyección de material docente basado en Raspberry Pi

Autor

JUAN RAMÓN GUTIÉRREZ MARTÍNEZ

Director

JORGE NAVARRO ORTIZ



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Granada, 7 de julio de 2014



ugr

Universidad
de **Granada**

Servicio de proyección de material docente basado en Raspberry Pi

Realizado por:

JUAN RAMÓN GUTIÉRREZ MARTÍNEZ

Dirigido por:

JORGE NAVARRO ORTIZ

Departamento:

TEORÍA DE LA SEÑAL, TELEMÁTICA Y COMUNICACIONES



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Granada, 7 de julio de 2014

Servicio de proyección de material docente basado en Raspberry Pi

Juan Ramón Gutiérrez Martínez

Palabras clave:

Raspberry Pi, docencia, material docente, proyección, exposición, diapositivas.

Resumen:

Ante la nueva forma de enseñanza en la que es necesaria la utilización de material en forma de diapositivas, se hace tedioso tener que transportar con el portátil en todo momento, además de la incomodidad de tener que estar conectado mediante el cable VGA.

Ante esta cuestión surge este proyecto cuyo objetivo es poder hacer uso de los proyectores en las aulas docentes sin necesidad de tener un equipo conectado a través de un cable VGA. Para ello se conectará un PC del tamaño de una tarjeta de crédito (Raspberry Pi) a través del interfaz HDMI del proyector, de forma que el profesor pueda navegar a través de sus transparencias u otros materiales docentes desde cualquier equipo conectado a la red WiFi de la UGR, esto es, sin necesidad de estar conectado a través de un cable VGA.

De esta manera, se mejora la usabilidad del servicio de proyección y también la movilidad del docente por el aula, ya que podría utilizar el portátil desde cualquier lugar del aula.

Para ello, se ha de diseñar un servidor con autenticación de usuarios para el almacenaje del material docente, así como integrar los diferentes protocolos que sean necesarios para que la Raspberry Pi visualice dicho material a través del proyector y pueda ser controlada a través de los dispositivos inalámbricos mencionados.

Projection service of docent material based on Raspberry Pi

Juan Ramón Gutiérrez Martínez

Keywords:

Raspberry Pi, teaching, teaching material, projection, exposition, slides.

Abstract:

In the new and predominant way of teaching that currently exists, which is necessary for almost any class, the using of docent material in form of transparencies or slides shown by teachers to students. A number of problems arise in the using of this teaching method when performing that work, of which no one has noticed, or at least nobody has not given importance or never has not anybody stopped to think carefully a global solution for all these.

For instance, it becomes tedious for the professor, or student, having to carry his personal computers at all times to can display such material. In addition, on the other hand, is much the inconvenience of having to have the computer continuously connected by a cable. Something that greatly reduced or eliminated in some cases, the possibility of mobility of professors for classroom teaching.

The market has been studied and all possible solutions which have been implemented, or could be nowadays or in the no far future, and it is has not found any solution which is good enough, to satisfy all needs of professor and solvent the problems set forth above in turn. And with some of these issues, the idea for this project arises, whose main objective is to make use of projectors present in teaching classrooms without having a computer connected to them via a cable constantly, either VGA or HDMI.

To the implement, it is connected a PC with reduce dimensions, a size of a credit card more o less, but also with high performance in relation to this (Raspberry Pi), through the HDMI interface of projector. Thus it is intended that the professor can navigate through your slides or other teaching materials thanks to a client application and from any device connected to the Wi-Fi network of the UGR, namely, without necessity of being connected by a cable.

In this manner, the usability of the service projection that is currently used is improved, as well as the mobility of professors is also enhanced in

the teaching classroom, as in this case, since there is no need to connect the computer with the cable, could also use any kind of device such as your smartphone, tablet or even the laptop itself but in this case, from anywhere in the classroom. Something that also improve the interconnection of devices and unify all existing technologies which we are working all the time.

For the final implementation of this project, it is has designed an authentication server with a great database to differentiate the access of different users, to the files server where all teaching material which be used in the future will be stored. As well as, they are had to integrate the different protocols which are necessary for the server which are installed in the Raspberry Pi can to display such teaching material through the HDMI interface of projector, and that it can be controlled through the aforementioned wireless devices.

Yo, Juan Ramón Gutiérrez Martínez, alumno de la titulación Grado en Ingeniería de Tecnologías de Telecomunicación de la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicaciones de la Universidad de Granada, con DNI XXX, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro y/o departamento para que pueda ser consultada libremente por las personas que lo deseen.

Fdo: Juan Ramón Gutiérrez Martínez

Granada a 7 de julio de 2014

D. Jorge Navarro Ortiz, profesor del área de Telemática del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada, como director del siguiente Trabajo Fin de Grado de D. Juan Ramón Gutiérrez Martínez.

Informa:

que el presente trabajo, titulado:

Servicio de proyección de material docente basado en Raspberry Pi

Ha sido realizado y redactado bajo mi supervisión por el mencionado alumno, y con esta fecha autorizo la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expido y firmo el presente informe.

Fdo: Jorge Navarro Ortiz

Granada a 7 de julio de 2014

Agradecimientos

Ha sido un largo camino hasta llegar a este punto. Más de una persona me advirtió de la dificultad que esto supondría, y no les faltaban razón. Pero a día de hoy, no me arrepiento en absoluto de lo decidido en aquel momento.

Este proyecto representa un momento que todo estudiante, que disfruta con lo que ha estado estudiando a lo largo de estos años de universidad, espera ansiadamente. La culminación de una etapa importante de mi vida, la realización de un sueño, y la confirmación de las expectativas que todos los que me han apoyado tenían sobre mí.

Me siento orgulloso con el resultado de este proyecto, y esto se lo debo a Jorge, mi tutor, gracias a la idea que en su día me dio, he tenido la oportunidad de terminar esta etapa con algo que me gusta y me apasiona. Además, he de agradecer todos los consejos que me ha brindado a lo largo de toda esta etapa.

Por supuesto, debo agradecer a mis padres, y en general a mi familia, su apoyo incondicional y su total confianza en mí desde el primer momento que tomé la decisión. Porque saben perfectamente de la dificultad que esto ha supuesto y de todos los obstáculos que he tenido que superar.

Tampoco puedo olvidar a los profesores que han sabido transmitirme sus conocimientos de esta profesión. Ni tampoco, a todos los compañeros que he tenido en estos años, por los buenos momentos, tanto en clase como fuera de esta, y en especial a los 6 con los he compartido interrail. Y por su puesto a Miriam.

Gracias a todos

Índice general

1. Introducción y referencias bibliográficas	1
1.1. Contexto	1
1.1.1. Evolución de los métodos docentes	1
1.1.2. Tecnología en las aulas	4
1.2. Motivación	5
1.3. Estudio del estado del arte	7
1.3.1. Proyectores inalámbricos	7
1.3.2. Pizarra interactiva y <i>Tablet Monitor</i>	8
1.3.3. Apple TV y <i>AirPlay</i>	9
1.3.4. Wireless Display y Miracast	12
1.3.5. Chromecast	13
1.3.6. Resumen	13
1.4. Objetivos y alcance del proyecto	15
1.5. Estructura de la memoria	15
1.6. Principales referencias utilizadas	16
2. Planificación y estimación de costes	19
2.1. Recursos	19
2.1.1. Humanos	19
2.1.2. Hardware	19
2.1.3. Software	20
2.2. Fases de desarrollo	20
2.2.1. Revisión del estado del arte	20
2.2.2. Especificación de los requisitos	21

2.2.3.	Diseño	21
2.2.4.	Implementación	21
2.2.5.	Evaluación y pruebas	21
2.2.6.	Documentación	21
2.3.	Estimación de costos	21
2.3.1.	Recursos humanos	22
2.3.2.	Herramientas	22
2.4.	Presupuesto	23
3.	Análisis de objetivos y metodología	25
3.1.	Descripción global	25
3.2.	Especificación de requisitos	28
3.2.1.	Requisitos funcionales	28
3.2.2.	Requisitos no funcionales	28
3.2.3.	Requisitos de la interfaz de usuario	31
3.3.	Metodología	32
3.4.	Valoración de objetivos	34
4.	Diseño y resolución del trabajo	35
4.1.	Diseño	35
4.1.1.	Diseño global	35
4.1.1.1.	Diseño previo	35
4.1.1.2.	Estudio de las posibles implementaciones	37
4.1.1.3.	Diseño final	41
4.1.2.	Diseño de los servidores	42
4.1.2.1.	Servidor WEB	43
4.1.2.2.	Servidor RPi	46
4.1.3.	Diseño de la aplicación cliente	49
4.1.4.	Diseño del protocolo	51
4.2.	Implementación	54
4.2.1.	Servidor SFTP	54
4.2.2.	Servidor LDAP	55

ÍNDICE GENERAL

4.2.3. Servidor RADIUS	56
4.2.4. Servidor HTTP	57
4.2.4.1. index.php	57
4.2.4.2. registrar.php	58
4.2.4.3. inicio.php	59
4.2.4.4. applet.php	61
4.2.4.5. cambiarPassword.php	61
4.2.4.6. cerrarSerion.php	62
4.2.5. Servidor RPi	62
4.2.6. Cliente RPi	67
5. Evaluación y pruebas	71
5.1. Análisis del tráfico	71
5.1.1. Entorno web	72
5.1.2. Entorno de la aplicación cliente	75
5.2. Escenario típico de aplicación	76
6. Conclusiones y líneas futuras	83
6.1. Resultados	83
6.2. Líneas de trabajo futuras	85
6.3. Valoración personal	86
Anexos	87
A. Manual de usuario	89
A.1. Interfaz web	89
A.2. Aplicación cliente	96
A.3. Servidor RPi	97
B. Manual de instalación	99
B.1. Raspberry Pi	99
B.2. Servidor VSFTPD	103
B.3. Servidor OpenLDAP	105

ÍNDICE GENERAL

B.4. Servidor FreeRADIUS	109
B.5. Servidor Apache	111
B.6. Servidor RPi	111
Bibliografía final	113

Índice de figuras

1.1. Estructura de funcionamiento de sesión multimedia estándar con iPad [26]	11
2.1. Diagrama de Gantt con la planificación temporal	20
2.2. Coste monetario asociado a cada fase del proyecto	23
3.1. Diagrama de flujo de datos de alto nivel	29
4.1. Diseño previo	36
4.2. Diseño global de la implementación	37
4.3. Uso de servidores HTTP	38
4.4. Diseño general de funcionamiento	42
4.5. Interfaz web de usuario	43
4.6. Elementos del servidor WEB	44
4.7. Diagrama de estados del ServidorRPi	47
4.8. Diagrama de estados del ClienteRPi	50
4.9. Intercambio de mensajes ClienteRPi-ServidorRPi	52
4.10. Fichero de configuración grupos.ldif	56
4.11. Fichero de configuración profesor.ldif	56
4.12. Interfaz portada	57
4.13. Interfaz página principal	60
5.1. Escenario de aplicación del ejemplo	72
5.2. Captura de Wireshark del establecimiento de la conexión	72
5.3. Captura de Wireshark del mensaje Alert	73

ÍNDICE DE FIGURAS

5.4. Captura de Wireshark de la autenticación de usuario	73
5.5. Captura de Wireshark del protocolo RADIUS	74
5.6. Captura de Wireshark de la conexión FTP	75
5.7. Captura de Wireshark del intercambio de mensajes en la co- nexión	75
5.8. Captura de Wireshark del intercambio de mensajes en una petición	75
5.9. Captura de Wireshark del intercambio de mensajes en el cierre de la conexión	76
5.10. Captura de Wireshark del seguimiento del stream TCP . . .	76
5.11. Escenario típico de aplicación	77
5.12. Interfaz del servidor RPi	78
5.13. Captura de Wireshark del inicio de una conexión en un en- torno de aplicación real	78
5.14. Captura de Wireshark de autenticación de usuario en un en- torno de aplicación real	79
5.15. Captura de Wireshark de una conexión FTP en un entorno de aplicación real	79
5.16. <i>Debug</i> del servidor RPi en un entorno de aplicación real . . .	80
5.17. Contenido del fichero <i>log</i> del servidor RPi en un entorno de aplicación real	81
A.1. Portada	90
A.2. Portada para los alumnos	90
A.3. Página principal	91
A.4. Ejemplo de notificación en la portada	91
A.5. Página para crear una nueva cuenta	92
A.6. Ejemplo de notificación en la página para crea una nueva cuenta	93
A.7. Página de configuración para cambiar de contraseña	93
A.8. Página principal con opciones desplegadas	94
A.9. Página de selección de lugar de proyección	95
A.10. Página con la aplicación cliente para un documento PDF . .	95
A.11. Página con la aplicación cliente para una imagen	96
A.12. Menú archivo de la aplicación cliente para un documento PDF	96

ÍNDICE DE FIGURAS

A.13.Menú ver de la aplicación cliente para un documento PDF . .	97
A.14.Menú ver de la aplicación cliente para una imagen	97
A.15.Página con la aplicación cliente para un documento PDF . .	98
A.16.Página con la aplicación cliente para un documento PDF . .	98
B.1. Ejemplo del panel de configuración	100
B.2. Ejemplo del panel de configuración	101
B.3. Ejemplo del panel de configuración	101
B.4. Interfaz de configuración del servidor OpenLDAP	106
B.5. Resultado de añadir nuevas entradas a la base de datos . . .	108
B.6. Resultado de realizar una búsqueda en la base de datos . . .	109
B.7. Comprobación del correcto funcionamiento del servidor Free- RADIUS	110

Índice de tablas

1.1. Evolución de Powert Point	3
1.2. Comparación de las soluciones existentes	14
2.1. Planificación temporal del proyecto	20
2.2. Coste monetario personal del proyecto	22
2.3. Coste monetario material del proyecto	24
2.4. Presupuesto total	24

Capítulo 1

Introducción y referencias bibliográficas

1.1. Contexto

El auge en las últimas décadas en el uso de la tecnología es una realidad. Actualmente, las tecnologías de la información y la comunicación (TICs) están sufriendo un desarrollo vertiginoso, algo que afecta prácticamente a todos los campos de nuestra sociedad, y la educación no es una excepción. Las tecnologías son cada vez más utilizadas en los nuevos métodos docentes, por parte sobre todo, de las universidades [1]. Lejos de ser ajenos a todo este cambio, los métodos de aprendizaje y de enseñanza se han visto modificados conforme a la evolución que se produce cada día en la tecnología de una forma notable.

Las TICs se han convertido en algo muy arraigado en los centros educativos. Su uso ha fomentado cambios cualitativos en la forma de enseñanza, especialmente en cuanto a la presentación de los contenidos por medios audiovisuales.

1.1.1. Evolución de los métodos docentes

En el comienzo de la enseñanza, el único recurso en el que podía apoyarse un docente, y por tanto el único utilizado por éste era la pizarra y los libros de texto que los alumnos debían comprar. Y así fue hasta la invención del proyector de retroalimentación, el cual se ha estado utilizando hasta hace bien poco en muchas de las clases. Este equipo utilizaba la luz para proyectar una única imagen sobre una pared generalmente blanca. Se utilizaban por entonces transparencias de acetato, una buena solución cuando el docente

necesitaba apoyarse de un dibujo o representación, ya que no sería necesario tenerlo que hacer entero desde cero en una pizarra.

Posteriormente llegó la famosa era TIC, la nueva era tecnológica en la que aún vivimos. Aparecieron entonces los primeros ordenadores personales. Y esto a su vez trajo consigo un nuevo método, evolución del anterior y basado en éste: las presentaciones de diapositivas, que hacían uso del programa creado en 1987 para Macintosh, PowerPoint de Microsoft Office [2], que hoy en día todo el mundo conoce.

Aunque en su primera versión solamente permitía la creación de diapositivas en blanco y negro que podían imprimirse para su posterior proyección, a partir de la década de 1990, el programa PowerPoint se impuso progresivamente como el soporte privilegiado en el universo de la docencia profesional en muchas de las universidades más importantes de América, y años posteriores, de Europa.

Los docentes universitarios comprendieron enseguida el interés en el uso de este programa, ya que este método suponía una nueva forma de dar clase que presentaba gran cantidad de ventajas. Por ejemplo, sirve como alternancia de medios que facilitan la atención de los alumnos, facilita la presentación de los contenidos y conceptos de los temas a tratar, permite volver sobre temas tratados y repetir la presentación de ciertos aspectos cuando no son suficientemente asimilados por los alumnos, permiten señalar directamente sobre las transparencias para incidir en ciertos contenidos o atraer la atención, y por último, ahorran tiempo en la exposición. Por todo esto, tuvo una alta y rápida aceptación entre todo el público en general [3][4].

Tal fue la aceptación y el apoyo de este método que, a día de hoy, es muy habitual el uso de recursos informáticos por parte de los docentes de una forma cada vez superior y, como ya se sabe, ha sufrido una grandísima evolución a lo largo de estos últimos años. En concreto, según los últimos estudios, el 90 % de los profesores de universidad son partidarios del uso de PowerPoint en las clases teóricas [1]. Esto se debe gracias, entre otras cosas, a la evolución de las prestaciones de los programas de presentaciones, como en el caso de PowerPoint (véase la Tabla 1.1). Además, en base a estas evoluciones, hoy en día también se pueden encontrar infinidad de diversos tipos de proyectores, con una cantidad de prestaciones que eran impensables hasta hace muy poco tiempo.

En la Tabla 1.1 se puede observar cómo el programa ha ido evolucionando en función de las nuevas necesidades que han ido surgiendo a lo largo del tiempo. Se ve notoriamente el aumento exponencial, en la última década, del aumento de las funcionalidades, quedando sin mencionar además la versión de 2013, donde aún se han seguido aumentando las mismas, y probablemente se seguirá haciendo en versiones futuras.

1. Introducción y referencias bibliográficas

Versión	Año de aparición	Características o mejoras
PowerPoint 1.0	Septiembre de 1987	• Disponible en blanco y negro para la Apple Macintosh y para el sistema operativo DOS de la PC • Tenía sólo las herramientas más básicas de dibujo • Incluía plantillas pre-diseñadas
PowerPoint 2.0	Mayo de 1988	• Incluye gráficos más prácticos • Se desplazó la paleta de 256 colores, por la paleta digital que se extendían a 16,8 millones de colores • Se incluyeron las funciones de corrección gramatical, traer al frente, llevar detrás, buscar y reemplazar. • Se podían importar archivos
PowerPoint 3.0	Octubre de 1992	• Nuevas herramientas de dibujo • Habilidad de rotar objetos • Copiar las características de los objetos • Introducción de 84 tipos de gráficos estadísticos pre-diseñados
PowerPoint 4.0	Septiembre de 1994	• Se introdujo nuevas características para ayudar a los presentadores y conferenciantes a adaptarse mejor al mundo de las presentaciones gráficas
PowerPoint 95	Agosto de 1995	• Mejoras en la personalización de las opciones de menú • Soporte de Visual Basic • Múltiples comandos de deshacer • Animaciones • Una interfaz más amistosa y personalizable
PowerPoint 97	Enero de 1997	• Se incrementaron las animaciones • Se incluyó Custom Shows • Grabar narraciones • Se incluyó un reproductor de media incrustado • Se pueden guardar las presentaciones para internet
PowerPoint 98	Enero de 1998	• Por primera vez, la casi perfecta compatibilidad entre plataformas Mac y Windows • Aparece Microsoft PowerPoint • Admite archivos creados en las versiones anteriores PowerPoint
PowerPoint 2000	Marzo de 1999	• Se incluyeron animaciones y gráficos estadísticos • Formateo de errores • Se mejoró las diferencias de resolución entre computadora y proyector.
PowerPoint XP	Enero de 2002	• Los cambios en las opciones de revisión reducen drásticamente • La vista normal incluye el texto de cada diapositiva o RGP • La vista en miniatura de sus gráficos colocada bajo unos indicadores a la izquierda de la pantalla • Los paneles permiten tener acceso con un solo clic a las plantillas de diseño • Las combinaciones de colores aumentaron • Se mejoran las opciones de animación y las herramientas de revisión
PowerPoint 2003	Un año después de la aparición de la versión XP	• No se diferenció mucho con respecto a la versión XP • Contiene la opción empaquetar para CD • Agrega el visor en CD-ROM para su distribución • Se agrega un soporte mejorado para gráficos y multimedia
PowerPoint 2007	Noviembre del 2006	• Nueva interfaz de usuario • Mejoradas capacidades gráficas • Se incluyó un nuevo formato de archivo basado en XML (.pptx, .pptm, .potx, etc.) • Mejorar la recuperación de archivos dañados • Más control sobre la información personal • Mejorar la integración de los datos profesionales
PowerPoint 2010	Enero del 2010	• Se incluyó nuevas librerías de animación • Tratamiento de imágenes • Organizar las diapositivas en secciones • Trabajar con archivos de presentación de separados en diferentes ventanas • Mejoras e incorporaciones de edición de video e imágenes • Nuevos diseños de imágenes gráficas SmartArt • Transiciones de diapositivas 3D

Tabla 1.1: Evolución de Power Point

Y no sólo hay que pensar en el ámbito docente, sino que el gran auge que ha sufrido este método de presentación se debe también a la gran utilización del mismo por parte de las empresas, en sus continuas reuniones cada vez más frecuentes.

Tal es la necesidad del uso de estas herramientas día a día por parte tanto de alumnos como de profesores, o de las empresas, que en la actualidad Microsoft ya ha lanzado su paquete Office no sólo para PC, sino también para los nuevos *smartphones*, tabletas o incluso ordenadores con otros sistemas operativos diferentes al suyo propio.

Este hecho es bastante indicativo de cómo se encuentra la situación actual en este mercado. Ésto permite hacerse una idea de la constante evolución que ha sufrido en este ámbito el tema de la enseñanza, reuniones de empresas o conferencias. *“Somos una de las unidades más rentables de Microsoft, generamos el 48 % de los beneficios de la venta de programas”*, explicaba Robert Gaskins, uno de los inventores de PowerPoint [6].

1.1.2. Tecnología en las aulas

Como afirma Natalia (2008): *“Las tecnologías de la información y las comunicaciones aportan al mundo educativo un complemento o mejora a las enseñanzas tradicionales proporcionando numerosas nuevas opciones”* [7].

En la nueva era tecnológica es impensable que en una empresa no haya una sala de reuniones con proyector, o que un profesor no pueda dar una clase con diapositivas. Esto se debe a la utilidad y facilidad que proporciona trabajar con ellos. Pero no solo hay que limitarse a esto. El uso de la tecnología en las aulas docentes ha evolucionado mucho en la última década.

El uso del proyector no es el único recurso empleado para la práctica docente, aunque sí el más utilizado. Tiempo atrás se comenzaron diversas campañas para la implantación de ordenadores en las aulas de primaria, y posteriormente en secundaria. Lo que se conoce como *“Escuela TIC 2.0”* [10].

También se empezaron a implantar las primeras pizarras digitales interactivas, que consisten en un ordenador conectado a un vídeo proyector, que muestra la imagen de dicho ordenador. Y desde la superficie de la pizarra es posible controlar el ordenador mediante un bolígrafo especial. De forma que se puede escribir, dibujar, señalar, etcétera.

Aunque este método no ha tenido todo el éxito esperado y que no ha sido altamente aceptado. Según un estudio de la Universidad de Salamanca (2009) la pizarra digital interactiva se convertiría en el recurso docente más utilizado al cabo de pocos años, y un gran número de aulas docentes contarían con

una de estas pizarras [11]. Pero finalmente, parece que no ha sido así.

El caso más reciente sobre la tecnología en las aulas es el reparto de ordenadores portátiles para los alumnos de últimos cursos de primaria. Se ha llegado incluso a repartir tabletas PC según apuntaba Rosa J. en el diario El Mundo el 6 de agosto de 2012. Hecho que da una idea de la época en la que nos encontramos.

Toda esta evolución se debe a que la tecnología cambia continuamente, y se quiere aprovechar este efecto para facilitar en cualquier medida todo el trabajo posible. Por parte de los alumnos resulta más sencillo seguir una clase o una explicación cuando tiene una referencia visual. Y por parte del docente también es más sencillo seguir con la explicación si éste tiene una referencia visual en la apoyarse.

Las TICs son cambiantes, siguen el ritmo de la continua evolución y avances tecnológicos. Y es un hecho que falta aún mucho camino por recorrer en el ámbito de la educación. Como por ejemplo, las aulas de tele enseñanza, que permiten que el profesor esté ubicado en cualquier sitio remoto dotado de videoconferencia. El sistema de audio de ambiente permite participar a cualquier alumno [7].

1.2. Motivación

Llegados a este punto ha quedado claro que el uso de la tecnología, y en especial del proyector, es necesario hoy en día para la docencia y para otras actividades. *“No hay trabajo sin reunión, y no hay reunión sin presentación”* apuntaba Franck (2011) [6]. Pero... ¿cómo se puede mejorar algo que está en continua evolución y que tiene detrás grandes empresas que se encargan de las mejoras?

En este punto se han de tener en cuenta varios aspectos. En el sistema actual de proyección de material docente es condición necesaria que el portátil esté conectado en todo momento al proyector mediante un cable con conector VGA. Ese cable molesto y que no permite al docente disponer de la movilidad que le gustaría para dar su clase. Ya que, es necesario estar cerca del portátil para poder cambiar de diapositiva. Hecho que provoca que la clase pierda dinamismo, y consecuentemente sea mucho más rígida en movimientos que las explicaciones en la pizarra.

En ese sentido, existe la posibilidad de comprar un mando inalámbrico para trabajar con libertad de movimiento. Pero esto supone un gasto para el docente que él no tiene porqué asumir, así como tener que llevarlo siempre consigo.

Por otro lado, un inconveniente que también presenta el método actual

es la necesidad de transportar constantemente el ordenador personal para poder proyectar el material utilizado. Una solución simple a este inconveniente es dejar un PC siempre conectado al proyector y únicamente llevar los archivos en una memoria externa, como un lápiz o una tarjeta de memoria. Pero esta solución supone un gasto grande añadido al presupuesto de la instalación del sistema, además de que es necesario llevar siempre dicho dispositivo de almacenamiento actualizado para cada vez que se quiera utilizar. Y, evidentemente, es más fácil perder u olvidar dicho dispositivo que el ordenador personal.

A todo esto se añade una clara tendencia a la creación de los nuevos dispositivos cada vez de menores dimensiones, como la nueva generación de portátiles *ultrabook*. Lo que está provocando la pérdida o reducción de las posibles conexiones en estos dispositivos. Esto, además del auge en las nuevas tecnologías portátiles como tabletas o móviles, provoca que los nuevos dispositivos tecnológicos no puedan conectarse a los proyectores normales, los habitualmente instalados en las clases, mediante un conector VGA convencional. En su lugar, son necesarios adaptadores a partir de una entrada USB o HDMI, con el inconveniente añadido de tener que comprarlo y tener que llevarlo siempre. Otra opción sería el uso de proyectores con conexión inalámbrica, generalmente Wi-Fi, pero esta es una solución más cara en términos presupuestarios en comparación con un proyector normal. No sería una diferencia excesiva si se tratase de un único proyector, pero sí lo es cuando se trata de un número considerable. Sin ir más lejos, para la propia universidad supondría una inversión económica muy grande cambiar los proyectores de todas sus clases.

Por tanto, es motivante buscar una solución que trate de proporcionar la comodidad que supone no tener que depender de cables y conexiones con el proyector. Algo que, a su vez, facilitaría el movimiento del docente en el aula. Además, no debería ser necesario transportar el portátil, sino que sería interesante la posibilidad de poder utilizar cualquier dispositivo electrónico del momento, tal como una tableta o el teléfono móvil que siempre llevamos encima. Y por último, pero no menos importante, una solución lo más económica posible.

De esta forma, por un lado existe un campo que está en continua evolución y una tecnología que se utiliza diariamente, como es el método de enseñanza actual o la forma de realizar exposiciones. Y por otro lado se presentan los inconvenientes y problemática asociados, además de las posibles mejoras que se han recopilado. De todas estas ideas surge el tema de este proyecto. Un campo que vivimos diariamente y que nos es muy familiar, con el objetivo de implantar una serie de mejoras sobre el sistema actual pensando que en un futuro no muy lejano pueda llegar a ser una solución real para los problemas anteriormente planteados.

De esta manera, la solución final que se plantee podrá implantarse en las aulas docentes de la propia universidad, algo que es una motivación extra muy especial, para cualquier alumno que realice un trabajo fin de grado de estas características.

1.3. Estudio del estado del arte

El objetivo de las tecnologías de la información y comunicación es facilitar nuestra calidad de vida. Cada día se ven avances tecnológicos, estamos totalmente conectados y hemos superado la revolución digital para entrar de lleno en la era tecnología. En la sección 1.1 se ha visto cómo se ha trasladado esto a la educación y los nuevos problemas que se han generado.

En esta sección se hará un estudio de algunas de las soluciones existentes, totales o parciales, sobre el ámbito de la tecnología usada y aplicada a los nuevos métodos docentes.

1.3.1. Proyectoros inalámbricos

Sin duda alguna el proyector es la solución actual más utilizada por parte de cualquier docente, en especial, de una universidad. Es la forma habitual de trabajo hoy en día aun con los inconvenientes que se han visto anteriormente, en especial la movilidad del docente y la necesidad de conexión mediante un cable.

Alguno de estos inconvenientes puede resolverse con los proyectores existentes en estos momentos. En la actualidad existen variedad de tipos de proyectores donde elegir, con diferentes características. En este sentido se pueden diferenciar dos grandes grupos: los proyectores que precisan de conexión física mediante un cable para poder mostrar la imagen, y los que por el contrario, se pueden utilizar de forma inalámbrica. El primer tipo de proyector es el actualmente implantado, y ya se ha comentado la problemática asociada en la sección anterior. El segundo tipo de proyector es el que interesa en este ámbito.

Este tipo de proyectores utilizan tecnología Wi-Fi o *Bluetooth* para la conexión y comunicación con el ordenador proveedor de imágenes. Esta solución es claramente mejor que los proyectores normales ya que, al utilizar las conexiones inalámbricas no se limita la movilidad del docente sino que se puede utilizar y proyectar la imagen de un ordenador desde cualquier parte de la clase.

Por otro lado, también se soluciona el nuevo problema que está surgiendo sobre la pérdida de conexiones, VGA entre otras, en los portátiles actuales de

nueva generación. En este caso solo sería necesaria una conexión USB donde conectar un dispositivo que trabaja de antena para la transmisión de datos entre ordenador y proyector (solo en algunos casos, depende del producto y del fabricante). Conexión que está presente en prácticamente la totalidad de los portátiles.

Tantas son las ventajas que presentan estos proyectores que su evolución y utilización está en presente auge. Las principales empresas del sector como Sony, Panasonic o Mitsubishi ofrecen muchas prestaciones y posibilidades que mejoran el sistema actual utilizado [13][14][15].

Tal está siendo dicha evolución que Sony, una de las principales empresas presentes en el mercado actual, diseñan sus productos en función del sector al que está destinado; escuelas, universidades o empresas entre otros, como así se puede ver en sus propios artículos [16].

Aunque por otra parte, estos proyectores hoy en día son poco comunes. Es raro encontrar un proyector de estas características en un aula normal de universidad. Aunque sí puede ser una solución más habitual para una sala de reuniones de una empresa. Pero por lo general, este tipo de proyectores suelen ser más caros que los proyectores normales. El principal problema de esta solución puesto que las alternativas más económicas sitúan el valor mínimo de un proyector de estas características en los 1000€ según las principales página de ventas.

Por tanto, la implementación de esta solución supondría un gasto muy elevado en la adquisición de dichos proyectores. Además de tener que sustituir los proyectores presentes en las aulas de la universidad actualmente. Algo que no resulta necesario puesto que se puede encontrar una solución más económica a ésta y donde se puedan reutilizar los proyectores actualmente instalados.

En este punto se podría utilizar un adaptador mediante el cual se permita conexión inalámbrica al proyector. Uno de los ejemplos es el dispositivo ofrecido por Ipevo [17]. Pero dicho dispositivo sigue sin ser lo suficiente económico, por lo que el problema de esta solución, aunque menor que los proyectores con Wi-Fi, sigue siendo el precio.

1.3.2. Pizarra interactiva y *Tablet Monitor*

Se ha visto en la sección 1.1.2 en qué consistía y cuál era la principal función de la pizarra digital interactiva. Por lo general, es un accesorio que se suele adherir a una superficie rígida para convertirla en una pizarra interactiva. Así, la única diferencia entre esto y el caso anterior es la interactividad con la pizarra, pero esto no soluciona los problemas mencionados. La ventaja ofrecida no es del interés de este estudio.

Para encontrar ventajas que puedan ser de utilidad para el caso que concierne es necesario un segundo elemento, una tableta monitor. Este dispositivo se conecta al ordenador sin cables (por ejemplo a través de *Bluetooth*) y en algunos casos hasta permite que varios usuarios actúen simultáneamente en trabajos en equipo, lo que permitiría interacción entre alumnos y profesor.

El término *Tablet Monitor* identifica al periférico desde el que se realiza el control del ordenador y las anotaciones manuscritas. Es un monitor especial (combinación de monitor y tableta) diferente a una tableta convencional, una *Tablet-PC*. En este caso el ordenador se conecta a un proyector y la imagen de la pantalla es proyectada en la pizarra y visualizada en la tableta monitor [19].

Este dispositivo tiene la ventaja de que se puede trasladar a cualquier lugar, uno de los principales inconvenientes mencionados. Y no se limita sólo al aula, sino que se pueden realizar clases y conferencias a distancia en tiempo real a través de Internet, permitiéndose la conexión entre la tableta monitor y el ordenador conectado al proyector.

Se presentan múltiples ventajas en el trabajo cooperativo entre las pizarras interactivas y las tablets monitor. Esta solución mejoraría el problema de la movilidad. Pero presenta un grave problema esta solución. Al igual que el caso anterior, el precio de este producto es muy elevado. Una pizarra interactiva supera los 1000€ según la web informativa <http://www.pizarrasinteractivas.com/> [20], siendo éstas las de menor precio y por tanto las de menores prestaciones. A lo que hay que añadirle el precio de una tableta monitor para cada profesor que vaya a impartir clase con este método docente como mínimo.

Sumando todo el gasto supone un muy elevado inconveniente. Además de ser totalmente innecesario para una entidad que necesite instalar muchos sistemas de este tipo, por ejemplo la universidad. La inversión sería de una pizarra interactiva por cada clase, y una tableta monitor por cada profesor. Y la mejora introducida no deja de ser únicamente la movilidad del docente. Por lo que se debe descartar esta opción como posible solución final. Es una única mejora para una elevada inversión presupuestaria.

1.3.3. Apple TV y *AirPlay*

Apple TV [21] es un receptor digital multimedia diseñado, fabricado y distribuido por Apple [22]. El dispositivo está diseñado para reproducir contenido multimedia digital desde iTunes Store, YouTube e iCloud entre otros, en una televisión de alta definición. Pero Apple TV no sólo se limita al contenido almacenado en estos sitios, sino que puede utilizarse el contenido

multimedia de un ordenador Mac OS X o Windows con iTunes.

Éste es un dispositivo diseñado principalmente para el ocio que puede ser utilizado en casa para uso personal del contenido multimedia. Pero Apple TV no solo se limita al contenido multimedia presente en la red. Gracias a la tecnología *AirPlay* [24] de Apple, se puede ver todo el contenido almacenado en un Mac o cualquier dispositivo iOS. De forma que puede ser utilizado para proyectar imágenes, vídeos, o presentaciones digitales, que es lo importante en este caso de estudio y además es posible hacerlo desde portátil, tableta o *smartphone*.

Apple TV es pequeño y compacto, potente por la gran cantidad de posibilidades que ofrece y de fácil configuración para cualquier persona. Algo que encajaría en cualquier sitio, y que puede ser usado en términos profesionales, como es el caso que nos concierne, la docencia. En este sentido, se puede colocar este dispositivo conectado a un proyector mediante el adaptador HDMI, y proyectar el material que se desee desde cualquier dispositivo.

De hecho, ya se ha realizado un estudio del uso de este dispositivo en este ámbito por la Universidad Rovira i Virgili (URV), *Aumentando la interacción en el aula ordinaria mediante el iPad y demás dispositivos de m-learning* (2014) [25]. En él se ha empezado a utilizar dicha tecnología junto con iPads de forma que todo el mundo puede ver lo que se está viendo en un iPad concreto, lo que ayuda a seguir la clase sin distracciones. La arquitectura básica de funcionamiento que se ha implantado puede verse en la siguiente imagen (Figura 1.1).

Esta arquitectura es la propuesta por GoClass [26], una plataforma de enseñanza que se basa en el uso de dispositivos móviles comunes para realizar cualquier tipo de tarea. De esta forma se permite conexión total entre alumnos y profesor, tanto fuera como dentro de clase.

GoClass propone un nuevo tipo de enseñanza basado en la nube y los nuevos dispositivos portátiles. De forma que se puedan utilizar dispositivos móviles, PC y pantallas de proyección para conectar a los estudiantes unos a otros, así como al profesor y consecuentemente lo que este está proyectando. Aunque este método aún no es muy extendido tiene posibilidades para un futuro.

Esta solución al contrario que soluciones anteriores, no solo mejoraría la movilidad del docente, sino que gracias a la tecnología *AirPlay* se produce la uniformidad de las tecnologías, pudiendo utilizar para la presentación el portátil, tableta o móvil. Pero esto no son todo ventajas, también tiene un inconveniente bastante importante. Es absolutamente necesario que el dispositivo tenga instalado un sistema operativo iOS para poder hacer uso de la tecnología *AirPlay*. Dispositivos que únicamente son distribuidos por la misma compañía que el dispositivo principal, Apple.



Figura 1.1: Estructura de funcionamiento de sesión multimedia estándar con iPad [26]

Esto quiere decir que los profesores tendrán que tener portátil, tableta o móvil de la compañía Apple, algo que no es del agrado de todos ellos. Hecho que descarta totalmente esta solución, aunque el dispositivo presente un precio asequible (según la página oficial 112€)[21].

La solución finalmente adoptada no debe suponer ningún gasto o inconveniente adicional para ningún docente, puesto que se trata de mejorar lo actual sin que esto conlleve un cambio importante en él.

1.3.4. Wireless Display y Miracast

Como alternativa a la tecnología propuesta por Apple (Apartado 1.3.3), se puede encontrar *Wireless Display* [28]. Un dispositivo cuya función es básicamente la misma que Apple TV. *Wireless Display* trata de compartir contenido desde cualquier teléfono inteligente, tableta o PC hasta una televisión HD sin necesidad de cables.

Sus características y funciones son prácticamente iguales a las del caso anterior, pero hay una diferencia más que notable. Apple TV corresponde a un único dispositivo y que funciona únicamente con otros dispositivos Apple. En cambio existen gran cantidad de dispositivos pueden ser utilizados en este caso. Lo que elimina la necesidad de tener en posesión un dispositivo Apple.

Estos dispositivos utilizan la tecnología Miracast [30]. Miracast es un estándar creado por la Alianza Wi-Fi y que consiste fundamentalmente en una red *peer-to-peer* utilizada para enviar *screencast* de forma inalámbrica formando conexiones *Wi-Fi Direct* de forma similar a como lo hace *Bluetooth*. De este modo, permite la distribución inalámbrica de la imagen desde o hacia ordenadores, tabletas, teléfonos móviles y otros dispositivos conectados. Tanto el dispositivo emisor como el receptor deben soportar la tecnología Miracast para funcionar, uno de los principales inconvenientes de la actualidad.

Según se publicó en algunas páginas de noticias relacionadas con este ámbito todos los dispositivos con Android 4.2 o superior tienen soporte para Miracast, aunque también explican que el rendimiento en la práctica no es tan bueno como el que teóricamente se espera.

Según se ha podido leer en <http://www.miracast.co.uk/> Miracast no ha triunfado como se le esperaba. Pocos fabricantes han adoptado el estándar, y los que lo han hecho no funcionan lo bien que se desearía. Un serio inconveniente de compatibilidad que dificulta el estudio de este sistema como posible solución.

1.3.5. Chromecast

Dentro este mismo ámbito cabe destacar *Chromecast* [32], que es el principal competidor de Apple TV, por nombre y sobre todo por precio, en torno al 70 % más económico que el anterior (35€ según la página web oficial). En cuanto a las características, parecido a los dispositivos comentados anteriormente.

Aunque existen varias diferencias que hacen de este dispositivo una mejor solución a las anteriores. En primer lugar es compatible con todas las versiones de los sistemas operativos más utilizados del momento a partir de Android 2.3, iOS 6.0, Windows 7, Mac OS 10.7 y por supuesto Chrome OS.

Pero este dispositivo, al igual que el resto de los anteriores, también presenta varios inconvenientes. Éstos son unos dispositivos relativamente recientes, y con un margen de crecimiento y mejora aún por concretar, por lo que lo ideal sería esperar antes de adquirirlos. Para una persona normal no supone un gran gasto adquirir uno para ocio personal, pero para la universidad supone una inversión grande, por lo que se ha realizado un estudio en profundidad antes de realizar la compra.

Además, al contrario que los anteriores, no permite *screen mirroring* por lo que no se puede ver la imagen de la pantalla de nuestro dispositivo. Simplemente se puede seleccionar el contenido y reproducirlo. Llegando así a otro de sus inconvenientes, es necesario tener el contenido almacenado en el dispositivo para poder reproducirlo. Algo que intenta solucionar este proyecto.

1.3.6. Resumen

Finalmente, la Tabla 1.2 muestra un esquema-resumen de todos los productos o métodos actuales vistos en este capítulo como posibles soluciones al problema planteado en el capítulo anterior.

Como se puede ver, en todas las soluciones propuestas en la actualidad, salvo Apple TV, es necesario tener el documento a mostrar guardado en el propio dispositivo, o descargarlo justo antes de mostrarlo. La solución que se propone en este proyecto pretende eliminar este inconveniente para que en todo momento el docente pueda utilizar el dispositivo que mejor le convenga a la hora de dar clase; ordenador, tableta o móvil, sin necesidad de llevar la presentación preparada y guardada en dicho dispositivo.

Además se ve en la tabla que el inconveniente que más veces aparece es el precio del dispositivo o de la instalación. La solución final debe tener un precio total suficientemente bajo como para que se convierta en una solución real por la que puedan optar las empresas o la propia universidad.

<i>Solución</i>	<i>Ventajas</i>	<i>Inconvenientes</i>
<i>Proyectores inalámbricos</i>	▪ Movilidad ▪ Conexión de diferentes dispositivos	▪ Precio ▪ Desperdicio de los proyectores actuales
<i>Pizarra interactiva</i>	▪ Movilidad ▪ Interacción con la pizarra	▪ Precio por unidad ▪ Necesidad de una <i>Tablet Monitor</i> por docente
<i>Apple TV</i>	▪ Movilidad ▪ Conexión de diferentes dispositivos ▪ Almacenamiento centralizado (iCloud)	▪ Precio ▪ Necesidad de un dispositivo iOS
<i>Dispositivos Miracast</i>	▪ Movilidad ▪ Económico ▪ Conexión de diferentes dispositivos ▪ Screen mirroring	▪ Necesidad de un dispositivo compatible con <i>Miracast</i> ▪ Aún en desarrollo
<i>Chromecast</i>	▪ Movilidad ▪ Solución más barata ▪ Compatibilidad con la mayoría de S.O. actuales	▪ Sin screen mirroring ▪ Aún en desarrollo ▪ Sin almacenamiento centralizado

Tabla 1.2: Comparación de las soluciones existentes

1.4. Objetivos y alcance del proyecto

En este apartado se exponen brevemente, a modo de introducción, los principales objetivos del proyecto. Detallándose posteriormente en profundidad los mismos en el Capítulo 3.

El principal objetivo del presente proyecto consiste en solventar los problemas descritos anteriormente. Para ello la solución final debe facilitar la movilidad del docente por toda el aula de enseñanza. Además deberá ser fácilmente portable de forma que en un futuro se permita el uso de diferentes tipos de dispositivos electrónicos para proyectar e interactuar con el material docente. Además, se busca centralizar el almacenamiento de los materiales didácticos para así evitar su transporte al aula.

Se realizará en este sentido un servicio en el que cada usuario tenga una cuenta donde poder guardar los archivos que pretende utilizar. Y posteriormente, con esa misma cuenta poder mostrarlos en el proyector que él mismo decida, mediante el uso de una plataforma web. Dicha plataforma es soportada por todo tipo de dispositivos, lo que consigue unificar esta tecnología para que cualquier dispositivo de los anteriormente mencionados pueda ser utilizado de forma indistinta por el docente para proyectar su material.

Para el cumplimiento de estos objetivos, se especificarán una serie de requisitos (véase la sección 3.2) que la solución final debe cumplir. En este ámbito se podrán distinguir requisitos funcionales, y no funcionales como seguridad, rendimiento y usabilidad entre otros.

Este proyecto se ha pensado y realizado con vista a un futuro cercano, donde se pueda llevar a cabo el montaje del producto final, de forma rápida y sencilla, por parte de cualquier persona o entidad que lo desee. Además se dejará la puerta abierta para las futuras mejoras que se puedan implantar.

1.5. Estructura de la memoria

La presente memoria del proyecto está formada por ocho capítulos. A continuación se realiza una breve descripción de cada uno de ellos:

- **Capítulo 1. Introducción y referencias bibliográficas.** Este capítulo establece el contexto donde se sitúa el presente proyecto, trata los motivos fundamentales que han suscitado la creación del mismo, se muestran las soluciones que existen actualmente al problema y se exponen los principales objetivos a cumplir por éste.
- **Capítulo 2. Planificación y estimación de costes.** Se describe el presupuesto del proyecto y la evolución temporal estimada para su

realización.

- **Capítulo 3. Análisis de objetivos y metodología.** Se describen los objetos que el proyecto quiere alcanzar y requisitos tanto funcionales como no funcionales que debe cumplir el producto resultante.
- **Capítulo 4. Diseño y resolución del trabajo.** Se describe el diseño realizado para la solución del problema planteado. Esto incluye tanto el diseño general, como el diseño de los distintos elementos que constituyen el producto. Además, se describen los pasos seguidos y cómo ha sido la realización la implementación del anterior diseño.
- **Capítulo 5. Evaluación y pruebas.** Se definen los posibles entornos de utilización del producto y se evalúa el resultado en cada uno de ellos.
- **Capítulo 6. Conclusiones y vías futuras.** Se muestran las conclusiones obtenidas en función de los resultados anteriores y se expone una posible línea de mejora para posibles trabajos futuros.

Posteriormente se encuentran anexos a este documento un manual de usuario con las funciones básicas del producto y la forma correcta de utilización, así como un manual de instalación para quien quiera desplegar la solución propuesta. Y finalmente se muestran todas las referencias bibliográficas que se han consultado para realizar este proyecto.

1.6. Principales referencias utilizadas

En este apartado se exponen las principales fuentes consultadas para la realización de este proyecto, mostrándose la totalidad de ellas en la bibliografía.

- E. Upton and G. Halfacree, Raspberry Pi. Madrid: Anaya Multimedia, 2014.
- G. Carter, *LDAP System Administration*. Sebastopol, CA: O'Reilly, 2003.
- Y. Fernández Hansen, A. A. Ramos Varón and J. García Morán, *AAA - RADIUS - 802.1x: Sistemas Basados En La Autenticación En Windows y Linux-GNU: Seguridad Máxima*. Madrid: Ra-Ma, 2008.
- S. McCracken, “Curso de programación Web con HTML5, CSS, JavaScript, PHP 5.6 y MySQL” pp. 1148, 2011.

1. Introducción y referencias bibliográficas

- Manual oficial de **PHP** disponible en: <http://www.php.net/manual/es/index.php>
- Documentación oficial de **Java** disponible en: <http://docs.oracle.com/javase/7/docs/api/>

Capítulo 2

Planificación y estimación de costes

2.1. Recursos

2.1.1. Humanos

- D. Jorge Navarro Ortiz, profesor del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada, en calidad de tutor del proyecto.
- Juan Ramón Gutiérrez Martínez, alumno de la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación.

2.1.2. Hardware

- Ordenador personal con:
 - Servidor Web (Apache) [36]
 - Servidor FTP (VSFTP) [37]
 - Servidor RADIUS (FreeRADIUS) [38]
 - Base de datos (LDAP) [39]
- Raspberry Pi [40]
- Proyector con entrada HDMI

2.2. Fases de desarrollo

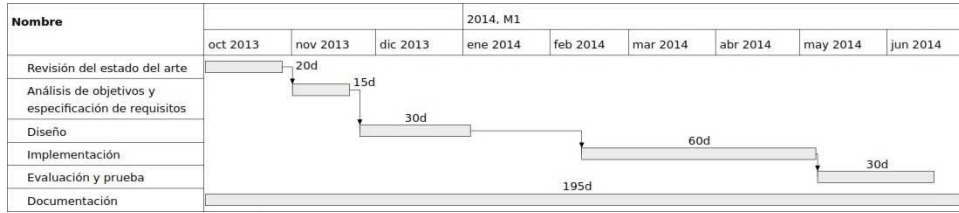


Figura 2.1: Diagrama de Gantt con la planificación temporal

Concepto	Comienzo	Fin	Duración
<i>Revisión del estado del arte</i>	01/10/13	28/10/13	20
<i>Análisis de objetivos y requisitos</i>	01/11/13	21/11/13	15
<i>Diseño</i>	25/11/13	03/01/14	30
<i>Implementación</i>	11/02/14	05/05/14	60
<i>Evaluación y pruebas</i>	06/05/14	16/06/14	30
<i>Documentación</i>	01/10/13	30/06/14	195

Tabla 2.1: Planificación temporal del proyecto

2.1.3. Software

- Sistema operativo GNU/Linux Ubuntu 14.04 para el ordenador personal [41]
- Sistema operativo GNU/Linux Raspbian Debian Wheezy para la Raspberry Pi [42]
- Java versión 8.5 [43]
- Lenguaje de programación PHP versión 5.5.13 [44]
- Entorno de desarrollo Netbeans 8.0 [45]
- Visor de archivos PDF MuPDF [46]

2.2. Fases de desarrollo

La distribución temporal de las distintas fases del proyecto se puede ver en el diagrama de Gantt de la Figura 2.1 y en la Tabla 2.1.

2.2.1. Revisión del estado del arte

Se hace un repaso de las soluciones existentes hasta el momento, totales o parciales, sobre el mismo ámbito de aplicación del problema a resolver con

el objetivo de sentar una base que justifique la realización de este proyecto.

2.2.2. Especificación de los requisitos

Se analiza el problema en pos de una correcta especificación de requisitos, funcionalidades y restricciones que delimitan el posterior diseño de una solución en función de unas especificaciones y medidas de seguridad de una solución.

2.2.3. Diseño

Se ha de adoptar justificadamente una solución previamente especificada y se procede a su diseño atendiendo en todo momento a los requisitos previamente definidos sobre prestaciones y seguridad.

2.2.4. Implementación

Se ha de implementar el diseño anterior atendiendo a los lenguajes de programación y características escogidas. Realizando un rediseño de alguna característica una vez comenzada la implementación si fuese oportuno.

2.2.5. Evaluación y pruebas

Tras la comprobación y puesta a punto, se procede a la evaluación de la solución, contrastando su funcionamiento y resultados, si procede, con otras soluciones vistas durante la revisión del estado del arte.

2.2.6. Documentación

La redacción de la documentación es una tarea que se ha de llevar a cabo de forma paralela al resto del proyecto.

2.3. Estimación de costos

La duración total estimada para la realización del proyecto es de 160 días aproximadamente, sí en la planificación anterior (Figura 2.1) no se tienen en cuenta fines de semana ni días festivos, así como tampoco se contará el periodo de exámenes. Una jornada laboral normal dura 8 horas, pero teniendo en cuenta que el proyecto no es lo único en lo que se ha trabajado se reducirá a

Concepto	Coste/Tiempo	Cantidad	Total
Trabajo individual	15 €/hora	640 horas	9.600 €
Tutorías	40 €/hora	15 horas	600 €
Total:			10.200 €

Tabla 2.2: Coste monetario personal del proyecto

la mitad el número de horas de una jornada para un cálculo más realista. Lo que hace un total de 640 horas de trabajo.

Con respecto al tutor del proyecto, se estiman unas 10 tutorías de aproximadamente 1 hora y media de duración cada una. Por tanto, el número total de horas consideradas para el completo desarrollo del proyecto asciende a un total de: 655 horas.

El precio de cada hora del proyecto se ha calcula de acuerdo a los honorarios de un Ingeniero Técnico de Telecomunicación. Según la Junta General del Colegio Oficial de Ingenieros de Telecomunicación (COIT) [47] los honorarios de un Ingeniero de Telecomunicación, son libres y responden al libre acuerdo entre el profesional y el cliente. Dada esta situación, los honorarios deben definirse en función de una serie de factores: costes del ingeniero, desplazamientos, volumen de la actividad, etcétera. Teniendo en cuenta estos elementos se han definidos unos honorarios de 15 €/hora. Y con respecto a los honorarios del tutor (Dr. Ingeniero de Telecomunicación), se han fijado a 40€/hora.

2.3.1. Recursos humanos

En la Figura 2.1 y Tabla 2.1 puede verse el coste temporal de las diferentes partes del trabajo. En la Tabla 2.2 se detalla el coste asociado a los recursos humanos.

Por lo que los gastos de personal con el I.V.A. son 10.200 €.

Y con estos datos de la Tabla 2.1 y Tabla 2.2, se obtiene la Figura 2.2, que refleja el coste monetario asociado a cada una de las fases del proyecto, siendo el coste asociado restante el equivalente a la finalización de la documentación y las horas de tutorías.

2.3.2. Herramientas

Para la implementación se han empleado herramientas software gratuitas siempre que ha sido posible para minimizar el coste del proyecto. En la

2. Planificación y estimación de costes

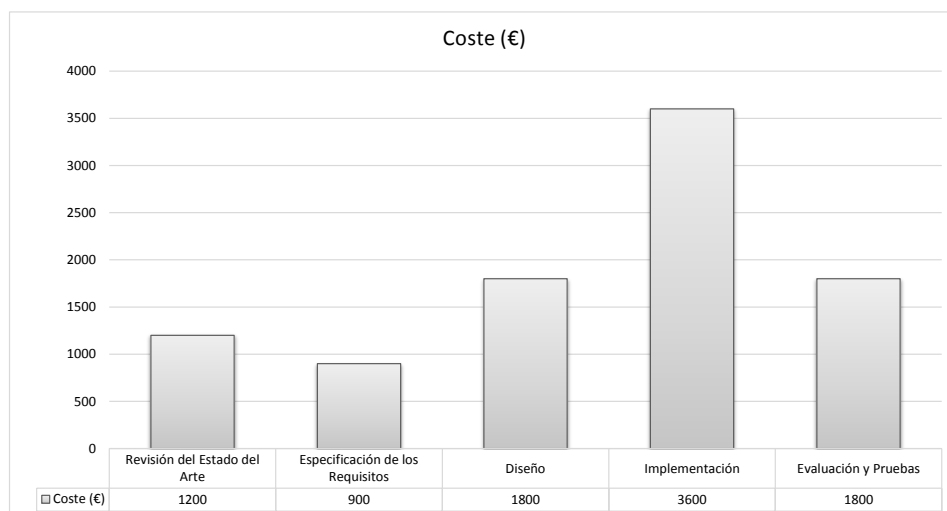


Figura 2.2: Coste monetario asociado a cada fase del proyecto

Tabla 2.3 se recoge el coste monetario de los recursos software y hardware en proporción al tiempo empleado en la realización del proyecto (9 meses).

Aclarar que el monitor ha sido empleado en las fases de implementación y evaluación para facilitar el trabajo.

2.4. Presupuesto

Por último, la Tabla 2.4 resume el presupuesto total del proyecto.

2.4. Presupuesto

Recurso	Coste unitario	Cantidad	Total
Ordenador personal	800 €	36 meses	200 €
Raspbery Pi	50 €	36 meses	12,5 €
Proyector	200 €	120 meses	15 €
Monitor con entrada HDMI	150 €	96 meses	14 €
Conexión ADSL	40 €/mes	-	360 €
Sistemas operativos	0 €	-	0 €
Entorno de desarrollo (Netbeans)	0 €	-	0 €
Software complementarios	0 €	-	0 €
Paquetes de ofimática	0 €	-	0 €
Total:			601,5 €

Tabla 2.3: Coste monetario material del proyecto

Concepto	Cantidad
Recursos humanos	
1 desarrollador x 160 días x 5 horas/día x 15 €/hora	9.600 €
1 tutor x 10 días x 1,5 horas/día x 40 €/hora	600 €
Recursos materiales	
Raspbery Pi	12,5 €
Proyector	15 €
Monitor con entrada HDMI	14 €
Conexión ADSL	360 €
Sistemas operativos	0 €
Entorno de desarrollo (Netbeans)	0 €
Software complementarios	0 €
Paquetes de ofimática	0 €
Total:	10.801,5 €

Tabla 2.4: Presupuesto total

Capítulo 3

Análisis de objetivos y metodología

En la sección 1.4 se describieron brevemente los objetivos y el alcance del proyecto. En este capítulo se hará un análisis con mayor profundidad de los objetivos del proyecto mediante un análisis de los requisitos generales que se precisan. En primer lugar se describe un análisis general de la situación y se exponen los principales objetivos que se han previsto al inicio de este proyecto. En segundo lugar se desarrollan en mayor detalle los principales requisitos que se consideran imprescindibles para el desarrollo del proyecto, de forma que se cumplan los objetivos anteriormente descritos. Y finalmente se realiza un estudio de la metodología seguida para la realización del proyecto y una valoración final sobre los objetivos alcanzados.

3.1. Descripción global

La especificación de objetivos y requisitos es uno de los temas importantes que se han de tratar desde el comienzo de un proyecto. Un punto fundamental de este proyecto es que el producto salga finalmente al mercado con relativo éxito, y para ello ha de cumplir una serie de requisitos específicos. En este apartado se hará una descripción general de la situación.

Perspectiva

El objetivo final del proyecto es la creación de un sistema de proyección de material docente basado en Raspberry Pi. Así, el esquema resultante consiste en una aplicación cliente desde la cual el docente pueda manejar de forma transparente dicho material. Para ello se utilizará un

servidor instalado en una Raspberry Pi conectada al proyector mediante la interfaz HDMI, de forma que finalmente se permita la movilidad del docente por el aula, así como la utilización de cualquier dispositivo para dicho fin.

Raspberry Pi

Raspberry Pi [40] es un ordenador de placa reducida de muy bajo costo, a un precio muy asequible [49], desarrollado por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

El diseño incluye un potente procesador central 700 MHz y 512 MB de memoria RAM. El diseño no incluye un disco duro ni unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente; tampoco incluye fuente de alimentación ni carcasa. Además tiene unas dimensiones suficientemente pequeñas como para colocar en cualquier lugar y transportar sin ningún tipo de problema [50].

Este sistema empotrado o embebido puede soportar diferentes sistemas operativos y lenguajes de programación. Esto convierte a Raspberry Pi en una herramienta muy versátil que aporta grandes ventajas al diseño e implementación de este proyecto.

Principales objetivos previstos

Como se ha venido describiendo, la solución finalmente implementada debe solucionar todos o la gran mayoría de los problemas planteados en el Capítulo 1 sobre el actual método docente utilizado en las universidades.

De esta forma se pueden diferenciar una serie de objetivos que ha de cumplir el proyecto para que la solución implementada sea la idónea:

- Se deben eliminar las conexiones mediante cable. De forma que la conexión inalámbrica permita la movilidad del docente por toda el aula de enseñanza mientras éste imparte su clase.
- Debe ser fácilmente portable a diferentes tecnologías y sistemas operativos de forma que en un futuro se pueda utilizar en diversos entornos diferentes.
- Debe existir un sistema de gestión del material centralizado de forma que no sea necesario llevar el material guardado en los dispositivos para que éste sea proyectado.

3. Análisis de objetivos y metodología

- Debe ser una solución de fácil instalación y transporte, de forma que el sistema pueda ser instalado en una clase y se pueda cambiar a otra de forma sencilla.
- Por supuesto debe ser muy económico, tanto como para que pueda ser instalado sin asumir un gasto considerable.

Como se puede intuir, simplemente gracias a utilizar una Raspberry Pi como base de nuestro sistema se van a poder cumplir la mayoría de estos objetivos.

Objetivos secundarios previstos

Estos objetivos secundarios corresponden a objetivos que no son obligatorios para la realización del proyecto, pero que sí añaden características interesantes a la funcionalidad final. Estos objetivos, al no ser obligatorios, sólo se intentarán cumplir al terminar los objetivos principales y si el tiempo lo permite. Son los siguientes:

- El material docente debe ser en formato PDF como mínimo. Así, es un objetivo secundario ampliar dicha funcionalidad posteriormente a otros formatos de imágenes o documentos.
- El sistema de seguridad y autenticación de usuarios se debe adaptar al sistema utilizado por la Universidad de Granada. De esta forma se facilitará la tarea de la posterior implantación en las aulas.
- El sistema inicial cuenta con una única Raspberry Pi. Se debe incorporar una base de datos adicional a la base de datos de los usuarios de forma que ésta contenga la información de las localizaciones de los proyectores. Así se podrá elegir en qué aula mostrar el material cuando existan varios sistemas instalados.
- Debe unificar las tecnologías actuales, de forma que sea indiferente el uso de ordenador, tableta o *smartphone* para proyectar el material.
- Debe incorporar un sistema que permita mostrar y visualizar en el proyector la imagen de la pantalla del dispositivo utilizado.

Estos objetivos son de gran complejidad y suponen una inversión de tiempo tal que no se puede asegurar su cumplimiento, de ahí que sean objetivos secundarios.

3.2. Especificación de requisitos

Esta sección describe los requisitos equivalentes a la implementación con el nivel de detalle suficiente que permita diseñar un producto final que cumpla con los objetivos previstos de la sección anterior. La sección está dividida a su vez en tres subsecciones, atendiendo a los requisitos funcionales, no funcionales y de las interfaces de usuario.

3.2.1. Requisitos funcionales

Para la definición de los requisitos funcionales, se ha optado por un diagrama de flujo de datos sencillo de nivel superior.

El diagrama de nivel superior define un nivel de abstracción más bajo que las relaciones del sistema con el exterior. Para este caso, esto se traduce en las entradas y salidas del mismo. Como se aprecia en la Figura 3.1, se requieren tres entradas –los datos de usuario, el archivo que se quiere mostrar y la localización del proyector (dirección IP de la Raspberry en concreto)– y ofrece una única salida: muestra dicho archivo por el proyector seleccionado.

El cliente necesita las tres entradas para poder establecer la conexión inicial con el servidor. Una vez que la conexión está establecida y se muestra el material, simplemente se mandan peticiones del tipo cliente-servidor donde cada petición corresponde con un acción que se pueda realizar sobre el material mostrado (e.g. avanzar a la siguiente página).

El servidor estará constantemente escuchando peticiones del cliente, y además, sólo será capaz de servir a un cliente cada vez. Sólo se puede mostrar un documento en el proyector.

3.2.2. Requisitos no funcionales

En este proyecto, quizá los requisitos no funcionales son los más restrictivos y los más importantes a cumplir de cara al usuario final del producto.

Software y compatibilidad

En la implementación se utilizarán los lenguajes de programación Java y PHP, de modo que el código sea de fácil comprensión para facilitar la incorporación de nuevas mejoras en un futuro. Además, de esta forma se consigue que sea fácilmente portable a diferentes dispositivos y sistemas operativos para conseguir finalmente que se pueda utilizar en cualquier dispositivo en un futuro.

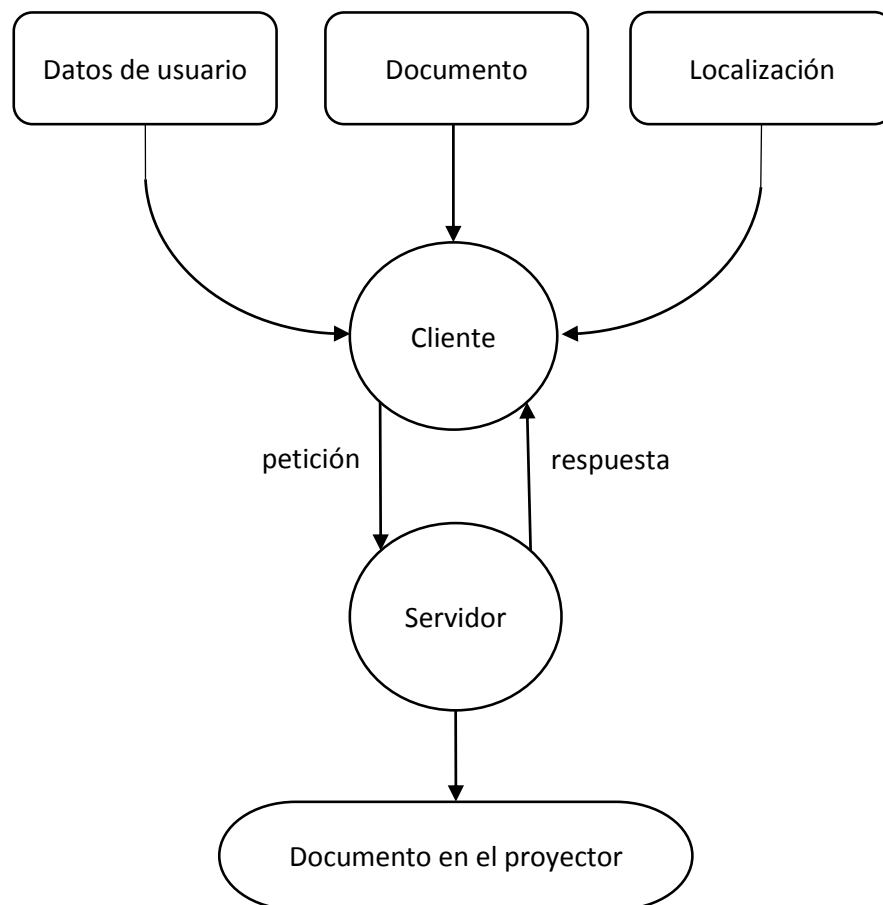


Figura 3.1: Diagrama de flujo de datos de alto nivel

Seguridad

Uno de los requisitos más importantes. Cada usuario debe tener su cuenta propia, de forma que solo él pueda utilizar su material docente y nadie más pueda verlo. Así, de esta forma, toda comunicación que se produzca debe ir cifrada.

Rendimiento

La implementación final no puede ser lenta en cuanto a velocidad de reacción frente a las órdenes mandadas por el docente, es decir, debe tener un buen rendimiento.

Disponibilidad y operatividad

El producto final debe tener un alto porcentaje de disponibilidad para que el docente no tenga ningún imprevisto a la hora de realizar su trabajo.

Gestión de fallos y estabilidad

El sistema final debe ser estable frente a cualquier suceso. Tendrá un sistema de control y recuperación frente a fallos, de forma que no afecte al comportamiento de cara al usuario final. Se incorporará además un completo registro de los mensajes de fallos, advertencias, o información que se produzcan.

Interoperabilidad

No sólo debe existir interoperabilidad a la hora de usar uno u otro dispositivo por parte del docente, sino que además debe de ser indiferente el tipo de proyector utilizado. Todas las combinaciones posibles de dispositivo y proyector deben funcionar correctamente.

Portabilidad

Según el requisito anterior, se podría cambiar el producto de una clase a otra y debe funcionar perfectamente. Por tanto, se ha de realizar de forma que sea portable y de fácil transporte (pequeñas dimensiones, poco peso, etcétera).

Interfaz y usabilidad

3. Análisis de objetivos y metodología

Se independizará la interfaz de usuario del resto de la implementación de forma que el usuario no necesite saber nada del funcionamiento del producto para poder usarlo.

Costo

El producto final debe tener un costo de implementación bajo, de forma que el precio final de cara al público sea muy pequeño en comparación a las funcionalidades que ofrece.

Extensibilidad

El producto final quedará con código abierto y perfectamente documentado para que cualquier persona que quiera pueda seguir mejorando y utilizando el producto, siempre y cuando sea para no comercial, y se pida antes el permiso de las autoridades pertinentes.

3.2.3. Requisitos de la interfaz de usuario

En este apartado se describen las interfaces diseñadas para que el usuario utilice la aplicación. Se pueden distinguir dos interfaces, la correspondiente al servidor colocado en la Raspberry Pi y la interfaz del cliente. En este sentido la interfaz del servidor se minimiza considerablemente puesto que el usuario solo debe lanzar el servidor mediante la línea de comandos y este queda activo hasta que el usuario vuelve a apagarlo. Por tanto, la interfaz se reduce a la línea de comandos donde se muestran los principales mensajes de los principales sucesos.

En lo que respecta a la interfaz que el usuario final utilizará debe de ser:

- Intuitiva: que el usuario sepa cómo y dónde realizar cada acción.
- Fácil de usar: que cualquier usuario que no conozca el producto aprenda de forma rápida.
- Recuperación frente a errores: que la interfaz esté siempre operativa.

En este aspecto, se han de diferenciar entre otras dos interfaces en la aplicación correspondiente al cliente. Entre ellas se diferencia una correspondiente al entorno web desde donde se administrará la cuenta de usuario, así como el material docente. Y otra interfaz que es la correspondiente al manejador de ficheros, es decir, la interfaz utilizada para mandar las órdenes utilizadas para controlar el documento que se está mostrando en el servidor de la Raspberry Pi. Aunque esta interfaz estará integrada también en el

entorno web, para facilitar la tarea y aumentar así la compatibilidad de los dispositivos.

Interfaz Web

Esta interfaz corresponde con una sencilla e intuitiva página web, donde cada usuario puede ver sus archivos guardados en el servidor de ficheros central, así como borrar, descargar, o guardar nuevos archivos entre otras acciones.

Esta interfaz no es uno de los puntos fuertes del producto, aunque se podrá ir mejorando con el tiempo, por lo que su desarrollo e implementación no debe llevar mucho tiempo.

Además desde dicha interfaz se debe de poder comenzar la conexión con el servidor situado en la Raspberry Pi seleccionada (según la clase en la que se encuentre), de forma que se abra el documento, se muestre por el proyector y se abra en el entorno web la interfaz de la aplicación.

Interfaz de la aplicación

En este ámbito, se denominará aplicación a la ventana que se utiliza para mandar las órdenes al servidor que se encuentra en la Raspberry Pi, e.g. avanzar una página. De esta forma dicha interfaz permite controlar lo mostrado por el proyector.

Esta interfaz tendrá la forma de una ventana como cualquier otro programa, con diferentes pestañas para organizar de forma intuitiva las órdenes según diferentes grupos. Los grupos estarán constituidos en función de lo que realicen (buscar, avanzar a una página determinada, cerrar, etcétera). Se incluyen además dos botones principales para avanzar y retroceder las páginas una a una, ya que esta es la acción que más veces se utilizará por parte del usuario final.

3.3. Metodología

En este apartado se mostrarán los diferentes pasos o etapas que se han seguido para la realización de este proyecto.

En primer lugar se hace un repaso de las soluciones existentes hasta el momento, totales o parciales, sobre el mismo ámbito de aplicación del problema a resolver con el objetivo de sentar una base que justifique la realización de este proyecto. Se estudian de esta forma las ventajas e inconvenientes de cada una de ellas, de modo que se pueda diseñar finalmente una solución que cubra un conjunto de algunas de las principales ventajas de todas ellas.

3. Análisis de objetivos y metodología

En segundo lugar se han de especificar claramente cuáles son los principales objetivos que se desean lograr a la finalización del proyecto, con la intención de tener presente en todo momento cual es el objetivo final y cuál será el mejor camino para llegar a él. Además, también se especifican una serie de objetivos de tipo secundario. Objetivos que se deben tener en cuenta mientras se realiza la implementación de producto final, pero que no son de igual importancia que los objetivos principales. Estos objetivos equivaldrán generalmente a requisitos adicionales que se pueden añadir al producto final para aumentar sus funcionalidades o facilitar las posibles vías futuras que puedan existir, y que requieran de un grado de complejidad mayor o de un tiempo de implementación demasiado excesivo.

Después se analizan problema y objetivos en pos de elaborar una correcta y completa especificación de los requisitos necesarios para cumplir con todos los objetivos que se deben cumplir, funcionalidades que se deben implementar y restricciones que delimitarán el posterior diseño.

Una vez se tienen claros cuales son los objetivos y requisitos que se deben cumplir comienza el proceso de diseño. Se realiza en primer lugar un diseño inicial con las ideas básicas que compondrán el primer esquema de implementación y por consiguiente la solución final. De esta forma dicho primer diseño debe cumplir los objetivos y requisitos además de servir como base para futuros diseños e implementaciones, siendo de esta forma y a grandes rasgos la idea básica de la solución elegida.

Con este primer diseño presente, se estudian los diversos métodos o formas de trabajo sobre los que se implementarán las ideas básicas. Se seleccionan los que se consideren oportunos en ese momento. Y por último se realiza un segundo diseño, el cual será más específico y contendrá con mayor nivel de detalle las soluciones optadas para cada parte que compone el diseño inicial.

Comienza en estos momentos una etapa de implementación y rediseño constante de la solución. Siempre siguiendo esta síntesis. Implementación de lo diseñado en primer lugar y rediseño de la solución en caso de ser necesario –por diversos motivos como problemas a la hora de realizar el primer diseño, obtención de una mejor solución para una parte del diseño, tecnologías obsoletas o nuevas tecnologías que no se habían tenido en cuenta, entre otros–.

Una vez que se termina por completo la implementación comienza la etapa de evaluación y pruebas donde se deben de identificar los posibles errores de implementación o funcionamiento y, por supuesto, solucionarlos. Esta última parte puede dar lugar a un nuevo rediseño e implementación en función de la gravedad y el tipo de error.

Por último se han de estudiar los resultados obtenidos en el proceso anterior, analizando con visión crítica el funcionamiento y el cumplimiento tanto

de los objetivos previstos, como de los requisitos que se establecieron para la solución final. Obteniendo finalmente una visión global de las posibles vías de implementación futuras, para posibles mejoras, o soluciones de fallos que no se han logrado solucionar.

3.4. Valoración de objetivos

En este apartado se realizará una valoración crítica sobre los objetivos finales que se han cumplido con la finalización del proyecto. Aunque esta valoración será muy breve puesto que en el Capítulo 6 se analizan en mayor profundidad los resultados y conclusiones obtenidos en este proyecto.

Por lo general, se puede decir que la finalización de este proyecto ha sido muy gratificante y satisfactoria para mí en cuanto a la consecución de todos los objetivos principales que se marcaron al inicio del mismo. Además también se ha logrado alguno de los objetivos secundarios con lo que finalmente se facilita la tarea de instalación esencialmente en el ámbito de la Universidad de Granada, una gran ventaja de cara a futuras implementaciones y vías de desarrollo.

Capítulo 4

Diseño y resolución del trabajo

En este capítulo se pretende mostrar la realización de todo el trabajo realizado hasta la consecución del proyecto final. El capítulo se estructura en dos secciones claramente diferenciadas. Por un lado se detallará el proceso de diseño que se ha realizado de la solución final atendiendo a las diferentes partes que componen el mismo (Sección 4.1). Y por otro se detallará la implementación final realiza a partir de dicho diseño (Sección 4.2).

4.1. Diseño

En esta sección se establecerá el diseño realizado de la solución final atendiendo a las diferentes etapas del proyecto. En primer lugar y con un mayor nivel de abstracción se mostrará el diseño de la arquitectura de forma que quede definido el comportamiento global del sistema y se muestren todas las posibles soluciones posibles que se planteaban al inicio de este proyecto. Posteriormente se profundizará en su estructura a través de los diferentes diagramas atendiendo a su vez al esquema general y las subdivisiones que este pueda tener.

4.1.1. Diseño global

4.1.1.1. Diseño previo

En este primer apartado se muestra el primer diseño que se realizó al inicio del proyecto, diseño que se realizó previamente a cualquier investigación o búsqueda de solución. Este es un diseño que sirve para comprender la utilidad

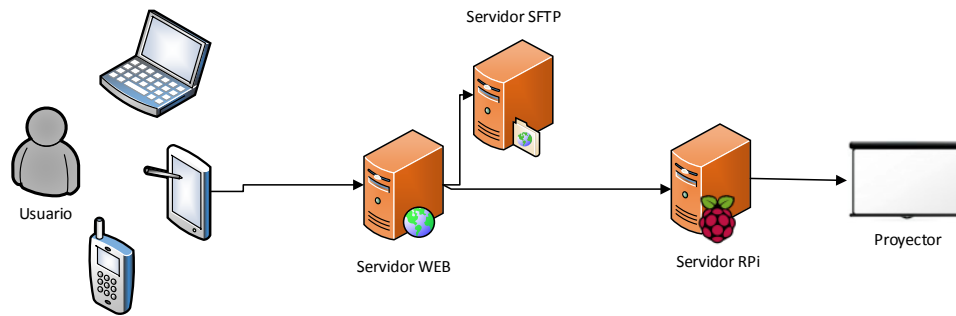


Figura 4.1: Diseño previo

final del proyecto y a partir del cual se puede comenzar a pensar en las posibles soluciones que se podrían implementar. Dicho diseño se puede ver en la Figura 4.1.

Tal y como se ha especificado en el Capítulo 3, uno de los objetivos del proyecto es que se pueda extender a diferentes plataformas con facilidad en un futuro, de forma que cualquier dispositivo pueda hacer uso de él. Por esta razón se ha elegido implementar el servidor web. Un servidor al que cualquier dispositivo accederá sin problemas de compatibilidad.

Además, uno de los objetivos fundamentales es el almacenamiento centralizado del material docente. Por esta razón se implementa el servidor FTP que se muestra en la Figura 4.1. Y por último, se ha de implementar un tercer servidor, situado éste en una Raspberry Pi, también descrita anteriormente, que será el encargado de manejar el material a proyectar.

Una vez queda definido de forma clara este diseño básico es fácil ver como existen multitud de posibles implementaciones diferentes para esta solución. Así como también se pueden introducir algunas mejoras.

En primer lugar, y para cumplir con un requisito secundario pero importante de cara a vías futuras, se añade un servidor RADIUS como servidor autenticación y un servidor LDAP como gestor de base de datos de usuarios. La razón de utilizar este tipo de servidor es porque la autenticación utilizada por los servicios de Internet ofrecidos por la Universidad de Granada se basa fundamentalmente en un servidor RADIUS centralizado y una base de datos LDAP para el control de todos los usuarios pertenecientes a la institución [51]. De esta forma, se consigue mayor facilidad para la implantación del sistema en las aulas docentes. Por esta razón, el diseño inicial se ve modificado añadiendo estos dos nuevos servidores, obteniendo finalmente como diseño básico la Figura 4.2

Por otro lado, cumpliendo con las especificaciones del Capítulo 3, la seguridad es un punto fundamental en la implementación. En este sentido, se

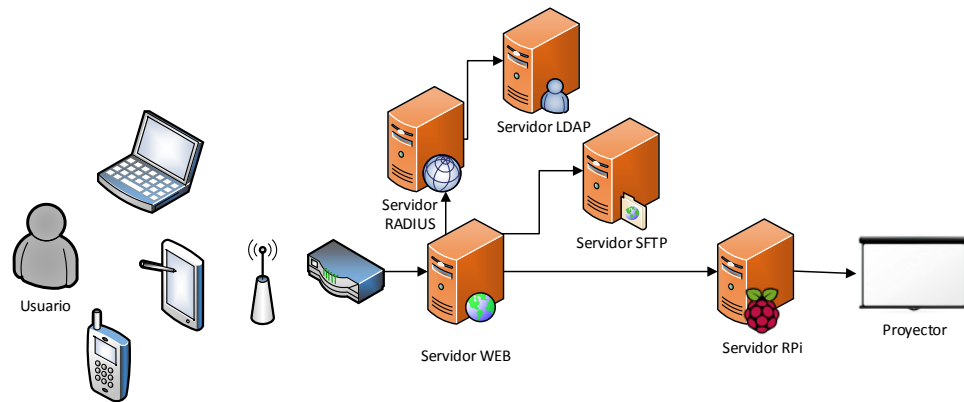


Figura 4.2: Diseño global de la implementación

ha de cuidar la comunicación entre servidores y entre usuario y servidor. Por esta razón, se decide también desde el primer momento utilizar, en lugar de un servidor FTP, un servidor SFTP. Aumentando de esta forma la seguridad de nuestro sistema. Además, también se configurará el servidor web de forma que utilice HTTPS en lugar de HTTP. Otra medida que aumentará dicha seguridad.

4.1.1.2. Estudio de las posibles implementaciones

Una vez se tiene claro cuál será el diseño final de la solución, se debe empezar a buscar las posibles soluciones, estudiar todas sus ventajas e inconvenientes, y elegir la mejor solución comparando unas con otras.

La primera elección que se debe hacer es la situación donde se instalarán todos los servidores. Evidentemente, el servidor instalado en la Raspberry Pi queda al margen de esta decisión. La solución más fácil y cómoda para esto es utilizar un ordenador personal con sistema operativo Linux [41] donde se instalarán todos los servidores de forma fácil y rápida. El sistema operativo elegido es Linux ya que en estos sistemas la instalación, configuración y programación de los servidores es más cómoda. Además se pueden aprovechar los conocimientos ya adquiridos a lo largo del grado sobre el uso de la consola.

Hay que aclarar que esta decisión está pensada para el proceso de realización de este proyecto. Para casos futuros en los que se realice la implementación del proyecto en la universidad por ejemplo, los servidores deberán ser instalados en una máquina de mayor potencia que pueda soportar todo el tráfico de datos que éstos generarán. En especial, el diseño se ha realizado de forma que se puedan utilizar los servidores actuales de los que dispone la universidad de forma que la implementación de este proyecto no requiera

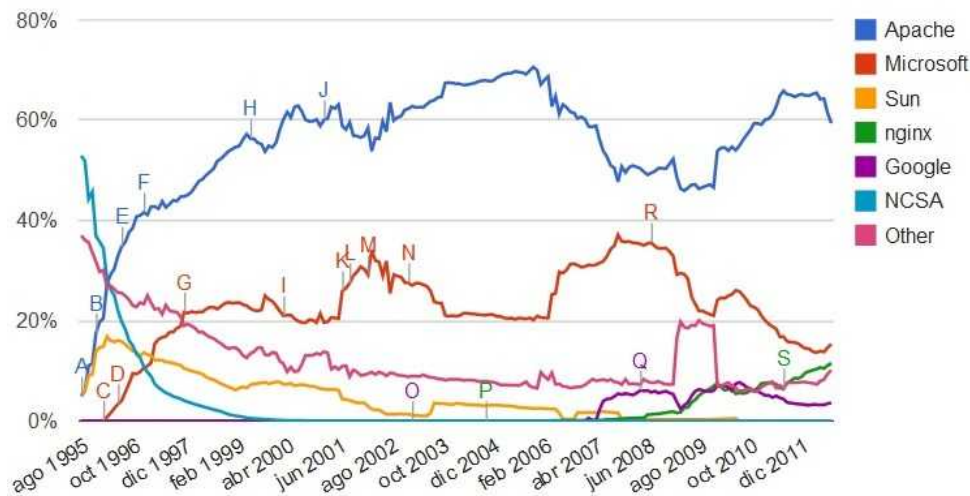


Figura 4.3: Uso de servidores HTTP

grandes modificaciones y resulte sencilla de realizar.

Con esta información se comienza con la toma de decisiones en lo que al diseño respecta:

Servidor WEB

Para el caso del servidor web la elección es bastante simple. Existen diversos servidor HTTP que son livianos, pero el servidor Apache [36] es el más utilizado debido a la facilidad de instalación, las extensas posibilidades de configuración y los módulos de seguridad que dispone. Apache se ha convertido en el servidor web más utilizado entre el resto de opciones, tal y como se puede ver en la Figura 4.3 obtenida de un estudio de INTECO (Instituto Nacional de Tecnologías de la Comunicación)[52] [53].

Servidor FTP

Para la elección del servidor FTP hay que tener varias consideraciones en cuenta; debe ser compatible con Linux, debe poder implementar SFTP para proporcionar seguridad a las comunicaciones y además debe ser de fácil configuración y manipulación. Ante esto se ha elegido usar el servidor VSFTPD [37].

Este servidor no es la única opción válida, Filezilla [54] y OpenSSH [55] son dos ejemplos de las diferentes posibilidades que existe de forma

gratuita actualmente. Se ha elegido utilizar VSFTPD porque es un servidor altamente configurable y muy utilizado en los entornos Linux.

Servidor RADIUS + base de datos

En este caso, las opciones están algo limitadas. La base de datos debe ser LDAP, puesto que es la utilizada en el modelo de autenticación de la universidad, como se ha descrito anteriormente. Así, el servidor utilizado es OpenLDAP [39].

Para el caso del servidor RADIUS se utilizará un servidor gratuito, compatible con el entorno Linux y que permita utilizar la base de datos tipo LDAP anterior para la autenticación. En este caso el servidor RADIUS más utilizado habitualmente y cuyos módulos permiten autenticación mediante LDAP es FreeRADIUS [38], por lo que es el servidor que se utilizará en este caso.

Servidor RPi

Este servidor corresponde al situado en la Raspberry Pi. Dicho servidor será el encargado de obtener, mostrar y manejar el material que utilizará el docente.

El diseño de este servidor está condicionado por la Raspberry Pi. La distribución elegida para instalar es de tipo Linux por compatibilidad y facilidad de configuración. Pero no se puede elegir un Linux cualquiera, ya que Raspberry Pi tiene una arquitectura y *hardware* específicos. En su web oficial [40] se pueden encontrar diferentes tipos de distribuciones adaptadas a diferentes necesidades, las cuales están optimizadas y preparadas exclusivamente para ser instaladas en Raspberry Pi [42]. Entre ellas se pueden distinguir:

- RASPBIAN *Debian Wheezy* [56]
- OPENELEC *An XBMC Media Centre* [57]
- PIDORA *Fredora Remix* [58]
- RASPBMC *An XBMC Media Centre* [59]
- ARCH LINUX *A lightweight Linux distribution* [60]
- RISC OS *A non-Linux distribution* [61]

Para el caso que nos interesa, las apropiadas son RASPBIAN, PIDORA y ARCH LINUX, ya que son entornos Linux. De las tres, la elegida es RASPBIAN, un sistema muy parecido al entorno Debian. La razón por la que se escogió esta distribución es porque es la más usada por la

comunidad y para la que mayor número de programas existen es estos momentos. Todo resulta más fácil de instalar e implementar cuando las librerías y el código fuente ya están incluidos en la distribución.

Por tanto, el diseño del servidor finalmente estará condicionado por esta distribución, de modo que hay que tener en cuenta el rendimiento que puede tener y el *software* que es compatible.

El diseño de este servidor marcará además en algunos aspectos el diseño de la aplicación cliente con la que el usuario final se conectará a dicho servidor. Esto se debe a que cliente y servidor deben establecer conexión e intercambiar mensajes de forma que ambos deben ser compatibles.

Teniendo estos aspectos en cuenta, muchas son las propuestas iniciales a la hora de realizar el diseño de este servidor, aunque sólo tres adquieren un mayor peso.

- Se puede utilizar un segundo servidor web al que se redireccionaría desde el servidor web anterior, de forma que se manejaría el archivo en este nuevo entorno web. Así, la aplicación cliente se reduce a una página o interfaz web.
- Se puede utilizar un servidor con *socket* utilizando lenguajes de programación C o Java (los más comunes). En este caso, la aplicación cliente debe estar también implementada en dicho lenguaje de programación.
- Se puede utilizar un servidor con *socket* pero utilizando en este caso PHP como lenguaje de programación. De esta forma se podría realizar la conexión *socket* desde la propia interfaz web y mandar Y enviar las órdenes a través de esta.

Finalmente la solución por la que se ha optado es la utilización de una aplicación cliente y un servidor con *socket* implementado con Java. En el proceso de implementación se han probado todas y cada una de estas tres posibles soluciones y en todas ellas se ha llegado a una encrucijada difícil de resolver (librerías o implementaciones que no se han logrado que funcionasen, entre otros). En este sentido la solución elegida es la que más fácil resultó y la que finalmente se ha logrado implementar.

Dentro de este mismo punto se ha de elegir también el *software* que se utilizará para mostrar el material docente. Existen gran cantidad de visores de ficheros en formato PDF, pero se necesita encontrar uno que sea lo más liviano posible, que se pueda abrir desde línea de comandos (esto simplifica su ejecución desde un programa servidor) y que sea fácilmente usable desde ésta o en su defecto mediante el teclado. Para este caso también se han barajado numerosas opciones. Las tres que más en profundidad se han estudiado son:

- FBIDA [62]
- Green [63]
- MuPDF [46]

El primero de ellos, fue la primera opción utilizada pero finalmente no resultó ser tan potente y bueno como en un principio parecía. El control del archivo PDF que se estaba mostrando no funcionaba bien y surgían ciertos problemas cuando se realizaban las primeras pruebas.

Con el segundo *software* ocurrió algo parecido. En este caso, se debe a problemas de compatibilidad por la antigüedad del *software*.

Finalmente la opción utilizada es MuPDF, un visor de PDF ligero, que puede ser utilizado mediante consola y que permite la total manipulación del fichero desde el teclado. Además con este *software* también se pueden mostrar imágenes, no sólo documentos en formato PDF, lo que ayuda a incrementar las funcionalidades del proyecto.

4.1.1.3. Diseño final

Una vez que se han elegido las implementaciones que se van a realizar en cada una de las partes que componen el diseño de la Figura 4.2, se puede establecer un diseño con mayor nivel de detalle que ayude a comprender el final funcionamiento y a establecer los pasos para la implementación posterior.

En la Figura 4.4 se puede ver el diseño final de la solución. En la imagen se muestra con un grado de detalle mayor al anterior el funcionamiento que tendrá la solución implantada. Se puede ver cómo será la comunicación entre servidores, los datos necesarios en cada momento y cuáles serán los diferentes pasos que se realizarán hasta conseguir finalmente que se muestre el material docente en el proyector.

En esta imagen, las figuras cuadradas corresponden a los servidores o aplicaciones y las figuras con forma de rombo a los datos necesarios que se utilizan en dichos servidores o aplicaciones.

El proceso mostrado en la Figura 5.4 es el siguiente. Inicialmente se utilizan los datos de usuario (usuario y contraseña) para poder entrar al entorno web. Antes de poder acceder a él, el servidor HTTP establece conexión con el servidor RADIUS, al cual le manda los datos del usuario introducidos. El servidor RADIUS a su vez establece una nueva conexión con el servidor LDAP. Dicho servidor será el encargado de comprobar la veracidad de los datos introducidos comparándolos con los datos almacenados en la base de datos.

En caso de ser datos incorrectos el proceso vuelve al inicio. Si los datos son correctos el usuario tiene permiso para entrar en el entorno web. En este

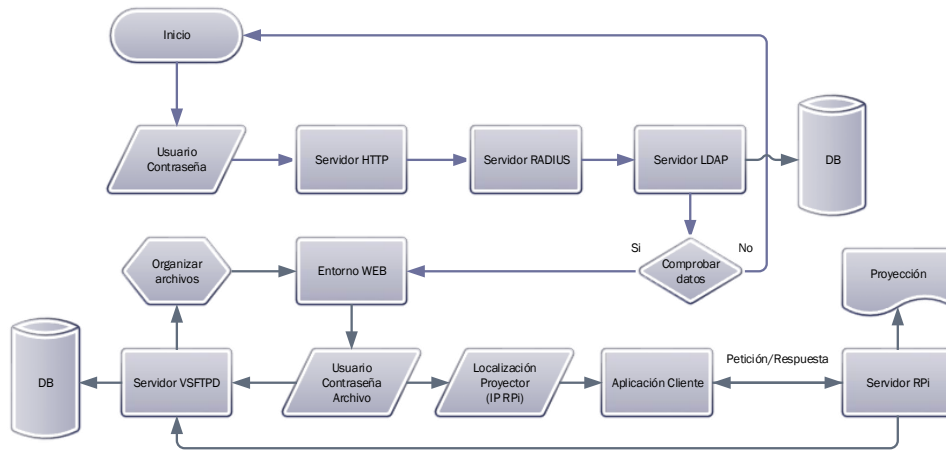


Figura 4.4: Diseño general de funcionamiento

punto se pueden diferenciar dos acciones. Una correspondiente al manejador de ficheros. La otra acción es la correspondiente a mostrar un fichero en el proyector.

En el primer caso son necesarios los datos de usuario y el nombre del archivo para poder acceder al servidor de ficheros. Y una vez establecida la conexión se podrá borrar, renombrar o ver los ficheros guardados, así como añadir nuevos.

Con respecto a la acción correspondiente a mostrar el fichero en el proyector se debe hacer uso en este caso de la aplicación cliente implementada. Dicha aplicación necesita como datos de entrada los datos de usuario, el fichero que se quiere mostrar y la dirección IP del servidor RPi, es decir, la localización del proyector que se va a utilizar.

Desde la aplicación cliente se establecerá una conexión mediante *socket* con el servidor situado en la Raspberry Pi, el cual necesitará también los datos de usuario y el fichero que se quiere mostrar.

Finalmente, una vez que el documento es proyectado, se produce una comunicación continua de petición-respuesta con el envío de las órdenes utilizadas para manejar las proyecciones.

4.1.2. Diseño de los servidores

En este apartado, se profundizará en el diseño de los diferentes servidores. Para el caso de los servidores VSFTPD, FreeRADIUS y LDAP se ampliará la información en la sección 4.2 puesto que no necesitan ser diseñados, sino simplemente configurados.



Figura 4.5: Interfaz web de usuario

4.1.2.1. Servidor WEB

Para el caso del servidor web se ha utilizado lenguaje HTML para el diseño web y PHP para lograr un entorno web dinámico. El entorno debe ser muy intuitivo y sencillo de utilizar por el usuario. Por esta razón la interfaz que el usuario final ve cuando accede a dicho servidor es una única página de inicio en la que se mostrarán los ficheros almacenados en el servidor de ficheros.

Desde dicha página de podrá subir un nuevo archivo, borrar un archivo existente, renombrar o descargar un archivo. Todo esto gracias a la programación dinámica de PHP. Además desde dicho entorno también se podrá acceder a la aplicación cliente implementada para establecer la conexión con el servidor situado en la Raspberry Pi y mostrar el material docente. Un ejemplo de esta interfaz se puede ver en la Figura 4.5. Para mayor información acerca de las utilidades y el funcionamiento, ver el Anexo A. Manual de usuario.

Para que todo esto sea posible, se han diseñado diferentes ficheros PHP que ayudan a la realización de todas estas tareas. En la Figura 4.6 pueden verse las relaciones entre dichos ficheros.

Index

Es la página inicial que aparecerá al entrar en el servidor WEB. En ella se puede acceder al entorno web de usuario o una nueva página desde la cual se puede crear una nueva cuenta de usuario. Contiene la



Figura 4.6: Elementos del servidor WEB

función:

autenticarUsuario() Función que comprueba la veracidad de los datos introducidos por el usuario antes de redirigir a la página de inicio.

Registrar usuario

Página utilizada para crea una nueva cuenta de usuario. Contiene las funciones:

registrarNuevoUsuario() Función que crea una nueva cuenta de usuario en el sistema global.

nuevoUsuarioLDAP() Función utilizada para introducir una nueva entrada en la base de datos LDAP.

comprobarDatos() Método auxiliar al anterior, comprueba que el usuario no exista.

registroUsuarioLDAP() También método auxiliar al anterior, realiza la creación de la nueva entrada.

nuevoUsuarioFTP() Crea una nueva cuenta de usuario en el servidor VSFTPD.

Página de inicio

Página principal del entorno web donde el usuario puede realizar todas las acciones correspondientes con el servidor de ficheros, así como acceder a la aplicación cliente. Esta página contiene los métodos:

conexionnuevoUsuarioFTP() Establece una conexión con el servidor de ficheros VSFTPD.

verArchivos() Obtiene la lista de ficheros del usuario con sus características a partir de una conexión al servidor ya realizada.

subirArchivo() Guarda un nuevo fichero en el servidor de ficheros. Requiere de una conexión al servidor de ficheros ya realizada.

eliminarArchivo() Elimina un archivo del servidor de ficheros. Requiere de una conexión al servidor ya realizada.

renombrarArchivos() Cambia el nombre de un archivo existente en el servidor de ficheros. Requiere de una conexión al servidor ya realizada.

Página aplicación cliente

Esta página contiene los datos necesarios para lanzar la aplicación cliente en el entorno web. Como la aplicación está implementada en lenguaje Java y se está trabajando en un entorno HTML la aplicación lanzada es un Applet (la forma que tiene Java de diseñar aplicaciones web).

La web primero pide la selección del aula donde se quiere mostrar el material. Esta selección está en formato texto y se debe pasar posteriormente a IPv4.

Además, se hará una distinción según se trate de un documento en formato PDF o de una imagen. Esto se debe a que la interfaz utilizada en cada caso será diferente. Aunque la única diferencia reside en la cantidad de acciones que se podrán realizar.

Contiene el método:

obtenerIP() Función para obtener la IP de una determinada localización (generalmente una habitación o aula), a partir de un pequeño fichero donde se almacena dicha información.

Cambiar contraseña

Permite cambiar la contraseña de la cuenta que utiliza un determinado usuario. Contiene los métodos:

cambiarPassword() Función para cambiar la contraseña de una cuenta de usuario en el sistema global.

cambiarPasswordLDAP() Cambia la contraseña de una entrada en la base de datos LDAP.

`comprobarDatos()` Método auxiliar al anterior utilizado para comprobar que la entrada exista.

`cambiarPassword FTP()` Cambia la contraseña de la cuenta del usuario en el servidor VSFTPD.

Cerrar sesión

Destruye todos los datos de sesión que se han estado utilizando en el entorno web y se redirecciona al inicio. Si no se realiza este proceso puede que al volver a abrir la página en el mismo navegador redirija a la página principal sin necesidad de tener que introducir los datos de usuario.

4.1.2.2. Servidor RPi

Este servidor, como se ha dicho anteriormente, se implementará con el lenguaje de programación Java y será el encargado de realizar las acciones sobre el documento que se mostrará en el proyector.

Su funcionamiento consiste básicamente en el establecimiento de conexiones con los clientes. Una vez que se establece una conexión, recibe los datos de usuario y los datos del fichero. Posteriormente se conecta al servidor de ficheros para obtener el archivo y guardarlo de forma temporal en la Raspberry Pi. En este momento comienza el período donde se puede manejar el documento. Y finalmente cuando se quiere cerrar la conexión, se borra el fichero temporal de la Raspberry Pi y se cierran todas las conexiones existentes con el cliente, quedando de nuevo escuchando el puerto a la espera de un nuevo cliente. Esto se verá de forma más clara en la Figura 4.7 donde se muestra el diagrama de estados del servidor.

En este el diagrama de estados se puede ver como el servidor está en constante ejecución. La única forma de parar el servidor es de forma manual por el usuario haciendo uso del método `cerrarServidor()`.

Atributos

puerto Variable introducida con el usuario desde línea de comandos. Indica el número de puerto en el que comenzará el servidor a escuchar. Su valor por defecto se establece a 8888.

IP_ftp_server Variable introducida con el usuario desde línea de comandos. Indica la dirección IP del servidor de ficheros VSFTPD. Valor por defecto 192.168.1.1.

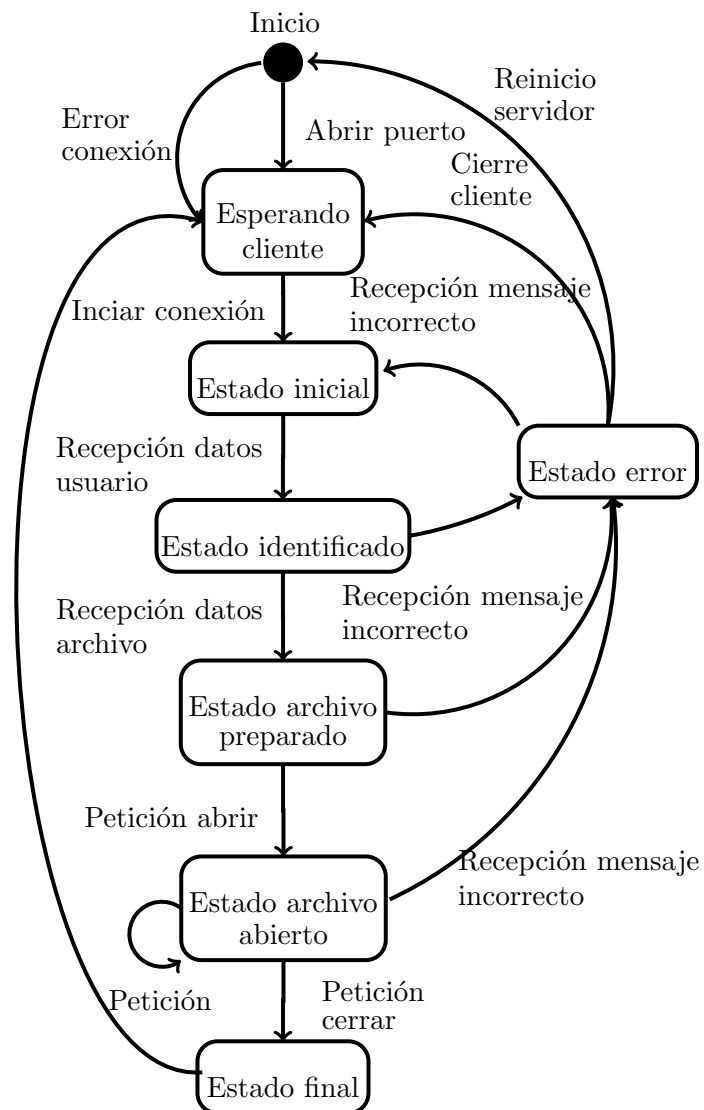


Figura 4.7: Diagrama de estados del ServidorRPi

socketConexion Variable con la información referente la información del *socket*.

canalSalida Variable utilizada para mandar datos por el buffer de salida desde el servidor.

canalEntrada Variable para recibir datos en el buffer de entrada del servidor.

teclado Variable de la clase Robot proporcionada por Java, la cual es utilizada para simular las pulsaciones de teclas en la máquina donde corre el servidor, es decir, en la Raspberry Pi.

cerrar Variable utilizada cuando se quiere cerrar por completo el servidor.

Métodos

abrirServidor() Inicia el servidor en el puerto indicado.

nuevoCliente() Espera la conexión con un nuevo cliente.

cerrarConexion() Cierra la conexión actual con el cliente.

cerrarServidor() Cierra por completo el servidor.

reiniciarServidor() Reinicia el servidor.

servidorToCliente() Envía un mensaje al cliente.

guardarLog() Guarda un mensaje en el fichero de logs.

errorSolicitud() Maneja un error producido en la recepción de una solicitud/mensaje.

manejarErrorDescargar() Maneja el error que se produce cuando se descargar un archivo del servidor de ficheros.

errorConexionFTP() Maneja el error producido cuando no se ha podido establecer una conexión al servidor de ficheros.

errorLoginFTP() Maneja el error producido durante el login con el servidor de ficheros.

errorObtenerArchivoFTP() Maneja el error producido en la descargar del fichero.

errorInternoDescargar() Maneja un error interno no reconocible por el servidor.

abrirArchivo() Abre un archivo en el escritorio de la máquina donde se está ejecutando el servidor.

accionPDF() Decide que acción realizar sobre el documento pdf en función de la orden que llega desde el cliente.

cambiarPaginaPDF() Cambiar la página que se está visualizando de un archivo pdf.

buscarPalabraPDF() Busca una palabra en el fichero pdf que se está utilizando.

accionIMG() Decide que acción realizar sobre la imagen en función de la orden que llega desde el cliente.

borrarArchivoTemporal() Borra el fichero que se había descargado anteriormente para mostrar.

4.1.3. Diseño de la aplicación cliente

La aplicación cliente debe estar implementada con el lenguaje de programación Java. Además, como se ha dicho en anteriores apartados de este capítulo, ha de ser una aplicación Applet.

En este sentido, se han creado dos clases, una que contendrá la interfaz gráfica del usuario, el Applet, y otra que contendrá los métodos utilizados por el cliente.

La primera clase tiene como único propósito la creación de una interfaz gráfica sencilla para el usuario, a partir de la cual se creará un objeto de la clase ClienteRPi. Y dicho objeto será el que se encargue de toda la comunicación con el servidor.

En este sentido, la clase ClienteRPi se encarga de establecer la conexión con el servidor RPi, de enviar los datos de usuario, enviar los datos del fichero que se quiere mostrar y enviar finalmente una petición para mostrar dicho fichero. A partir de este instante la conexión ya está establecida por completo sin errores y el documento es mostrado en el proyector. Por tanto, a partir de este momento los mensajes intercambiados corresponden a las peticiones del cliente (e.g. avanzar página) y respuestas del servidor. El diagrama de estados de la Figura 4.8 ayuda a comprender su funcionamiento.

La clase ClienteRPi viene descrita por:

Atributos

servidor Variable que contiene la dirección IP del servidor RPi.

puerto Variable que indica el puerto el que conectará al servidor RPi.

usuario Variable que contiene el nombre de usuario del cliente que está utilizando la aplicación.

password Variable que indica la contraseña de usuario del cliente que está utilizando la aplicación.

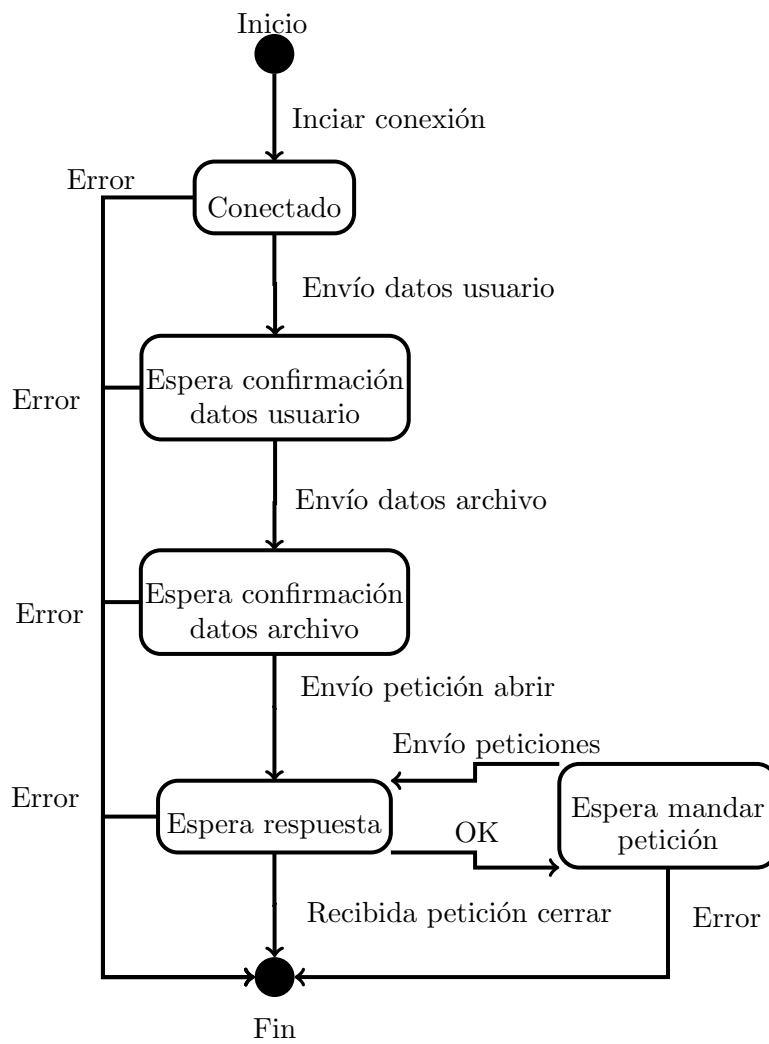


Figura 4.8: Diagrama de estados del ClienteRPi

archivo Variable que contiene el nombre del archivo que se quiere mostrar en el proyector.

socketConexion Variable con la información referente la información del *socket*.

canalSalida Variable utilizada para mandar datos por el buffer de salida desde el cliente al servidor.

canalEntrada Variable para recibir datos en el buffer de entrada desde el servidor al cliente.

Métodos

abrirCliente() Inicia una nueva conexión con el servidor en el puerto indicado.

cerrarCliente() Cierra la conexión actual con el servidor y termina la ejecución de la aplicación cliente.

clienteToServidor() Envía un mensaje al servidor RPi.

esperandoServidor() Espera hasta que recibe un mensaje procedente del servidor RPi.

4.1.4. Diseño del protocolo

En este apartado se describirá el protocolo diseñado para la comunicación entre servidor y cliente RPi, de forma que esta información pueda servir de documentación para entender mejor el comportamiento del sistema y ayudar a la elaboración futura de modificaciones en el código.

En la Figura 4.9 se muestra un esquema con el intercambio de mensaje común entre cliente y servidor. En dicho esquema, el servidor está inicialmente escuchando en el puerto 8888, y al cerrar la conexión con el último mensaje recibido *“BYE”* por parte del cliente, vuelve a escuchar en el mismo puerto.

Evidentemente, en este esquema no se han tenido en cuenta los posibles fallos durante el proceso. Se ha mostrado una visión general del correcto funcionamiento. En caso de fallo el mensaje enviado al cliente por parte del servidor tiene la estructura *“ERROR! + breve descripción del error”*.

Con esta estructura fundamental de los mensajes y la ayuda de los diagramas de estados mostrados anteriormente (Figura 4.7 y Figura 4.8) es fácil entender el funcionamiento básico de la aplicación cliente-servidor diseñada. Con estas tres imágenes se puede ver cuál es el comportamiento global del sistema y cómo cliente y servidor avanzan a través de los distintos estados en función de los mensajes que se van enviando en cada momento.

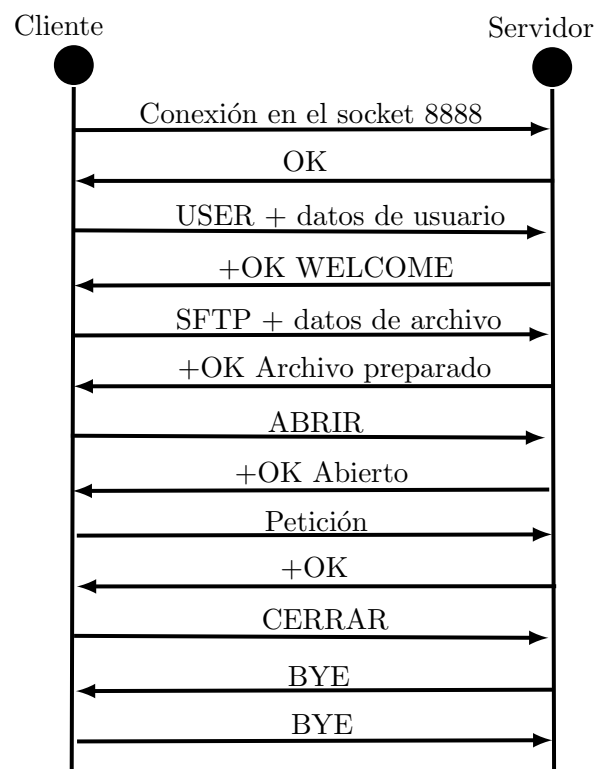


Figura 4.9: Intercambio de mensajes ClienteRPi-ServidorRPi

El primer mensaje que debe mandar el cliente hacia el servidor sigue la siguiente estructura; “*USER + usuario + password*”, donde *usuario* y *password* son los datos de la cuenta del cliente. Los datos deben ir separados por un espacio.

Cuando el servidor recibe dicho mensaje guarda los datos y responde con un mensaje “*+OK WELCOME*”.

El segundo mensaje que se debe mandar sigue la siguiente estructura; “*FTP + archivo*”, donde *archivo* corresponde al nombre del fichero que se quiere mostrar en el proyector

Cuando el servidor recibe este dato, inicia el proceso de obtención del archivo mediante una conexión SSL al servidor de ficheros, y una vez finalizado responde con un mensaje “*+OK Archivo preparado*”.

En tercer lugar, el cliente manda la petición de abrir el archivo mediante el mensaje “*ABRIR*”. En este momento el servidor abre y muestra el fichero en el proyector, responde con un mensaje “*+OK Abierto*” y se inicia en estos momentos el proceso de petición – respuesta para manejar el documento.

Los pasos descritos anteriormente no pueden variar, siguen obligatoriamente esta estructura. Cualquier variación en uno de estos mensajes provoca un fallo en el servidor y el consiguiente cierre de la conexión con el cliente.

A partir de este momento las peticiones son libres, es decir, las peticiones variarán en función del tipo de acción que se quiera realizar. Así se pueden diferenciar:

- AVANZAR	Para avanzar página
- RETROCEDER	Para retroceder página
- PAGINA	Para avanzar a una página concreta
- COMPLETA	Para mostrar en pantalla completa
- ANCHO	Para ajustar la imagen al ancho de la pantalla
- ALTO	Para ajustar la imagen al alto de la pantalla
- GIRAR DERECHA	Para girar la imagen a la derecha
- GIRAR IZQUIERDA	Para girar la imagen a la izquierda
- ZOOM MAS	Para aumentar el zoom
- ZOOM MENOS	Para disminuir el zoom
- MOVER ARRIBA	Para mover hacia arriba
- MOVER ABAJO	Para mover hacia abajo
- MOVER DERECHA	Para mover hacia la derecha
- MOVER IZQUIERDA	Para mover hacia la izquierda
- BUSCAR PARALBA	Para buscar una palabra
- BUSCAR SIGUIENTE	Para buscar siguiente palabra
- BUSCAR ANTERIOR	Para buscar palabra anterior

- ACTUALIZAR Para actualizar la imagen
- REDIMENSIONAR Para redimensionar la imagen
- CERRAR Para cerrar la imagen

En todos los casos anteriores, cuando el servidor recibe cualquiera de estos mensajes, realiza en primer lugar la acción correspondiente, y confirma su realización posteriormente con un mensajes “+OK”. Aunque existe una salvedad, el caso de recibir un mensaje “CERRAR”. Cuando el servidor recibe este mensaje comienza el proceso de cierre de la conexión. En este proceso el servidor cierra el archivo, borra el archivo temporal, y manda un mensaje de vuelta al cliente “BYE”. El cliente al recibir esta petición confirma con otro mensajes “BYE”. Cuando este segundo mensaje llega al servidor, éste cierre la conexión con el cliente y vuelve a escuchar de nuevo en el puerto para establecer una nueva conexión con otro cliente.

4.2. Implementacion

En esta sección se procede a detallar la implementación del sistema de acuerdo a los criterios de diseño presentados en la sección anterior (Sección 4.1. El presente capítulo se estructura a las diferentes secciones que componen la arquitectura del proyecto.

4.2.1. Servidor SFTP

Como se vio en la sección anterior, el servidor seleccionado para funcionar como servidor de ficheros es VSFTPD. Para realizar la instalación e implementar la correcta configuración se utiliza la documentación facilitada en su propia página web, desde donde también se puede descargar dicho servidor [37].

Adjunto a este documento, en el Anexo B. Manual de instalación, se puede encontrar el manual de instalación donde se detallan y describen los pasos a seguir para la correcta configuración del servidor. Este apartado se ceñirá por tanto a explicar en menor nivel de detalle cómo se ha llevado a cabo la configuración de dicho servidor.

Una vez que el servidor está instalado el servidor se cambiar la configuración del mismo mediante el fichero `/etc/vsftpd.conf`. En este caso se realizan algunas configuraciones básicas con permitir el uso de usuarios locales y utilizar un fichero para el enjaulado de usuarios, es decir, para que sólo los usuarios que se encuentran dentro de dicho fichero puedan utilizar el servidor. Este fichero es `/etc/vsftpd.chroot_list`.

El siguiente paso es la creación de usuarios. La forma de crear usuario

también se describe en la documentación adjunta por lo que tampoco se detallara la explicación. Se ha de crear una carpeta donde el usuario tendrá sus ficheros guardados. También se creará una *shell* virtual o fantasma de modo que los usuarios no puedan conectarse a una sesión del sistema operativo, sino que solo puedan conectarse a la carpeta de usuario. Finalmente se procede a la creación de la nueva cuenta de usuario en la *shell* virtual y en la carpeta anteriormente creada. Por último se añade el usuario a la lista de usuarios enjaulados y se reinicia el servidor.

Un aspecto importante del servidor es la seguridad. Se debe implementar el servidor de forma que utilice una capa de transporte segura para las comunicaciones [64]. El protocolo FTP, aunque muy práctico, no codifica los datos ni siquiera de las credenciales del usuario. El hecho de que alguien pudiera capturar un paquete en la ruta sería muy grave para la seguridad del sistema. Por tanto, la solución es utilizar un protocolo seguro, en concreto SFTP.

Así, se configura el servidor VSFTPD para que trabaje de forma que la conexión esté cifrada. En primer lugar hay que generar un certificado. Dicho certificado se puede generar fácilmente utilizando OpenSSL [65]. Y en segundo lugar se ha de volver a cambiar la configuración del servidor de forma que se habilite el soporte para SSL en sus distintas versiones (TLS, SSLv2 y SSLv3) [66][67][68] y se añade la ruta hacia el certificado creado anteriormente.

4.2.2. Servidor LDAP

El servidor LDAP será el encargado de la gestión de la base de datos de usuarios. Al igual que en apartado anterior, en la página web oficial de OpenLdap [39] se puede descargar este servidor. Además, también se puede encontrar gran cantidad de información relevante a la configuración de este servidor.

Puesto que en el manual de instalación se detallan los pasos a seguir para configurar completamente este servidor, este apartado se centrará en explicar a grandes rasgos como se ha realiza su configuración.

LDAP es un servidor al que se añaden o modifican las entradas mediante un tipo de ficheros específicos, únicamente usados ente servidor. Se trata de los ficheros *.ldif*. En este sentido se debe añadir en primer lugar la descripción de los grupos de usuarios. Estos grupos marcarán una serie de características comunes a un grupo de usuario. El fichero utilizado para ello es:

En cuanto al fichero que se utiliza para la adición de nuevas entradas (usuarios) en la base de datos es el siguiente:

```
dn: ou=profesores, dc=ugr, dc=es
objectClass: top
objectClass: organizationalUnit
ou: profesores
description: Profesores

dn: ou=alumnos, dc=ugr, dc=es
objectClass: top
objectClass: organizationalUnit
ou: alumnos
description: Alumnos
```

Figura 4.10: Fichero de configuración grupos.ldif

```
dn: cn=nick ou=profesores, dc=ugr, dc=es
objectClass: top
objectClass: organizationalPerson
sn: nombre_completo
ou: Nick
userPassword: password
description: Profesor
```

Figura 4.11: Fichero de configuración profesor.ldif

Donde *nick* corresponde con la cuenta de usuario del correo electrónico de la Universidad de Granada, y *password* se cifra mediante usando una función hash de PHP antes de añadir la información a la base de datos.

4.2.3. Servidor RADIUS

Para habilitar la base de datos y que ésta sea de utilidad, se ha de configurar un servidor RADIUS. Como se vio en la sección anterior (Sección 4.1), se utilizará para ello FreeRADIUS [38]. De nuevo en su página web se puede obtener el *software* de forma gratuita, así como diversa información para la configuración de este servidor.

En este caso se ha de configurar el servidor para que utilice la base de datos LDAP anterior como si de su propia base de datos de usuarios se tratase. Al igual que en los anteriores casos, en el manual de instalación se detallará esta información.

Para la correcta configuración, como se va a hacer uso de la base de datos LDAP basta con configurar el servidor para que pueda utilizarla. Se debe instalar para ello el módulo *freeradius-ldap*. Una vez que la base de datos está completamente configurada, en la configuración de FreeRADIUS se habilita el módulo previamente instalado. Y se configura además que la autenticación en dicho servidor será servida por defecto por LDAP en lugar de por los ficheros de configuración propios. De esta forma se consigue integrar



Figura 4.12: Interfaz portada

completamente LDAP en FreeRADIUS.

4.2.4. Servidor HTTP

Tal y como se ha descrito en la fase de diseño, la implementación del entorno web, en lo que a diseño y apariencia se refiere, es muy sencilla, aunque por otra parte adquiere mayor relevancia la programación dinámica del mismo con PHP. Por ello, en primer lugar se debe realizar la instalación tanto del servidor web Apache como del lenguaje PHP y sus módulos asociados a dicho servidor Apache. Además será necesario instalar también los paquetes *php-radius* y *php-ldap* para poder manejar tanto el servidor RADIUS como la base de datos desde el entorno PHP.

Con PHP se puede conseguir realizar multitud de tareas gracias a sus numerosas clases [76]. En los siguientes apartados se detallará la implementación de los diferentes ficheros que constituyen el conjunto del servidor web y se demostrará la afirmación anterior.

4.2.4.1. index.php

Es página corresponde con la interfaz de portada de este servidor web. Esta portada es muy sencilla y minimalista. Se asemeja en cierto sentido a las interfaces utilizadas por la Universidad de Granada para acceder al correo electrónico propio. En la Figura 4.12 se muestra como es dicha interfaz.

Este fichero es además el encargado de la autenticación del usuario y la creación de la sesión del mismo que será utilizada durante la navegación por

el servidor. Esto se puede llevar a cabo gracias a la clase radius de PHP. El proceso seguido es el siguiente:

```
1 // Se crea un identificador para la autentificacion Radius
2 $radius = radius_auth_open();
3
4 // Se establecen las variables correspondientes al servidor RADIUS
5 $radius_host = 'localhost';
6 $radius_puerto = 1812;
7 $radius_secret = 'testing123';
8
9 // Se especifica el Servidor RADIUS
10 radius_add_server($radius, $radius_host, $radius_puerto, $radius_secret
    , 5, 3))
11
12 // Se especifica el tipo de solicitud RADIUS
13 radius_create_request($radius, RADIUS_ACCESS_REQUEST))
14
15 // Se establece el usuario y contraseña
16 radius_put_attr($radius, RADIUS_USER_NAME, $usuario);
17 radius_put_attr($radius, RADIUS_USER_PASSWORD, $password);
18
19 // Se obtiene la respuesta a la solicitud anterior
20 switch (radius_send_request($radius)) {
21 case RADIUS_ACCESS_ACCEPT:
22     //Todo ha ido bien. Se establece sesion
23     session_start();
24     $_SESSION['usuario'] = $usuario;           //ID usuario
25     $_SESSION['password'] = $password;         //Contraseña
26     $_SESSION['ftp_server_IP'] = 'localhost'; //IP del servidor FTP
27     //Redireccion a la pagina de inicio de usuario
28     header('Location: inicio.php');
29     break;
30 default:
31     //Algo no ha ido bien
32     return $error = 6;
33 }
```

Este código es una versión resumida de cómo se ha llevado a cabo la implementación de la autenticación con el servidor RADIUS. Por supuesto sólo se muestra cuando el funcionamiento es correcto, pero también se tienen presentes todos los errores o fallos que se puedan producir, de forma que se actúa de consecuentemente a estos.

4.2.4.2. registrar.php

Este fichero es el utilizado para añadir un nuevo usuario al sistema global del proyecto. Esta página sigue el mismo diseño que la página de portada (index). Pero en este caso se pide como datos a introducir el nombre del usuario, el correo electrónico referente a la Universidad de Granada y una contraseña. Dicha contraseña además debe ser de al menos 8 caracteres, y tener un mínimo de un número y una letra. Con esto se trata de aumentar la seguridad del sistema.

4. Diseño y resolución del trabajo

Una vez que se introducen los datos se utilizan las funciones expuestas en la sección anterior (Sección 4.1 para crear la nueva cuenta de usuario. En este caso se diferencian los servidores LDAP y VSFTPD. Para el primer caso se pueden utilizar las funciones que proporciona la clase LDAP de PHP. Un breve resumen de esto se muestra a continuación.

```
1 // Variables del servidor LDAP
2 $ldaphost = "ldap://localhost";           // servidor LDAP
3 $ldapport = 389;                          // puerto del servidor LDAP
4 $ldaprdn = 'cn=admin,dc=ugr,dc=es';       // ldap rdn or dn
5 $ldappass = 'admin';                     // contraseña
6
7 // Se realiza la conexion
8 $ldapconn = ldap_connect($ldaphost, $ldapport);
9
10 // Se establecen algunas opciones necesarias
11 ldap_set_option($ldapconn, LDAP_OPT_PROTOCOL_VERSION, 3);
12 ldap_set_option($ldapconn, LDAP_OPT_REFERRALS, 0);
13
14 // Se enlaza la conexion
15 $r = ldap_bind($ldapconn, $ldaprdn, $ldappass);
16
17 // Para comprobar los datos
18 $sr = ldap_search($ldapconn, "dc=ugr,dc=es", "(cn=$email)");
19
20 // Para agregar los datos ($info contiene la informacion)
21 $r = ldap_add($ldapconn, $dn, $info);
```

En lo que respecta al servidor VSFTPD la creación de la nueva cuenta es diferente. Será necesario en este caso ejecutar desde PHP comandos de la terminal de Linux para poder crear una nueva cuenta de usuario. La clase ftp de PHP solo permite manejar ficheros, en ningún caso se permite variar algo correspondiente a las cuentas de usuario. Será necesario por tanto añadir permisos al usuario del servidor Apache para que se puedan ejecutar dichos comandos.

4.2.4.3. inicio.php

Este fichero corresponde con la página principal del servidor web. Ésta página es la que el usuario deberá utilizar para realizar cualquier tipo de acción (e.g. lanzar la aplicación cliente o subir un archivo al servidor de ficheros). La interfaz de esta página se puede ver en la Figura 4.13. Esta interfaz es algo más compleja puesto que muestra los archivos que el usuario tiene guardados en el servidor de ficheros.

El diseño de la interfaz y la forma de utilizarla se describe en el Anexo A, donde se incluye un manual de usuario. En él se describe cómo utilizar todo el proyecto.

En lo que respecta al funcionamiento de esta página es simple. Cada vez



Figura 4.13: Interfaz página principal

que se requiere realizar una acción (subir archivo, borrar archivo, etcétera), al pulsar el botón que permite realizar dicha acción, se actualiza la página de tal forma que se captura la acción. Es decir, se utiliza un *form* para enviar datos pero hacia la misma página web en la que se encuentra. De esta forma se pueden realizar dichas acciones de manera instantánea.

Cada vez que se actualiza o se entra por primera vez en la página se comprueba en primer lugar si hay una sesión de usuario iniciada y en caso contrario se redirige a la página *index*. Si la hay, se comprueba además si hay que subir un archivo, borrarlo o renombrar y en caso contrario simplemente se muestran los archivos guardados en el servidor de ficheros. En el caso de que se deba realizar alguna de las acciones anteriores, ésta se realiza, se muestra una notificación y finalmente se muestran los archivos guardados.

Para la implementación de esta página se hace uso evidentemente de las funciones de la clase *ftp* que proporciona PHP. Por supuesto será necesario establecer una conexión segura desde el servidor web hasta el servidor *ftp*. Para ello se utiliza una función especial de PHP para establecer dicha conexión. Un breve resumen de la implementación es el siguiente:

```

1 // Establecer una conexion segura al servidor FTP
2 $ftp_puerto = 21;
3 $conn_id = ftp_ssl_connect($_SESSION['ftp_server_IP'], $ftp_puerto);
4
5 //Iniciar una sesion con nombre de usuario y contraseña
6 $login_result = ftp_login($conn_id, $_SESSION['usuario'], $_SESSION['
    password']);
7
8 //Para guardar un archivo
9 ftp_put($conn_id, $nuevo_archivo, $nombre_temporal, FTP_BINARY);

```

```
10
11 //Para borrar un archivo
12 ftp_delete($conn_id, $archivo);
13
14 //Para renombrar un archivo
15 ftp_rename($conn_id, $archivo, $nuevo_nombre)
16
17 //Para obtener los archivos
18 $lista = ftp_nlist($conn_id, "");
19
20 //Se cierra la conexion FTP en cualquier caso
21 ftp_close($conn_id);
```

Al igual que en casos anterior, éste es simplemente un breve resumen de las principales funciones que PHP tiene para poder actuar sobre *ftp*. En esta página también se manejan errores y existe además un sistema de notificaciones en función de estos.

4.2.4.4. applet.php

Esta página es la utilizada para mostrar la interfaz Java del cliente. Desde dicha interfaz se establecerá la conexión y se mandarán las peticiones al servidor situado en la Raspberry Pi.

Esta página simplemente selecciona el tipo de interfaz a mostrar en función del tipo de documento (PDF o imagen), y la aplicación cliente se encarga del resto. La forma para lanzar la aplicación es la siguiente:

```
1 <div align="middle">
2   <applet code="applet.JApplet" archive="applet.jar" width="400"
3     height="300">
4     <PARAM NAME="servidor" VALUE="<?php echo $IP_ServidorRPI;?>" />
5     <PARAM NAME="usuario" VALUE="<?php echo $_SESSION['usuario']; ?>" />
6     <PARAM NAME="password" VALUE="<?php echo $_SESSION['password']; ?>" />
7     <PARAM NAME="archivo" VALUE="<?php echo $archivo; ?>" />
8   </applet>
9 </div>
```

4.2.4.5. cambiarPassword.php

Esta página es prácticamente igual que la página para registrar un nuevo usuario con la salvedad de que solo será necesario introducir la nueva contraseña. Esta página pide una nueva contraseña que sea al menos de 8 caracteres, y que contenga mínimo un número y una letra, igual que anteriormente.

La implementación realiza sigue la misma estructura que la página registrar.php con la diferencia de que solo será necesario en este caso actualiza

la contraseña de una cuenta ya existente en lugar de crearla completamente, por lo que el proceso de implementación es muy similar.

4.2.4.6. cerrarSerion.php

Esta página es la última por la que pasa el usuario antes de terminar una sesión con el servidor web. Esta página no tiene interfaz, simplemente es un documento programado mediante lenguaje PHP que se ejecuta en el servidor web y que al terminar su ejecución, es decir, al cerrar la sesión de usuario complemente, redirige al usuario a la página de portada del servidor (*index*).

Adicionalmente a todo esto, el servidor HTTP debe cumplir también con unos requisitos de seguridad necesarios para la realización del proyecto. Por ello en primer lugar se rechaza cualquier intento de conexión a cualquier parte del servidor que no sea la página de portada cuando no existe una sesión de usuario iniciada. Como la sesión de usuario sólo se inicia cuando se realiza una correcta autenticación, de esta forma se evita que cualquier usuario pueda ver el contenido del servidor web sin tener cuenta de acceso.

Por otro lado, cualquiera puede crear una nueva cuenta en el sistema, algo que puede llegar a ser un inconveniente. Pero esto ya se ha tenido en cuenta y no se ha solucionado porque se sale fuera de los límites del proyecto. Teniendo en cuenta que en un futuro las cuentas utilizadas serán las correspondientes a la Universidad de Granada, no será necesario perfeccionar el sistema de las cuentas de los usuarios.

Por último pero no menos importante, se ha configurado el servidor HTTP como servidor HTTPS de forma que toda la comunicación vaya bajo la capa SSL. De esta forma toda la información relevante a los usuarios quedará siempre protegida. Para realizar esta implementación se ha generado un certificado y se ha firmado el entorno web creado para este proyecto.

4.2.5. Servidor RPi

Este servidor corresponde al situado en la Raspberry Pi. En la sección anterior (Sección 4.1 ya se ha descrito su diseño, el diagrama de estados correspondiente y el formato de los mensajes del protocolo de comunicación entre este servidor y la aplicación cliente. En este apartado por tanto, se profundizará en el proceso de implementación y funcionamiento de dicho servidor.

El servidor se ha implementado como una única clase Java, de forma que es lanzado por el usuario mediante la línea de comandos de la terminal de Linux. La forma de ejecutar el servidor es `java ServidorRPi [opciones] [parámetros]`, donde las opciones equivalen únicamente a “-x”. Si el servidor

4. Diseño y resolución del trabajo

se ejecuta con esta opción activada, se inicia en modo *debug* y por consiguiente se muestran en pantalla todos los mensajes intercambiados entre cliente y servidor, así como otras notificaciones o errores.

En cuanto a los parámetros se pueden distinguir “-p” y “-h”. Con el primero se puede seleccionar el puerto que el servidor abrirá para iniciar la comunicación. Con el segundo se selecciona la dirección IP del servidor de ficheros, es decir, el servidor VSFTPD. Si estos parámetros no se especifican al inicio, el servidor utiliza el puerto 8888 y la dirección IP 192.168.1.1 por defecto. En caso de que estos parámetros sí sean indicados, pero no tengan la sintaxis correcta, la ejecución se cancela sin llegar a iniciar el servidor y se muestra en pantalla un mensaje de ayuda explicando cómo se ha de lanzar el servidor. Un ejemplo de esto es:

```
> java Servidor RPi -x -p 8854 -h 192.168.3.12
```

Una vez que el servidor se ejecuta correctamente, atendiendo a la Figura 4.7, su ejecución comienza con la apertura del puerto y la posterior escucha esperando un nuevo cliente.

```
1 //Se abre el servidor en el puerto indicado
2 socketEspera = new ServerSocket(puerto);
3
4 //Cada nuevo cliente se debe cambiar al estado inicial
5 estado = estadoInicial;
6
7 //Se inicia la conexión y se espera un nuevo cliente
8 socketConexion = socketEspera.accept();
```

Esta última acción deja el servidor en modo espera o escucha y únicamente se acepta la conexión cuando se recibe una petición de conexión por parte de un cliente. Una vez que se acepta la conexión con el cliente se han de abrir los flujos de envío y recepción de mensajes, es decir, los *buffers* que se utilizarán para el intercambio de mensajes.

```
1 //Se abren los flujos de envío/recepcion de mensajes:
2 canalSalida = new PrintWriter(socketConexion.getOutputStream(), true);
3 canalEntrada = new BufferedReader(new InputStreamReader(socketConexion.
    getInputStream()));
```

Siguiendo con la implementación del servidor, una vez que se establece la comunicación cliente – servidor, el servidor entra en un bucle en el que lee constantemente las peticiones recibidas por parte del cliente y actúa en consecuencia a cada una de ellas. Si se produce un error en la recepción de la petición (generalmente por problemas en los canales de entrada o salida) se maneja el error con la función *manejarErrorSolicitud()*. Esta función actúa en consecuencia al error obtenido (generalmente con un reinicio de la conexión con el cliente).

De este bucle sólo se puede salir estableciendo el valor de la variable *cerrar = false*. Una vez que se sale del bucle se cierran todas las conexiones y finalmente se cierra por completo el servidor.

Dentro del bucle se implementa un *switch* principal en el que se determina el estado del servidor. Atendiendo al diagrama de la Figura 4.7 se han establecido tantas variables finales como estados componen el servidor. De esta forma cada vez que ocurre un evento determinado esta variable actualiza su valor en función del estado en el que se encuentre el servidor. La razón de esta implementación es porque en cada estado se puede recibir un tipo de mensajes diferente a los demás estados, de esta forma es más sencillo y eficiente manejar los mensajes leídos al inicio del bucle.

Siempre que se inicia una nueva conexión con un cliente *estado = estadoInicial*. En este estado el mensaje que se puede recibir es “USER”. Si se recibe dicho mensaje se guardan los datos referentes al usuario y se actualiza *estado = estadoIdentificado*. En caso de recibir cualquier otro mensaje se produce un error en la comunicación del protocolo y *estado = estadoError*. Esta acción es equivalente a todos los estados que componen el servidor cuando se recibe un mensaje incorrecto.

En este segundo estado, el mensaje que se debe recibir es “FTP”. Cuando se recibe el servidor se prepara para la obtención del fichero desde el servidor VSFTPD. Para ello se obtiene en primer lugar la extensión del archivo que se quiere mostrar, es decir, el tipo de archivo (PDF, JPG o PNG). Con esta extensión se crea la ruta que tendrá el fichero temporal utilizado, es decir, donde se guardará. Y por último se obtiene el fichero desde el servidor FTP.

Para la obtención del fichero, surgió un inconveniente adicional, Java no tiene implementada una clase que permita conexión mediante FTP. Ante esto la opción más sencilla es reutilizar el código que PHP de implementaciones anteriores. Por la implementación del entorno web se sabe que se puede establecer una conexión segura al servidor de ficheros sin ningún tipo de problema mediante PHP. Por tanto, la solución adquirida en este punto equivale a la creación de un fichero PHP que se utiliza únicamente para descargar el archivo indicado. Este fichero será lanzado desde la aplicación Java, la cual le pasará todos los parámetros necesarios.

```
1 //Obtener la extension del archivo para completar la
2 //ruta donde se guardará el fichero
3 extension = ftp_archivo.substring(ftp_archivo.length()-3);
4
5 //Se anade la extension al path
6 path = "/home/pi/Servidor/archivoTemporal." + extension;
7
8 int Npalabras = campos.length-1; //Variable auxiliar para el fichero
   descargar.php
9
10 //Comando para ejecutar el fichero que descarga el archivo temporal por
```

4. Diseño y resolución del trabajo

```
SFTP
11 String comando = "php -f descargar.php " + IP_ftp_server + " " +
    ftp_usuario + " " + ftp_password + " " + Npalabras + " " +
    ftp_archivo + " " + path + " " + debug;
12
13 //Ejecutar el comando
14 Process p = Runtime.getRuntime().exec(comando);
```

Éste es el proceso para obtener el archivo, a continuación se describirá la implementación de “*descargar.php*” de forma que se facilite la comprensión de la solución optada.

descargar.php

Este fichero es utilizado para descargar un archivo mediante el establecimiento de una conexión SSL hacia un servidor FTP, con unos datos de usuario y un destino de descarga. Su fundamento es igual al utilizado en el servidor HTTP.

El fichero necesita como datos de entrada, y este mismo orden; dirección IP del servidor de ficheros, nombre de usuario, contraseña, número de palabras que componen el fichero que se va a descargar, nombre de dicho fichero, la ruta donde se guardará el archivo y por último una variable auxiliar utilizada para activar el modo *debug*.

Un breve resumen de la implementación de este fichero es la siguiente:

```
1 //Establecer una conexion al servidor FTP
2 $conn_id = ftp_ssl_connect($ftp_IP_server, $ftp_puerto);
3
4 //Iniciar una sesion con nombre de usuario y contraseña
5 $login_result = ftp_login($conn_id, $ftp_usuario, $ftp_password);
6
7 //Obtener el archivo
8 ftp_get($conn_id, $path, $archivo, FTP_BINARY)
9
10 //Se cierra la conexiOn FTP en cualquier caso
11 ftp_close($conn_id);
12
13 //Finalmente se obtiene la salida del error
14 // 0 -> no hay error
15 // 1 -> parAmetros incorrectos
16 // 2 -> error de conexiOn
17 // 3 -> error de autenticaciOn
18 // 4 -> error de descarga
19 echo $error;
```

Como se puede ver, el fichero termina devolviendo una variable error de salida, la cual será cero cuando el proceso haya transcurrido de forma correcta y contendrá un tipo de error en caso contrario. Este error se maneja fuera del este fichero. Se utiliza la función implementada en el servidor Java

manejarErrorDescargar(), a la que hay que pasarle como parámetro el error resultante.

Si todo ha ido correctamente el servidor avanza *estado = estadoArchivoPreparado*. En este estado se puede recibir la petición “*ABRIR*”. Para abrir el archivo se utiliza la clase *Desktop* que proporciona Java. Con esta clase se puede abrir cualquier documento o archivo en el entorno de escritorio de la máquina que ejecuta el servidor. Hay que tener en cuenta en este caso que el programa usado para abrir los archivos es el que esté establecido por defecto en el equipo para cada tipo de extensión. Por lo que se debe establecer por defecto el visor de PDF, MuPDF, en la Raspberry Pi.

```
1 //Abrir el fichero
2 Desktop.getDesktop().open(new File(ruta));
```

Una vez que se ejecuta este comando, se espera un tiempo de 2 segundos (ya que la acción de abrir el documento no es instantánea) y se lanza una acción para poner en pantalla completa el documento.

En este paso el servidor ya ha avanzado a *estado = estadoArchivoAbierto*. Y en este estado sólo se aceptan las peticiones que han sido definidas con anterioridad en el servidor, en caso contrario se produce un error en el protocolo diseñado. Los mensajes que aquí se pueden recibir se han descrito en la Sección 4.1.4.

Las funciones *accionPDF* y *accionIMG* son las utilizadas para ejecutar las ordenes correspondiente a las peticiones recibidas desde el cliente. Estas funciones se usan según si el archivo mostrado es un documento PDF o una imagen. Ambas funciones tienen el mismo funcionamiento salvo que cambian la forma de ejecutar las órdenes recibidas. Estas funciones están constituidas por un *switch* donde se decide, en función del mensaje recibido, el código que se ejecuta en cada caso.

Para ejecutar las órdenes se hace uso de la clase *Robot* también proporcionada por java. Esta clase permite simular el teclado de la máquina donde está corriendo el servidor. De esta forma, aprovechando que el *software* utilizado puede ser controlado mediante teclado, todas las acciones corresponden a pulsaciones de teclas. Así un ejemplo de ejecución de la orden “*COMPLETA*” es:

```
1 case "COMPLETA":
2     //Ver en pantalla completa
3     teclado.keyPress(KeyEvent.VK_F);
4     teclado.keyRelease(KeyEvent.VK_F);
5     break;
```

Finalmente, cuando el servidor recibe una petición “*CERRAR*” avanza *estado = estadoFinal*. En este estado se utilizan las funciones *borrarArchivo*

voTemporal(), *cerrarConexion()* y *nuevoCliente()*. De forma que se borra el archivo que se ha estado utilizando, se cierra toda conexión existente con el cliente y finalmente vuelve a escuchar en el puerto hasta que llegue un nuevo cliente.

Por último, cabe destacar además que toda la implementación tiene un completo sistema de *debug* y log de forma que se pueda estudiar de forma sencilla qué está ocurriendo en cada momento en el servidor, y en caso de fallo, determinar dónde se encuentra y solucionarlo lo antes posible.

4.2.6. Cliente RPi

Esta aplicación corresponde a la aplicación que debe ejecutar el cliente para conectarse al servidor RPi y en definitiva hacer uso del proyecto. En este caso, como la aplicación cliente consta de interfaz gráfica (applet), se implementan dos clases diferentes. Una correspondiente a la interfaz gráfica y otra correspondiente al ClienteRPi propiamente dicho.

La primera clase, correspondiente al applet, se implementa como un JApplet. En ella se implementarán los botones y selecciones que permitirán la total manipulación del documento que se está mostrando en el proyecto. En este sentido se han diseñado dos interfaces diferentes en función del tipo de documento (PDF o imagen). El funcionamiento de ambos es el mismo, lo único que varía en este caso es el número de opciones del que dispone la interfaz. Por tanto, esta clase es muy sencilla, simplemente debe crear un objeto de la clase ClienteRPi y pasarle como entrada los parámetros procedentes del entorno web (dirección IP del servidor donde se va a mostrar, usuario, contraseña y archivo que se quiere mostrar).

```
1 //Se inicia el cliente
2 if ((servidor != null) && (usuario != null) && (password != null) && (
    archivo != null) {
3     cliente = new Cliente(servidor, puerto, usuario, password, archivo);
4 }
```

El cliente ejecuta por sí solo los mensajes obligatorios que el protocolo requiere a su inicio, como se vio en la Sección 4.1.4. El último mensaje corresponde a “ABRIR”, por lo que una vez que se establece la conexión con el servidor y se muestra el archivo en el proyector es cuando se puede usar la interfaz de la aplicación cliente.

Al realizar una petición desde la interfaz del applet, esta clase hace uso del objeto cliente creado anteriormente para enviar un mensaje con la petición deseada (la lista de posibles mensajes/peticiones se describió en la Sección 4.1.4) al servidor.

Por otro lado, la implementación de la clase ClienteRPi debe seguir la

estructura de estados descrita en la Sección 4.1.3, por lo que al enviar un mensaje al servidor RPi siempre debe esperar una respuesta de éste.

En primer lugar cuando se inicia el cliente se debe realizar la conexión con el servidor RPi:

```
1 //Inicio de la conexion con el servidor
2 socketConexion = new Socket(servidor, puerto);
3
4 //Apertura de los canales de envio/recepcion de mensajes:
5 canalSalida = new PrintWriter(socketConexion.getOutputStream(), true);
6 canalEntrada = new BufferedReader(new InputStreamReader(socketConexion.
    getInputStream()));
```

Una vez que la conexión está establecida comienza el envío de los mensajes iniciales del protocolo:

```
1 //Mandar datos de usuario al servidor
2 clienteToServidor("USER " + usuario + " " + password,
3     "Mandando datos de usuario");
4
5 //Espera a recibir la respuesta del servidor
6 esperandoServidor();
7
8 //Mandar peticiOn para descargar y preparar el archivo
9 clienteToServidor("FTP " + archivo, "Mandando datos de archivo");
10
11 //Espera a recibir la respuesta del servidor
12 esperandoServidor();
13
14 //Mandar peticiOn para abrir archivo PDF
15 clienteToServidor("ABRIR", "Mandando peticion para abrir");
16
17 //Espera a recibir la respuesta del servidor
18 esperandoServidor();
```

Nuevamente, esto es sólo una breve descripción de la implementación realizada. La función *clienteToServidor()* es la encargada de mandar los mensajes hacia el servidor RPi. Existen dos versiones en función del número de parámetros que se le pasen. Si sólo es uno, únicamente envía el mensaje al servidor. Si se trata de dos parámetros, el segundo corresponde a la información que se quiere mostrar por pantalla en la ventana del cliente, es decir, es un parámetro utilizado para *debug*.

La función *esperandoServidor()* es la utilizada para esperar a recibir el mensaje de respuesta del servidor. Esta función maneja la respuesta recibida en función de si ha ido todo correcto o se ha producido un error. En caso de error grave se cierra la conexión y se finaliza con la aplicación cliente mediante la función *cerrarCliente()*.

Estas dos funciones (*clienteToServidor()* y *esperandoServidor()*) son las utilizadas por la clase de la interfaz para enviar peticiones y recibir las respuestas. Un ejemplo de su utilización es el siguiente:

4. Diseño y resolución del trabajo

```
1 private void BotonAvanzarActionPerformed(java.awt.event.ActionEvent evt
   ) {
2 //Mandar peticion de avanzar pagina
3     cliente.clienteToServidor("AVANZAR");
4     cliente.esperandoServidor();
5 }
```

Finalmente hay que añadir que, para dar mayor información acerca de la implementación tanto del Servidor RPi como de la aplicación Cliente RPi, se adjuntan a este documento los ficheros Java con la implementación, así como la documentación *htdoc*. Con esto se pretende permitir una mayor comprensión de la implementación realizada, además de facilitar a futuros usuarios que puedan modificar y mejorar la implementación.

Capítulo 5

Evaluación y pruebas

La fase de evaluación y prueba de la implementación ha estado muy presente a lo largo de toda la realización del proyecto, puesto que ha sido un trabajo constante. Se han estado probando diferentes configuraciones y soluciones hasta llegar a la solución final y para ello ha sido necesario un periodo de pruebas muy largo.

No obstante, en este capítulo se hará un estudio y evaluación de los resultados obtenidos con la implementación final de este proyecto. En primer lugar, se hará una evaluación global haciendo uso de la herramienta Wireshark [83], mediante la cual se analizarán cuidadosamente los mensajes producidos en las comunicaciones. Después, se analizará el correcto funcionamiento en uno de los escenarios típicos de aplicación. Este escenario simulará el uso real del sistema en el ámbito de la docencia.

5.1. Análisis del tráfico

Mediante el analizador de protocolos Wireshark se puede analizar el correcto funcionamiento del sistema ante distintas situaciones. En primer lugar, se analizará el establecimiento de una conexión al servidor HTTPS. Y en segundo lugar, se revisará el protocolo creado para la comunicación cliente-servidor descrito en la Sección 4.1.4.

El escenario utilizado para el análisis consiste en un ordenador personal, donde se encuentran configurados el conjunto de servidores utilizados en este proyecto (OpenLDAP, FreeRADIUS, Apache, y VSFTPD), conectado mediante una red punto a punto con la Raspberry Pi. De forma que la dirección IP 192.168.1.1/24 corresponde con el equipo y la 192.168.1.2/24 a la Raspberry Pi (véase la Figura 5.1).

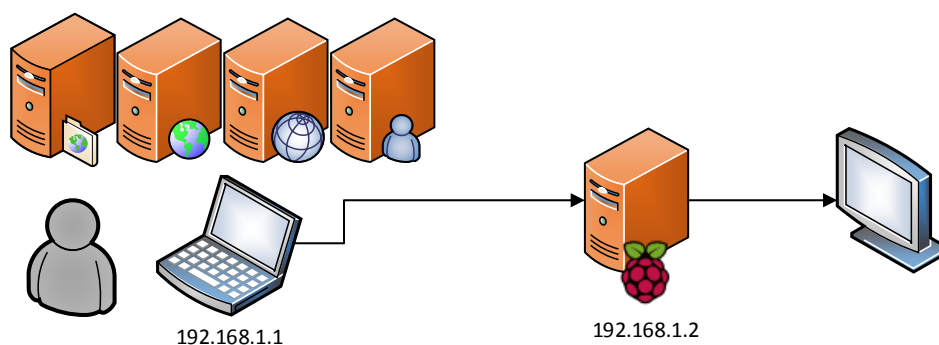


Figura 5.1: Escenario de aplicación del ejemplo

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	127.0.0.1	127.0.0.1	TCP	74	57973 > https [SYN] Seq=0 win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=40614 TSecr=
2	0.00002200	127.0.0.1	127.0.0.1	TCP	74	https > 57973 [SYN, ACK] Seq=0 Ack=1 win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=4
3	0.00004200	127.0.0.1	127.0.0.1	TCP	66	57973 > https [ACK] Seq=1 Ack=1 win=43776 Len=0 TSval=40614 TSecr=40614
4	0.00044300	127.0.0.1	127.0.0.1	TLSv1.2	242	Client Hello
5	0.00047600	127.0.0.1	127.0.0.1	TCP	66	https > 57973 [ACK] Seq=1 Ack=177 win=44800 Len=0 TSval=40614 TSecr=40614
6	0.00561500	127.0.0.1	127.0.0.1	TLSv1.2	1426	Alert (Level: warning, Description: unrecognized Name), Server Hello, Certificate,
7	0.00565300	127.0.0.1	127.0.0.1	TCP	66	57973 > https [ACK] Seq=177 Ack=1361 win=174720 Len=0 TSval=40615 TSecr=40615
8	0.08928500	127.0.0.1	127.0.0.1	TLSv1.2	192	client Key Exchange, change cipher Spec, Hello Request, Hello Request
9	0.09009600	127.0.0.1	127.0.0.1	TLSv1.2	340	New Session Ticket, change cipher Spec, Encrypted Handshake Message
10	0.09012100	127.0.0.1	127.0.0.1	TCP	66	57973 > https [ACK] Seq=303 Ack=1635 win=177408 Len=0 TSval=40636 TSecr=40636
11	0.13627700	127.0.0.1	127.0.0.1	TLSv1.2	412	Application Data
12	0.13775100	127.0.0.1	127.0.0.1	TLSv1.2	1542	Application Data, Application Data, Application Data
13	0.13780000	127.0.0.1	127.0.0.1	TCP	66	57973 > https [ACK] Seq=649 Ack=3111 win=308352 Len=0 TSval=40648 TSecr=40648
14	0.18164700	127.0.0.1	127.0.0.1	TLSv1.2	484	Application Data
15	0.18220500	127.0.0.1	127.0.0.1	TLSv1.2	16798	Application Data, Application Data

Figura 5.2: Captura de Wireshark del establecimiento de la conexión

5.1.1. Entorno web

Para este caso se ha realizado una captura del tráfico global mientras se navegaba sobre el entorno web. Como se puede ver en la Figura 5.2 lo primero que se realiza es la conexión mediante TLS versión 1.2. En la imagen se puede observar el intercambio de mensajes hasta llegar a la correcta conexión con el servidor web.

El mensaje número 6 corresponde con una alerta que se muestra al usuario en el navegador sobre el certificado. Esta alerta se produce porque la entidad certificadora que emite el certificado es desconocida. Esto es de esperar puesto que se trata de un servidor HTTP autofirmado y por tanto el certificado no ha sido creado por una entidad certificadora reconocida.

Si se profundiza en este mensaje se pueden obtener los datos correspondientes a dicho certificado, como puede observarse en la Figura 5.3.

El siguiente paso es la autenticación del usuario. Para ello se realiza una conexión entre servidor HTTP y servidor RADIUS, y éste a su vez al servidor LDAP. Los mensajes correspondientes a la autenticación se muestran en la Figura 5.4.

Si se observa la imagen, se pueden ver los mensajes intercambiados, tan-

5. Evaluación y pruebas

```
Secure Sockets Layer
  TLSv1.2 Record Layer: Alert (Level: warning, Description: Unrecognized Name)
  TLSv1.2 Record Layer: Handshake Protocol: Server Hello
  TLSv1.2 Record Layer: Handshake Protocol: Certificate
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 931
  Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 927
    Certificates Length: 924
  Certificates (924 bytes)
    Certificate Length: 921
    Certificate (pkcs-9-at-emailAddress=JR,id-at-commonName=JR,id-at-organizationalUnitName=JR,
      signedCertificate
        algorithmIdentifier (sha256withRSAEncryption)
          Algorithm Id: 1.2.840.113549.1.1.11 (sha256withRSAEncryption)
          Padding: 0
          encrypted: 5823d7d3f1bdc585964720c718a931826ef787fd56e43b6c...
  TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
  TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done
```

Figura 5.3: Captura de Wireshark del mensaje Alert

No.	Time	Source	Destination	Protocol	Length	Info
175	4.99958100	127.0.0.1	127.0.0.1	RADIUS	87	Access-Request(1) (id=106, l=45)
176	4.99986300	127.0.0.1	127.0.0.1	LDAP	1060	searchRequest(3) "dc=ugr,dc=es" wholesubtree
177	5.00040300	127.0.0.1	127.0.0.1	LDAP	98	searchResEntry(3) "cn=admin,dc=ugr,dc=es"
178	5.00050900	127.0.0.1	127.0.0.1	LDAP	80	searchResDone(3) success [1 result]
179	5.00056600	127.0.0.1	127.0.0.1	TCP	66	44988 > 1dap [ACK] Seq=995 Ack=47 win=342 Len=0 TSval=41864 TSecr=41864
180	5.00081500	127.0.0.1	127.0.0.1	TCP	74	45006 > 1dap [SYN] Seq=0 win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=41864 TSecr=41864
181	5.00083200	127.0.0.1	127.0.0.1	TCP	74	1dap > 45006 [SYN, ACK] Seq=0 Ack=1 win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=41864 TSecr=41864
182	5.00084900	127.0.0.1	127.0.0.1	TCP	66	45006 > 1dap [ACK] Seq=1 Ack=1 win=43776 Len=0 TSval=41864 TSecr=41864
183	5.00089100	127.0.0.1	127.0.0.1	LDAP	106	bindRequest(1) "cn=admin,dc=ugr,dc=es" simple
184	5.00090600	127.0.0.1	127.0.0.1	TCP	66	1dap > 45006 [ACK] Seq=1 Ack=41 win=43776 Len=0 TSval=41864 TSecr=41864
185	5.00115300	127.0.0.1	127.0.0.1	LDAP	80	bindResponse(1) success
186	5.00117000	127.0.0.1	127.0.0.1	TCP	66	45006 > 1dap [ACK] Seq=41 Ack=15 win=43776 Len=0 TSval=41864 TSecr=41864
187	5.00121600	127.0.0.1	127.0.0.1	LDAP	73	unbindRequest(2)
188	5.00124000	127.0.0.1	127.0.0.1	TCP	66	45006 > 1dap [FIN, ACK] Seq=48 Ack=15 win=43776 Len=0 TSval=41864 TSecr=41864
189	5.00128500	127.0.0.1	127.0.0.1	TCP	66	1dap > 45006 [FIN, ACK] Seq=15 Ack=49 win=43776 Len=0 TSval=41864 TSecr=41864
190	5.00129800	127.0.0.1	127.0.0.1	RADIUS	62	Access-Accept(2) (id=106, l=20)

Figura 5.4: Captura de Wireshark de la autenticación de usuario



```

    RADIUS Protocol
      Code: Access-Request (1)
      Packet identifier: 0x6a (106)
      Length: 45
      Authenticator: bb584446c9375fdb68fed18fe9fe1911
      [The response to this request is in frame 190]
      Attribute Value Pairs
        AVP: 1=7  t=User-Name(1): admin
          User-Name: admin
        AVP: 1=18  t=User-Password(2): Encrypted
          User-Password (encrypted): fad37cb11f7ca6588946796470295e92

```

Figura 5.5: Captura de Wireshark del protocolo RADIUS

tos los pertenecientes al protocolo RADIUS, como los pertenecientes al protocolo LDAP. En este sentido, el servidor RADIUS pasa instantáneamente la petición a su base de datos. En la Figura 5.5 se observa el mensaje correspondiente a dicho protocolo. En él se puede ver como la información correspondiente a la contraseña del usuario se transmite de forma cifrada.

En el intercambio de mensajes anterior se observa como el servidor RADIUS realiza una consulta LDAP para comprobar los datos. En los mensajes 180 y 181 se produce la conexión con el servidor LDAP. En los posteriores mensajes se realiza la búsqueda de la entrada y se comprueba que los datos son correctos. Finalmente en los mensajes 188 y 189 se cierra la conexión con el servidor LDAP. Y el mensaje 190 es el mensaje enviado desde el servidor RADIUS de vuelta al servidor HTTP indicando el resultado de la autenticación. Instantáneamente a esto se cierra la conexión con el servidor RADIUS.

Se observa también a simple vista cómo la contraseña de cada usuario viaja en todo momento de forma oculta, encriptada, pero en cambio el nombre del usuario viaja en texto plano. Este inconveniente no tiene importancia puesto que el nombre de usuario corresponde con el nombre utilizado por el correo electrónico de la Universidad de Granada, es decir, que es un nombre de usuario público. Lo importante en este caso es que la contraseña viaje encriptada.

El siguiente intercambio de mensajes corresponde con la conexión al servidor de ficheros VSFTPD, puesto que en la página principal del usuario se mostrarán los archivos que tiene guardados en dicho servidor. Este intercambio de mensajes se puede ver en la Figura 5.6.

En este caso, los mensajes 195 y 196 corresponden al establecimiento de la conexión entre el servidor HTTP y el servidor VSFTPD. Después de este instante se aprecia en el intercambio de mensaje como se requiere obligatoriamente comunicación mediante la capa TLS y a partir de este momento todos los mensajes intercambiados viajan encriptados, tanto los datos correspondientes al usuario como los datos correspondientes a los archivos guardados.

[illegible]

Figura 5.6: Captura de Wireshark de la conexión FTP

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	192.168.1.1	192.168.1.2	TCP	74	47600 > ddi-tcp-1 [PSH, ACK] Seq=1 Ack=1 win=229
2	0.00000000	192.168.1.1	192.168.1.2	TCP	76	[TCP Retransmission] 47600 > ddi-tcp-1 [PSH, ACK]
3	0.03968300	192.168.1.2	192.168.1.1	TCP	68	ddi-tcp-1 > 47600 [ACK] Seq=1 Ack=9 win=453 Len=0
4	0.04252400	192.168.1.2	192.168.1.1	TCP	72	ddi-tcp-1 > 47600 [PSH, ACK] Seq=1 Ack=9 win=453
5	0.04260500	192.168.1.1	192.168.1.2	TCP	68	47600 > ddi-tcp-1 [ACK] Seq=9 Ack=5 win=229 Len=0
6	0.03968300	192.168.1.2	192.168.1.1	TCP	66	ddi-tcp-1 > 47600 [ACK] Seq=1 Ack=5 win=453 Len=0
7	0.04252400	192.168.1.2	192.168.1.1	TCP	70	[TCP Retransmission] ddi-tcp-1 > 47600 [PSH, ACK]
8	0.04260500	192.168.1.1	192.168.1.2	TCP	66	[TCP Dup ACK #1] 47600 > ddi-tcp-1 [ACK] Seq=9 A

Figura 5.7: Captura de Wireshark del intercambio de mensajes en la conexión

Con todo este proceso queda completado el intercambio de mensajes referente a una conexión normal de usuario al servicio web ofrecido por este proyecto.

5.1.2. Entorno de la aplicación cliente

Para el análisis de este apartado, se obtienen las capturas referentes a la ejecución de la aplicación cliente, la cual se encuentra integrada en el entorno web. Para este caso se diferenciarán tres etapas, el momento del establecimiento de la conexión con el servidor, la fase en la que se mandan las peticiones y por último la fase de cierre de la conexión. El intercambio de estos mensajes puede verse en la Figura 5.7, Figura 5.8 y Figura 5.9 respectivamente.

En las imágenes se puede ver cómo se realiza el intercambio de mensajes a través del puerto TCP. En la información de los mensajes el alias “*ddi-tcp-1*” corresponde al puerto 8888, puerto utilizado por defecto por el servidor RPi. Además, se sabe que los mensajes que contienen la información corresponden con los mensajes que tiene en su descripción “/*PSH*”. Así, teniendo presente

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	192.168.1.1	192.168.1.2	TCP	74	47651 > ddi-tcp-1 [PSH, ACK] Seq=1 Ack=1 win=
2	0.00093100	192.168.1.2	192.168.1.1	TCP	66	ddi-tcp-1 > 47651 [ACK] Seq=1 Ack=9 win=453
3	0.02620200	192.168.1.2	192.168.1.1	TCP	70	ddi-tcp-1 > 47651 [PSH, ACK] Seq=1 Ack=9 win=
4	0.02630500	192.168.1.1	192.168.1.2	TCP	66	47651 > ddi-tcp-1 [ACK] Seq=9 Ack=5 win=229

Figura 5.8: Captura de Wireshark del intercambio de mensajes en una petición

5.2. Escenario típico de aplicación

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	192.168.1.1	192.168.1.2	TCP	75	47608 > ddi-tcp-1 [PSH, ACK] Seq=1 Ack=1 wi
2	0.02470600	192.168.1.2	192.168.1.1	TCP	72	ddi-tcp-1 > 47608 [PSH, ACK] Seq=1 Ack=8 wi
3	0.02477100	192.168.1.1	192.168.1.2	TCP	68	47608 > ddi-tcp-1 [ACK] Seq=8 Ack=5 win=229
4	0.02507500	192.168.1.1	192.168.1.2	TCP	72	47608 > ddi-tcp-1 [PSH, ACK] Seq=8 Ack=5 wi
5	0.02559100	192.168.1.1	192.168.1.2	TCP	68	47608 > ddi-tcp-1 [FIN, ACK] Seq=12 Ack=5 w
6	0.02699300	192.168.1.2	192.168.1.1	TCP	72	ddi-tcp-1 > 47608 [PSH, ACK] Seq=5 Ack=13 w
7	0.02705400	192.168.1.1	192.168.1.2	TCP	56	47608 > ddi-tcp-1 [RST] Seq=13 win=0 Len=0

Figura 5.9: Captura de Wireshark del intercambio de mensajes en el cierre de la conexión

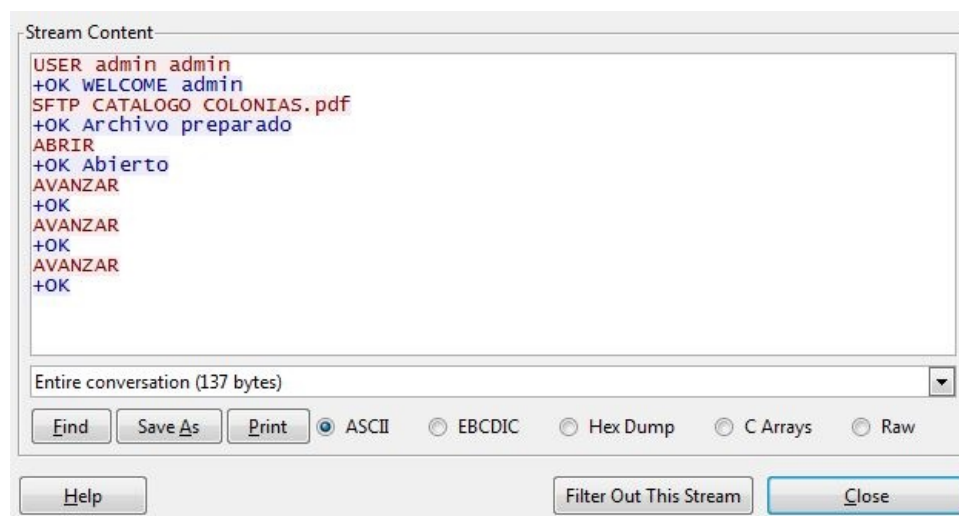


Figura 5.10: Captura de Wireshark del seguimiento del stream TCP

el diagrama de intercambio de mensajes de la Figura 4.9 es fácil darse cuenta, que por ejemplo en la Figura 5.8, el mensaje número 1 corresponde a la petición realizada por el cliente (e.g. avanzar) y el mensaje 3 equivale a la respuesta del servidor, en este caso “+OK”. En la Figura 5.10 puede verse un ejemplo del intercambio real de mensajes que se produce entre ambos.

5.2. Escenario típico de aplicación

El escenario típico de aplicación se compone de una Raspberry Pi, con el servidor RPi corriendo, conectada mediante la interfaz HDMI a un proyector o, como en este caso, a un monitor. Uno o varios servidores con Apache, VSFTPD, OpenLDAP y FreeRADIUS. En este caso todos los servidores están implementados en el ordenador personal, un portátil. Y por último una máquina cliente, en este caso otro portátil de uso personal. Todos los dispositivos deben estar conectados a la misma red local de forma similar a la Figura 5.11.

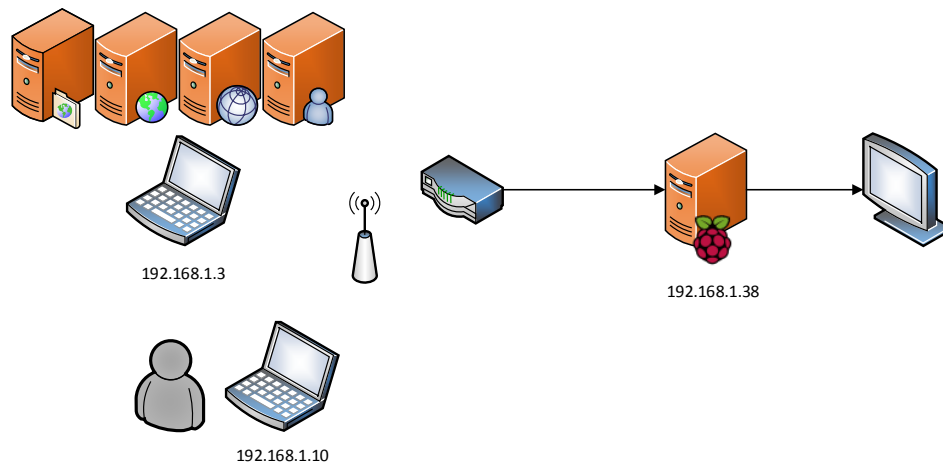


Figura 5.11: Escenario típico de aplicación

Con este escenario se pretende estudiar el funcionamiento y que tendrá el sistema en un entorno real de aplicación. De esta forma se pretenden solucionar los defectos o fallos que se detecten durante el proceso de prueba.

Para ello se realizará una evaluación de todas las posibilidades que ofrece el proyecto. Además paralelamente a esto, se realizará un análisis del tráfico en el ordenador personal (donde se sitúan los servidores) de forma que ayude a comprender mejor el comportamiento de la red mientras se está utilizando el sistema.

Lo primero que se debe hacer es ejecutar el servidor RPi en la Raspberry Pi. Como la dirección por defecto del servidor de ficheros es 192.168.1.1 habrá que modificar este parámetro. También se activará el modo *debug* para que el servidor muestre los mensajes por pantalla y saber que ésta sucediendo en cada momento. El puerto no importa. Por tanto, la forma para ejecutar el servidor RPi en este caso es:

```
> java ServidorRPi -x -h 192.168.1.3
```

El resultado puede verse en la Figura 5.12, el servidor esperando una conexión de un cliente. En lo que respecta a los demás servidores, éstos se inician de forma automática con el arranque del ordenador por lo que no es necesario iniciarlos. Así que todo está preparado para comenzar con la fase de pruebas.

El portátil con la dirección IP 192.168.1.10, en adelante PC2, será el encargado de inicial las conexiones. En primer lugar debe acceder al entorno web, y autenticarse con sus datos de usuario. Para ello es necesario iniciar un navegador web con <https://192.168.1.3/SPMD/>.

5.2. Escenario típico de aplicación

```
Archivo Edición Pestañas Ayuda
root@raspberrypi:/home/pi/Servidor# java ServidorRPI -x -h 192.168.1.3
[ServidorRPI]: Modo debug activado
[ServidorRPI]: Iniciando en el puerto 8888... OK!
[ServidorRPI]: Esperando nuevo cliente... □
```

Figura 5.12: Interfaz del servidor RPi

No.	Time	Source	Destination	Protocol	Length	Info
55	0.31001700	192.168.1.10	192.168.1.3	TCP	68	51145 > https [SYN] Seq=0 win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
56	0.31005500	192.168.1.3	192.168.1.10	TCP	68	https > 51145 [SYN, ACK] Seq=0 Ack=1 win=29200 Len=0 MSS=1460 SACK
57	0.31303000	192.168.1.10	192.168.1.3	TCP	56	51145 > https [ACK] Seq=1 Ack=1 win=17520 Len=0
58	0.31440100	192.168.1.10	192.168.1.3	TLSv1.2	573	client Hello
59	0.31444300	192.168.1.3	192.168.1.10	TCP	56	https > 51145 [ACK] Seq=1 Ack=518 win=30336 Len=0
60	0.31506300	192.168.1.3	192.168.1.10	TLSv1.2	193	server Hello, change Cipher spec, Encrypted Handshake Message
61	0.32202200	192.168.1.10	192.168.1.3	TLSv1.2	107	change Cipher spec, Hello Request, Hello Request
62	0.33059400	192.168.1.10	192.168.1.3	TCP	56	51145 > https [FIN, ACK] Seq=569 Ack=138 win=17380 Len=0
63	0.33072300	192.168.1.3	192.168.1.10	TCP	56	https > 51145 [FIN, ACK] Seq=138 Ack=570 win=30336 Len=0
64	0.33406600	192.168.1.10	192.168.1.3	TCP	56	51145 > https [ACK] Seq=570 Ack=139 win=17380 Len=0
65	0.43873000	192.168.1.10	192.168.1.3	TCP	68	51146 > https [SYN] Seq=0 win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
66	0.43875400	192.168.1.3	192.168.1.10	TCP	68	https > 51146 [SYN, ACK] Seq=0 Ack=1 win=29200 Len=0 MSS=1460 SACK
67	0.44513800	192.168.1.10	192.168.1.3	TCP	56	51146 > https [ACK] Seq=1 Ack=1 win=17520 Len=0

Figura 5.13: Captura de Wireshark del inicio de una conexión en un entorno de aplicación real

Las trazas obtenidas para este caso no difieren del caso anterior (véase la Figura 5.13). Se establece una conexión HTTPS en el puerto 443 y se transmiten los datos necesarios.

A la hora de la autenticación mediante el protocolo RADIUS y la obtención de la lista de ficheros con el protocolo FTP, como se puede ver en la Figura 5.14, los datos no salen del entorno de los servidores. Esto es, como todos los servidores están implementados en un único ordenador personal, en adelante PC1, los datos de usuario una vez que son mandados al servidor Apache no salen de dicho ordenador, por lo que resultaría imposible que alguien pueda interceptar dicha información si no es en el propio PC1.

Una vez dentro del entorno web todas las acciones corresponden al manejo del material docente en el servidor de ficheros, o a la configuración de la cuenta de usuario mediante el cambio de contraseña. Por tanto, las trazas que se obtienen en este sentido son prácticamente iguales a las de la figura anterior (Figura 5.14). La conexión es siempre entre el propio PC1 con los servidores VSFTPD y OpenLDAP.

El único caso diferente es el que se estudia a continuación, cuando el cliente (PC2) ejecuta la aplicación cliente del entorno web para conectarse al servidor alojado en la Raspberry Pi, mostrando así finalmente las proyecciones en el monitor.

En este caso, como la aplicación realiza una conexión *socket* entre cliente y servidor, los datos no pasan por el PC1, sino que se establece una comunicación punto a punto entre el PC2 y la Raspberry Pi. Aunque por otro lado, es necesario que el servidor RPi se conecte al servidor VSFTPD para

5. Evaluación y pruebas

No.	Time	Source	Destination	Protocol	Length	Info
28	2.59150500	127.0.0.1	127.0.0.1	RADIUS	89	Access-Request(1) (id=13, l=45)
32	2.59286000	127.0.0.1	127.0.0.1	LDAP	82	bindRequest(1) "<ROOT>" simple
34	2.60721500	127.0.0.1	127.0.0.1	LDAP	82	bindResponse(1) success
36	2.60737300	127.0.0.1	127.0.0.1	LDAP	1062	searchRequest(2) "dc=ugr,dc=es" wholeSubtree
37	2.62256900	127.0.0.1	127.0.0.1	LDAP	100	searchResEntry(2) "cn=admin,dc=ugr,dc=es"
38	2.62286500	127.0.0.1	127.0.0.1	LDAP	82	searchResDone(2) success [1 result]
43	2.62348800	127.0.0.1	127.0.0.1	LDAP	108	bindRequest(1) "cn=admin,dc=ugr,dc=es" simple
45	2.62399800	127.0.0.1	127.0.0.1	LDAP	82	bindResponse(1) success
47	2.62414500	127.0.0.1	127.0.0.1	LDAP	75	unbindRequest(2)
49	2.62432800	127.0.0.1	127.0.0.1	RADIUS	64	Access-Accept(2) (id=13, l=20)
64	2.75609100	127.0.0.1	127.0.0.1	FTP	88	Response: 220 (vsFTPd 3.0.2)
66	2.75620600	127.0.0.1	127.0.0.1	FTP	78	Request: AUTH TLS
68	2.75632000	127.0.0.1	127.0.0.1	FTP	99	Response: 234 Proceed with negotiation.
69	2.75660400	127.0.0.1	127.0.0.1	FTP	385	Request: \026\003\001\0018\001\000\0014\003\00
70	2.75701000	127.0.0.1	127.0.0.1	FTP	861	Response: \026\003\003\000\002\000\0006\003\0
71	2.75762900	127.0.0.1	127.0.0.1	FTP	278	Request: \026\003\003\000\000\000\000\003\000\0
72	2.76034600	127.0.0.1	127.0.0.1	FTP	302	Response: \026\003\003\000\000\0252\004\000\000\046\0
73	2.76070200	127.0.0.1	127.0.0.1	FTP	113	Request: \027\003\003\000\000\0236\0202

Figura 5.14: Captura de Wireshark de autenticación de usuario en un entorno de aplicación real

No.	Time	Source	Destination	Protocol	Length	Info
191	24.2179390	192.168.1.38	192.168.1.3	TCP	76	44863 > ftp [SYN] Seq=0 win=29200 Len=0 MSS=1460
192	24.2179920	192.168.1.3	192.168.1.38	TCP	76	ftp > 44863 [SYN, ACK] Seq=0 Ack=1 win=28960 Len=0
193	24.2199050	192.168.1.38	192.168.1.3	TCP	68	44863 > ftp [ACK] Seq=1 Ack=1 win=29248 Len=0 Ts
194	24.2237750	192.168.1.3	192.168.1.38	FTP	88	Response: 220 (vsFTPd 3.0.2)
195	24.2255060	192.168.1.38	192.168.1.3	TCP	68	44863 > ftp [ACK] Seq=1 Ack=21 win=29248 Len=0
196	24.2287320	192.168.1.38	192.168.1.3	FTP	78	Request: AUTH TLS
197	24.2287690	192.168.1.3	192.168.1.38	TCP	68	ftp > 44863 [ACK] Seq=21 Ack=11 win=29056 Len=0
198	24.2289830	192.168.1.3	192.168.1.38	FTP	99	Response: 234 Proceed with negotiation.
199	24.2360410	192.168.1.38	192.168.1.3	FTP	388	Request: \026\003\001\001;\001\000\0017\003\003s
200	24.2365620	192.168.1.3	192.168.1.38	FTP	861	Response: \026\003\003\000\002\000\0006\003\00s
201	24.2521020	192.168.1.38	192.168.1.3	FTP	278	Request: \026\003\003\000\000\000\000\003\000\00
202	24.2538960	192.168.1.3	192.168.1.38	FTP	302	Response: \026\003\003\000\000\0252\004\000\000\046\0
203	24.2619520	192.168.1.38	192.168.1.3	FTP	113	Request: \027\003\003\000\000\0255\0245

Figura 5.15: Captura de Wireshark de una conexión FTP en un entorno de aplicación real

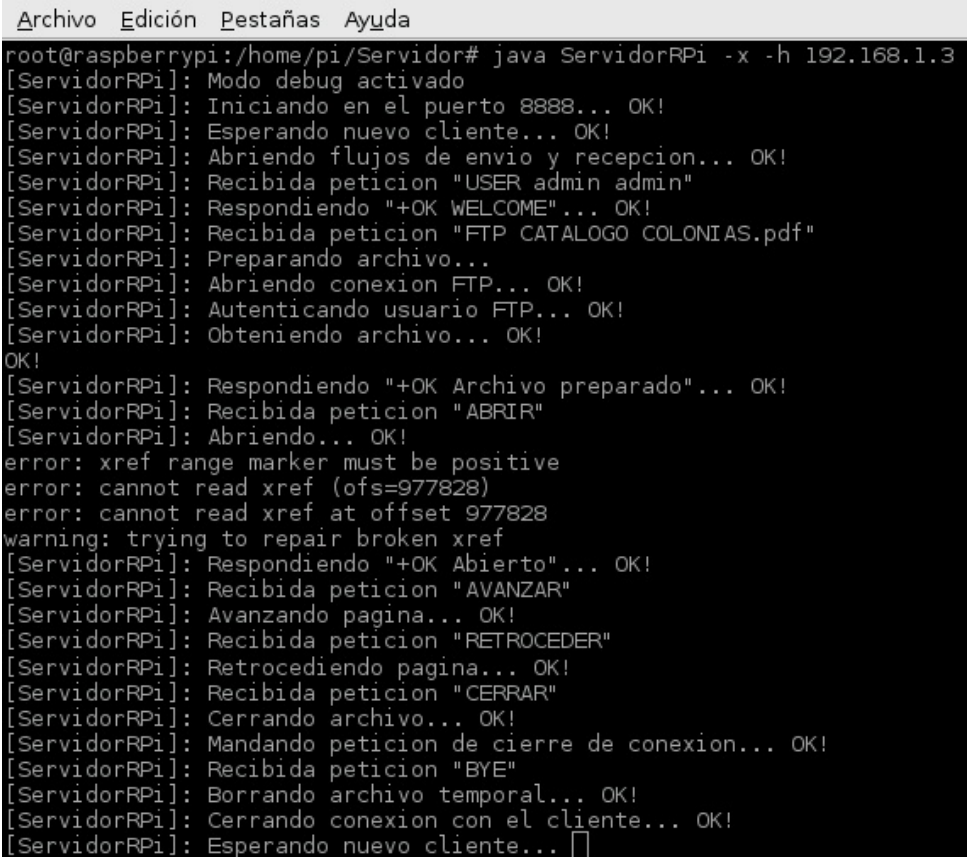
descargar el fichero que se quiere mostrar y guardarlo de forma local. Como se explicó en capítulos anteriores, esta conexión se realiza mediante PHP en la capa SSL por lo que toda la información viaja cifrada, como puede verse en la Figura 5.15.

Por otro lado, los mensajes intercambiados entre PC2 y Raspbery Pi son los mismos que se obtuvieron en las Figuras 5.7, 5.8 y 5.9, por lo que se evita añadir estas trazas.

Por otra parte, en lo que respecta al funcionamiento del sistema, se puede ver en la Figura 5.16, donde se muestra el *debug* del servidor RPi, como todo ha funcionado de forma correcta. En la imagen se pueden apreciar los mensajes intercambiados por cliente y servidor, correspondientes al protocolo diseñado y explicado en la sección 4.1.4.

Otra muestra del correcto funcionamiento es el contenido del fichero *log* resultante de la ejecución del servidor. Dicho contenido puede verse en la Figura 5.17.

Se puede concluir por tanto este capítulo afirmando que la implementa-



```
Archivo Edición Pestañas Ayuda
root@raspberrypi:/home/pi/Servidor# java ServidorRPI -x -h 192.168.1.3
[ServidorRPI]: Modo debug activado
[ServidorRPI]: Iniciando en el puerto 8888... OK!
[ServidorRPI]: Esperando nuevo cliente... OK!
[ServidorRPI]: Abriendo flujos de envio y recepcion... OK!
[ServidorRPI]: Recibida peticion "USER admin admin"
[ServidorRPI]: Respondiendo "+OK WELCOME"... OK!
[ServidorRPI]: Recibida peticion "FTP CATALOGO COLONIAS.pdf"
[ServidorRPI]: Preparando archivo...
[ServidorRPI]: Abriendo conexion FTP... OK!
[ServidorRPI]: Autenticando usuario FTP... OK!
[ServidorRPI]: Obteniendo archivo... OK!
OK!
[ServidorRPI]: Respondiendo "+OK Archivo preparado"... OK!
[ServidorRPI]: Recibida peticion "ABRIR"
[ServidorRPI]: Abriendo... OK!
error: xref range marker must be positive
error: cannot read xref (ofs=977828)
error: cannot read xref at offset 977828
warning: trying to repair broken xref
[ServidorRPI]: Respondiendo "+OK Abierto"... OK!
[ServidorRPI]: Recibida peticion "AVANZAR"
[ServidorRPI]: Avanzando pagina... OK!
[ServidorRPI]: Recibida peticion "RETROCEDER"
[ServidorRPI]: Retrocediendo pagina... OK!
[ServidorRPI]: Recibida peticion "CERRAR"
[ServidorRPI]: Cerrando archivo... OK!
[ServidorRPI]: Mandando peticion de cierre de conexion... OK!
[ServidorRPI]: Recibida peticion "BYE"
[ServidorRPI]: Borrando archivo temporal... OK!
[ServidorRPI]: Cerrando conexion con el cliente... OK!
[ServidorRPI]: Esperando nuevo cliente... □
```

Figura 5.16: *Debug* del servidor RPi en un entorno de aplicación real

ción realizada está lista para ser instalada y usada en un entorno real como pueden ser las aulas de la Universidad de Granada.

Este es el fichero de mensajes logs del Servidor RPi:

```
[29/06/2014 17:41:51] Iniciado Servidor RPi
[29/06/2014 17:42:36] Conexion con cliente: /192.168.1.10
[29/06/2014 17:42:36] Usuario: admin
[29/06/2014 17:42:38] Conexion con el servidor FTP exitosa
[29/06/2014 17:42:41] Controlando el archivo: CATALOGO COLONIAS.pdf, el usuario: admin
[29/06/2014 17:45:14] Cerrada conexion con el cliene.
[29/06/2014 17:58:56] Iniciado Servidor RPi
```

Figura 5.17: Contenido del fichero *log* del servidor RPi en un entorno de aplicación real

Capítulo 6

Conclusiones y líneas futuras

En este capítulo se detallarán los resultados obtenidos en el proyecto, analizando las aportaciones de éste y el cumplimiento de los objetivos y requisitos establecidos en el capítulo 3. También se describen las líneas de trabajo futuras con las que se podría continuar trabajando sobre los resultados obtenidos en este trabajo.

6.1. Resultados

En este proyecto se ha diseñado e implementado un sistema de proyección de material docente que aporta una solución diferente al sistema utilizado actualmente. Con esta solución se ha pretendido solucionar la problemática asociada a dicho sistema (véase en el capítulo 1).

Como se ha visto en el capítulo anterior (capítulo 5), el resultado final de la fase de evaluación y pruebas ha sido muy satisfactorio. Así, las contribuciones más destacables son las siguientes:

- El sistema implementado permite la proyección de material docente sin la necesidad de utilizar un cable para la conexión con el proyector, haciendo uso de una red inalámbrica. Para el caso, la red Wi-Fi de la Universidad de Granada.
- Como consecuencia del punto anterior, este nuevo sistema permite la total movilidad del docente en toda el aula de enseñanza mientras se proyecta el material.
- El sistema implementa además un gestor de almacenamiento centralizado, donde cada docente podrá guardar el material que quiera utilizar

en sus clases. De esta forma, se facilita el trabajo del docente con diferentes equipos, puesto que elimina la necesidad de tener guardado el archivo en el equipo utilizado.

- Para la gestión del sistema de almacenamiento anterior, se ha implementado un sistema gestor de base de datos donde se almacenarán todos los datos relevantes a las cuentas de los usuarios.
- Para la autenticación de usuarios, el sistema se ha implementado y adaptado al sistema actualmente utilizado por la Universidad de Granada en sus servicios web. De esta forma se allana el camino de implantación del sistema en el ámbito docente de la universidad.
- El sistema ofrece la posibilidad de trabajar con material docente en formato PDF, o con imágenes en formato JPG y PNG. Además es fácilmente ampliable a otros formatos y tipos de documentos diferentes. También se permite de forma sencilla la ampliación de las funcionalidades de los mismos.
- El sistema implementado permite la reutilización de los proyectores actualmente instalados. Como se viene comentando en capítulos anteriores, se utiliza una Raspberry Pi por clase como único gasto nuevo para la implantación (además del servidor centralizado). Esto supone que el sistema final sea una solución muy económica [49], de forma que se puede considerar seriamente la posibilidad de su implantación en las diferentes aulas de la universidad.
- Además, gracias también a la Raspberry Pi, se consigue que el sistema sea de fácil transporte y de fácil instalación (véase el Anexo B). Ésto permite que el sistema no sea una instalación permanente, aumentando así su usabilidad en diferentes escenarios.
- Finalmente, el sistema no ha conseguido unificar las tecnologías actuales, de forma que sea indiferente el uso de ordenador, tableta o *smartphone* para proyectar el material, que era uno de los objetivos secundarios de este proyecto. Pero sí se ha conseguido un sistema fácilmente portable a diferentes tecnologías y sistemas operativos que, en este sentido, era el objetivo principal. De esta forma se pretende que en un futuro se pueda extender y utilizar en diversos entornos diferentes.

En definitiva, el sistema implementado aporta una solución real a la problemática actual, y cubre las funcionalidades de un usuario del sistema de proyección docente actualmente utilizado.

6.2. Líneas de trabajo futuras

Aunque el proyecto ha logrado cumplir con la gran mayoría de los objetivos establecidos al inicio del mismo, tanto principales como secundarios, aún quedan abiertas las siguientes líneas de trabajo futuras:

- Se ha logrado que el sistema sea fácilmente portable a distintos dispositivos gracias al uso del lenguaje PHP en el entorno web y el lenguaje Java en el entorno de la aplicación cliente-servidor. Por tanto, una línea de trabajo clara es completar la expansión del sistema a los distintos dispositivos que actualmente nos rodean como pueden ser tabletas o *smarthphones*.
- Con las nuevas distribuciones y las nuevas versiones recientes implementadas y optimizadas para Raspberry Pi, se pueden estudiar las características de cada una de ellas para intentar mejorar de alguna forma el sistema implementado actualmente. Esto puede ser respecto al rendimiento o respecto al incremento de las funcionalidades del mismo con nuevo *software*. Recuérdese, como ya se comentó en el capítulo 4, que en el momento de comenzar con este proyecto se eligió RASPBIAN [56] porque era la más usada por la comunidad y para la que mayor número de programas existían. Pero puede que las demás distribuciones hayan aumentado sus funcionalidades a lo largo de este periodo, y sería bueno realizar un nuevo estudio de las posibilidades.
- Se puede incorporar un sistema que permita mostrar y visualizar en el proyector la imagen de la pantalla del dispositivo utilizado, característica denominada *screen mirroring*. Para ello se puede hacer uso de la tecnología Piracast [84], basada en Miracast [30] pero optimizada para Raspberry Pi. Esta tecnología aún se encuentra en fase beta, por lo que aún queda un tiempo para que esté totalmente terminada y optimizada para poder hacer uso de ella. Por esa razón se descartó la implementación de la misma en este proyecto después de haber realizado un estudio en detalle de ella.
- Por último, el sistema diseñado se ha concebido para su implantación de forma fácil en el ámbito de la Universidad de Granada. Pero para un futuro en el que se quiera instalar en otro ámbito fuera de ésta, será necesario por un lado, mejorar el sistema de administración de cuentas de usuarios, aumentando las funcionalidades. Y por otro lado, modificar el sistema de comunicación entre la aplicación cliente y el servidor situado en la Raspberry Pi, ya que ésta comunicación se realiza vía Wi-Fi aprovechando la seguridad que proporciona la red de la universidad.

6.3. Valoración personal

En este apartado se realizará una valoración autocrítica en lo que a la culminación del proyecto se refiere. A modo de valoración personal, existe un alto grado de satisfacción con el trabajo realizado a lo largo de este año. Se ha logrado completar con éxito todos los objetivos principales marcados al inicio del proyecto. Además, también se ha logrado implementar la mayoría de los objetivos secundarios, logrando así finalmente un sistema de proyección de material docente muy completo, con todas las funcionalidades necesarias para que el docente pueda realizar su trabajo sin problemas, y que logra salvar la problemática asociada al sistema actual.

Además de los objetivos marcados en cuanto a funcionalidad del proyecto, también se puede afirmar que se han cumplido los requisitos establecidos en el capítulo 3. El cumplimiento de estos objetivos y requisitos convierte finalmente este proyecto en un sistema de fácil incorporación a un mercado real. Y en especial, facilita su implantación en el entorno real de la Universidad de Granada. Algo que es muy gratificante.

En lo que respecta a la valoración del proyecto, después de este largo periodo de trabajo y aprendizaje considero este proyecto como uno de los trabajos más completos que he realizado. He tenido que aprender gran cantidad de conocimientos nuevos, o ampliar altamente los ya adquiridos con anterioridad a lo largo del periodo de aprendizaje para completar la gran tarea de este proyecto.

El proyecto ha requerido de conocimientos en diversos campos. Ha sido necesario aprender el lenguaje PHP, HTML y JavaScript para el entorno web, además de ampliar los conocimientos sobre el lenguaje Java utilizado para la aplicación cliente-servidor. También se han tenido que configurar varios tipos de servidores, Apache, FTP, RADIUS y LDAP, aparte del servidor *socket* Java, de modo que todos trabajen de forma conjunta. Algo que en el global del trabajo ha obligado a exprimir al máximo mis conocimientos y a empaparme de todos los nuevos conocimientos que he ido adquiriendo a lo largo de la realización de este proyecto.

Anexos

Anexos A

Manual de usuario

En este anexo se explicará a grandes rasgos las posibilidades que ofrece el proyecto y la forma de utilización del mismo.

A.1. Interfaz web

Lo primero que se muestra al entrar en el entorno web es una portada como la en la Figura A.1. En dicha imagen se pueden distinguir tres partes.

Por un lado, pulsando sobre el botón “*Alumno*”, el sistema permite cambiar la portada de los profesores a la portada de los alumnos (Figura A.2). Esta portada es para el acceso al sistema por parte de los alumnos, ya que se realiza distinción de cuentas de usuarios. El sistema ha sido diseñado principalmente para docentes, pero puede ser utilizado por los alumnos cuando éstos tengan que realizar una presentación.

En el centro puede verse el elemento principal, a través del cual se puede acceder a la pantalla de inicio o página principal de usuario (Figura A.3). Esta página se detallará a continuación.

Y por último, en la parte inferior se muestran las notificaciones. En este caso, la notificación corresponde a la posibilidad de crear una nueva cuenta de usuario. Pero estas notificaciones varían en función de las circunstancias. Por ejemplo, cuando se intenta acceder al servicio sin introducir datos, los datos son incorrectos o el servidor de autenticación está fuera de servicio entre otras (véase la Figura A.4). Finalmente, si se accede a la página para crear una nueva cuenta, el resultado es el de la Figura A.5.

En este caso, se piden los datos necesarios para la creación de una nueva cuenta: el nombre completo de usuario, el correo electrónico referente a la Universidad de Granada y una contraseña. Será necesario que esta contraseña



Figura A.1: Portada



Figura A.2: Portada para los alumnos



Figura A.3: Página principal



Figura A.4: Ejemplo de notificación en la portada

Bienvenido al Servicio de Proyección de
Material Docente

Nombre:

Email: @ugr.es

Contraseña:

Repetir contraseña:

[Volver al inicio](#)

Juan Ramón Gutiérrez Martínez


Raspberry Pi

Figura A.5: Página para crear una nueva cuenta

contenga como mínimo ocho caracteres, y al menos una letra y un número, aumentando de esta forma la seguridad de las contraseñas.

En esta página también existe un apartado para las notificaciones. Inicialmente se muestra vacío, solo aparece la opción de volver a la portada. Pero pueden aparecer notificaciones igual que en el caso anterior, cuando las contraseñas no coinciden o no cumplen con los requisitos, los campos están incompletos o la cuenta de usuario ya existe. Un ejemplo de esto se muestra en la Figura A.6.

Cuando se accede a la página principal de usuario (Figura A.3), se muestra en primer lugar un mensaje de bienvenida al usuario. Y al lado de este mensaje existen dos botones correspondientes a las imágenes de la derecha. Una es para cerrar la sesión de usuario (imagen de la puerta) y la otra permite acceder a la configuración de la cuenta de usuario para cambiar la contraseña (véase la Figura A.7). Esta página tiene el mismo funcionamiento que la página utilizada para crear una nueva cuenta (Figura A.5).

Seguidamente, en la parte derecha de la pantalla se observan dos botones. Uno es para actualizar la página y el otro se utiliza para subir nuevos archivos al servidor. Al pulsar sobre dicho botón aparece un cuadro de selección donde se deberá seleccionar el archivo que se desea subir (véase Figura A.8).

Finalmente, en la parte inferior se muestra un listado con los archivos guardados en el servidor de ficheros del usuario. Al pulsar sobre uno de estos archivos se muestra un menú desplegable con las opciones correspondientes a dicho archivo (véase Figura A.8). De izquierda a derecha las opciones co-

The screenshot shows a registration form titled "Bienvenido al Servicio de Proyección de Material Docente". The form is set against a dark background with a faint Raspberry Pi circuit pattern. It includes input fields for "Nombre:" (containing "usuario"), "Email:" (containing "usuario@ugr.es"), "Contraseña:" (containing three asterisks), and "Repetir contraseña:" (containing three asterisks). A "Registrar" button is located below the password fields. A message below the form states: "Contraseña demasiado corta. Debe tener al menos 8 caracteres". A link "Volver al inicio" is positioned below the message. The Raspberry Pi logo and the text "Raspberry Pi" are in the bottom right corner. The footer on the bottom left reads "Juan Ramón Gutiérrez Martínez".

Figura A.6: Ejemplo de notificación en la página para crea una nueva cuenta

The screenshot shows a password change page titled "Servicio de Proyección de Material Docente". In the top left, it says "Bienvenido: 👤 admin 🇪🇸". The central form has input fields for "Nueva contraseña:" and "Repetir contraseña:", with an "Enviar" button below them. A link "Volver a la pagina principal" is located below the form. The Raspberry Pi logo and the text "Raspberry Pi" are in the bottom right corner. The footer on the bottom left reads "Juan Ramón Gutiérrez Martínez".

Figura A.7: Página de configuración para cambiar de contraseña



Figura A.8: Página principal con opciones desplegadas

responden a: enviar el archivo al proyector, descargar el archivo, borrar el archivo y renombrar el archivo. En este último caso, el cuadro para introducir el nuevo nombre del archivo aparece al pulsar sobre el lápiz (se trata también de un cuadro dinámico). Todas estas acciones se realizan de forma instantánea, de forma que el usuario permanece de forma constante en esta página.

Esta página también dispone de un sistema de notificaciones. Estas notificaciones aparecen en la parte superior a los archivos mostrados y corresponden a fallos con la conexión SFTP, errores al realizar alguna de las acciones, o archivos duplicados, entre otras.

El último paso en esta interfaz web corresponde a la página donde se muestra la aplicación cliente. Como se ha mencionado anteriormente, para acceder a esta página se ha de pulsar sobre uno de los archivos, y posteriormente sobre el botón con un proyector. La página que inicialmente aparece es la correspondiente a la Figura A.9. En esta página se ha de seleccionar la clase donde se encuentra el proyector en el que se quiere mostrar el documento.

Una vez seleccionada la clase, se muestra la aplicación cliente en función de si se trata de un documento PDF (Figura A.9) o una imagen (Figura A.11).



Figura A.9: Página de selección de lugar de proyección



Figura A.10: Página con la aplicación cliente para un documento PDF



Figura A.11: Página con la aplicación cliente para una imagen



Figura A.12: Menú archivo de la aplicación cliente para un documento PDF

A.2. Aplicación cliente

La aplicación cliente corresponde a la ventana central que aparece en las páginas mostradas anteriormente. Las opciones y funcionalidades de esta aplicación variarán en función del documento, aunque éstas son muy parecidas.

Para el caso de un documento PDF pueden verse las diferentes opciones en las Figuras A.12 y A.13. Estas opciones se muestran de forma clara para que cualquier usuario pueda entenderlas.

En cambio, si el documento se trata de una imagen, el menú correspondiente a “archivo” solo tiene la opción que permite cerrar el archivo. Y para el menú ver las opciones son las mismas con la salvedad de que se permite

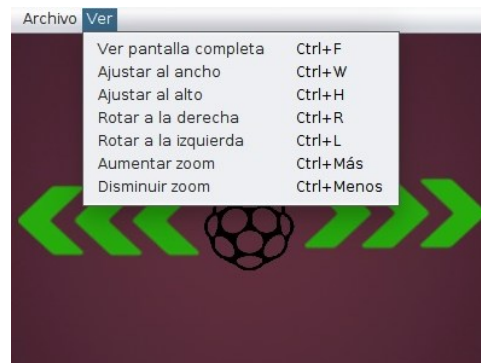


Figura A.13: Menú ver de la aplicación cliente para un documento PDF



Figura A.14: Menú ver de la aplicación cliente para una imagen

voltear la imagen en ambos ejes (véase la Figura A.14).

Por último, hay que aclarar que para un correcto funcionamiento del sistema y en especial de esta aplicación cliente, se debe tener instalado en el dispositivo utilizado la versión de Java 8.5, versión utilizada en la implementación del proyecto.

A.3. Servidor RPi

En lo que al servidor situado en la Raspberry Pi respecta, su uso es muy sencillo (véase la Figura A.15).

```
> java ServidorRPi [opciones] [parametros]
```

Dentro de opciones se puede seleccionar iniciar el servidor en modo *debug* (“-x”). De esta forma se muestran por pantalla todos los mensajes en


```

juanra@linux-Escritorio/pruebas$ java ServidorRPI error
Formato incorrecto. -> java ServidorRPI [opciones] [parametros]

[opciones]
  -x debug      Inicio en modo debug.
[parametros]
  -p puerto     Indica el numero de puerto en el que abrir el servidor.P
Por defecto de utiliza el puerto 8888.
  -h host       Indica la direccion IP del host donde se encuentra alojado el se
rvidor FTP.Por defecto de utiliza la direccion 192.168.1.1.

```

Figura A.15: Página con la aplicación cliente para un documento PDF

```

Archivo Edición Pestañas Ayuda
root@raspberrypi:/home/pi/Servidor# java ServidorRPI -x -h 192.168.1.3
[ServidorRPI]: Modo debug activado
[ServidorRPI]: Iniciando en el puerto 8888... OK!
[ServidorRPI]: Esperando nuevo cliente... █

```

Figura A.16: Página con la aplicación cliente para un documento PDF

la comunicación cliente-servidor, así como los posibles error que aparezcan durante el proceso.

Como parámetros se pueden distinguir dos: uno para seleccionar el puerto en el que se quiere iniciar (“-p”), y otro para seleccionar la dirección IP del servidor de ficheros (“-h”). Si estos parámetros no son seleccionados se establece la conexión por defecto en el puerto 8888 y con la dirección 192.168.1.1 para el servidor de ficheros.

Así, un ejemplo de utilización se muestra en la Figura A.16.

Anexos B

Manual de instalación

En este anexo se mostrará de forma detallada un manual de instalación de forma que se facilite la posterior tarea de implantación del proyecto. Además, con este manual se pretende ayudar a que cualquier persona pueda realizar dicha instalación para su uso personal o para seguir avanzando con las líneas futuras de este proyecto.

Este manual esta creado para la instalación en un sistema Linux, en especial Ubuntu 14.04 [41], versión utilizada en el proyecto. El manual está estructurado en diferentes apartados correspondientes a las diferentes regiones que componen el proyecto global. Se recomienda seguir este manual desde el principio, ya que realizar la instalación de forma desordenada puede causar problemas en un futuro.

B.1. Raspberry Pi

Para preparar la Raspberry Pi de forma que pueda ser utilizada en el proyecto, lo primero que se debe hacer es descargar el sistema operativo que se va a implantar. Como se vió en el capítulo 4, se utilizará RASPBIAN [55] disponible para su descarga desde la web oficial de Raspberry Pi [42]. En la propia web también se explica muy detalladamente cómo realizar la instalación del *software* por lo que es recomendable leer dicha información si se produce algún error.

En primer lugar, hay que tener en cuenta que se debe utilizar una tarjeta SD adecuada (2GB o más). En este caso, lo ideal es utilizar al menos una tarjeta SD de 8GB y clase 10, lo que proporciona una mayor velocidad y, por tanto, mejor rendimiento de Raspberry Pi. Para instalar la distribución en dicha tarjeta, será necesario para usuarios de Windows utilizar el *software* gratuito y redirigido directamente desde la web oficial de RPi,

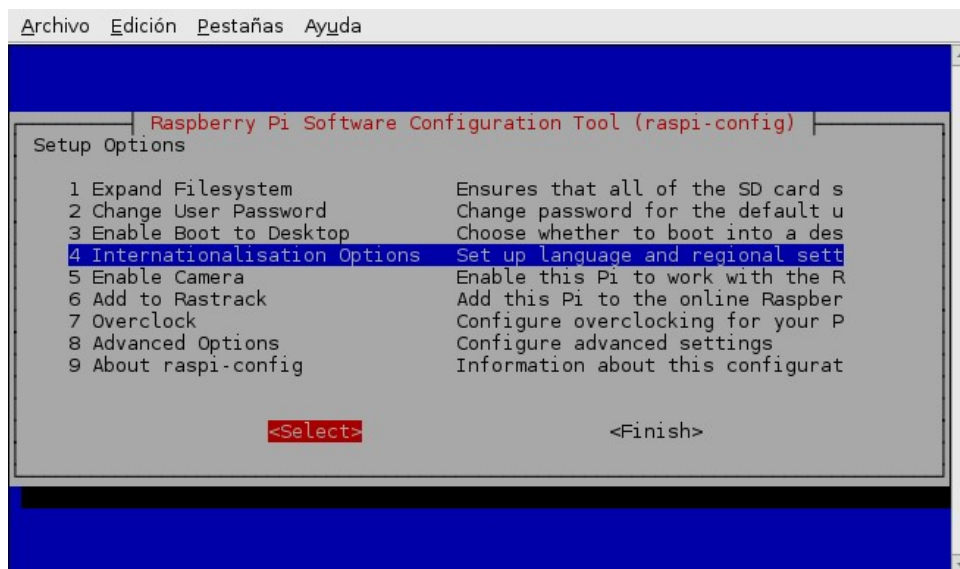


Figura B.1: Ejemplo del panel de configuración

Win32DiskImager.

Es sencillo utilizar este *software*, simplemente hay que seleccionar la imagen y la unidad de destino y pulsar en “write”. Una vez instala la distribución en la tarjeta, ésta puede empezar a funcionar.

Se puede utilizar Raspberry Pi conectando un monitor HDMI, un teclado y un ratón. En el caso de no disponer de esto se puede realizar la conexión mediante ssh desde un PC conectado en la misma red, basta con utilizar el comando:

```
ssh -X pi@x.x.x.x (Sustituir por la dirección IP de la Raspberry Pi)
```

Esto pedirá una contraseña: “raspberrry”.

Después se abrirá un menú de configuración donde se puede cambiar la hora, el tipo de teclado, el idioma y muchas cosas más. Se muestra uno de los ejemplos a continuación, lo demás se hace forma análoga.

Aquí podemos configurar tanto el idioma a utilizar por el sistema por defecto, como la zona horaria y el teclado que se está utilizando.

Para más información sobre la configuración se puede mirar el foro oficial de Raspberry Pi en el que se exponen muchos consejos y recomendaciones. Para el caso del proyecto no será necesaria ninguna configuración especial porque lo que no se profundizará en esta parte.

Una vez terminada la configuración, se ha de actualizar el firmware hasta

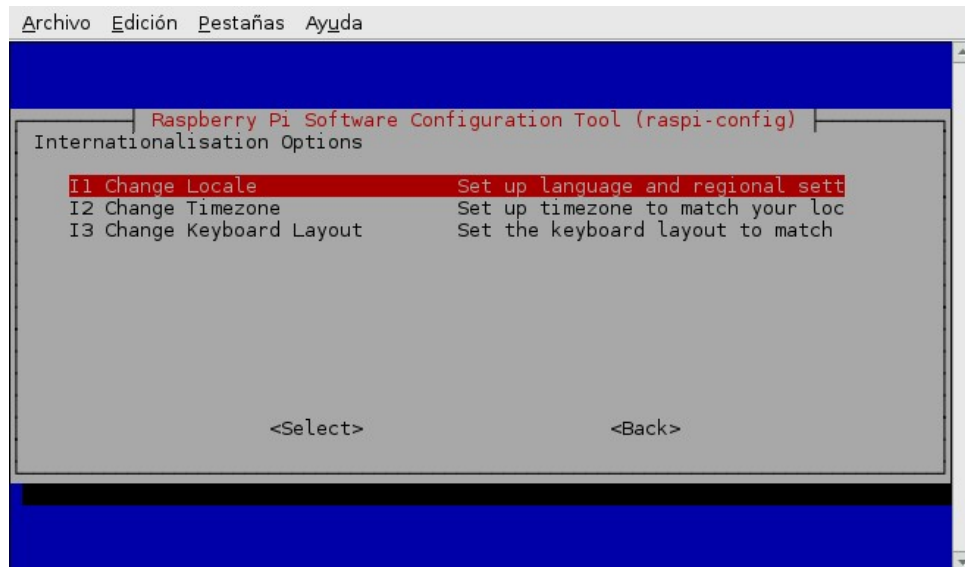


Figura B.2: Ejemplo del panel de configuración

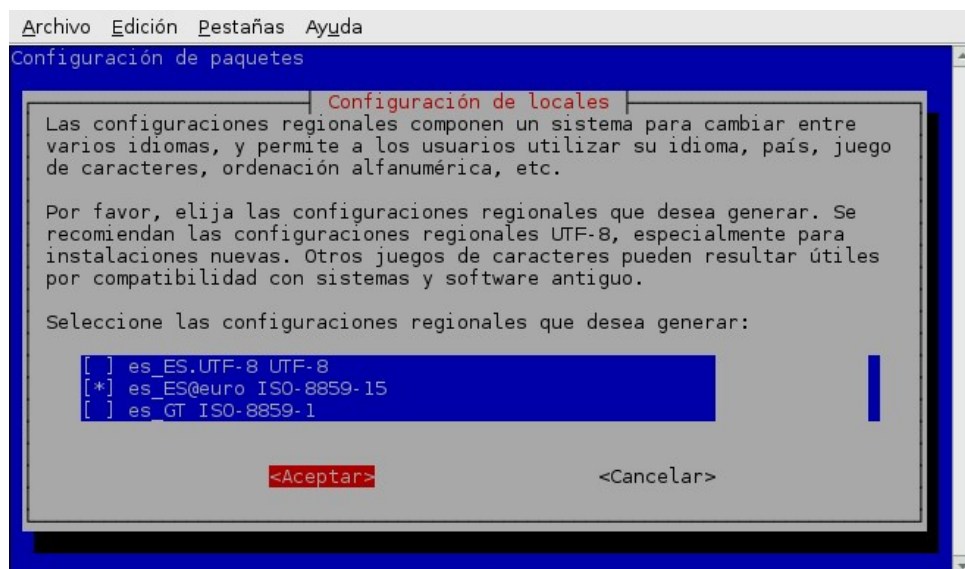


Figura B.3: Ejemplo del panel de configuración

la última versión:

```
> sudo apt-get update
> sudo apt-get upgrade
```

Finalmente, se reinicia la Raspberry Pi y ya estará lista para usarse. Lo siguiente que se debe hacer es instalar el *software* necesario. Generalmente, Java viene por defecto instalado, por lo que se utilizará el comando:

```
> java -version
```

de forma que se compruebe que corresponde con la versión utilizada en este proyecto (versión 8.5). De lo contrario, es necesario actualizar a dicha versión. Esto se puede hacer descargando el *software* desde la página web oficial.

También será necesario instalar el lenguaje PHP en su versión 5.5, la utilizada en este proyecto, que no viene instalado por defecto. Para realizar esto basta con ejecutar el comando:

```
> sudo apt-get install php5 php5-common
```

Por último también será necesario evidentemente la instalación del *software* utilizado para visualizar los documentos, es decir, el programa MuPDF ya mencionado anteriormente. En este caso se puede hacer desde la página web oficial o desde la línea de comandos tal que:

```
> sudo apt-get install mupdf mupdf-tools
```

siendo el segundo paquere el correspondiente los comandos utilizados mediante el teclado. Son además necesarias para que este *software* pueda funcionar las siguientes librerías:

```
> sudo apt-get install build-essential pkg-config libx11-dev libxext-  
dev
```

De esta formas quedará finalizada la instalación de todos los elementos necesarios para el correcto funcionamiento del proyecto. Simplemente queda añadir el *software* anexo a este documento (ServidorRPi.java y descargar.php) a la Raspberry Pi y lazar dicho servidor.

B.2. Servidor VSFTPD

En esta sección se explicará detalladamente cómo instalar y configurar el servidor VSFTPD de forma correcta para la instalación del proyecto. El servidor VSFTPD se encuentra en los repositorios de todas las distribuciones Linux, por lo que para su instalación, basta con un simple comando:

```
> sudo apt-get install vsftpd
```

Para el proceso de configuración el nivel de complejidad de la configuración variará según las necesidades. El fichero de configuración se encuentra en */etc/vsftpd.conf* y viene configurado por defecto con algunas características, por lo que el primer paso va a ser editarlo.

En primer lugar, se comenta la siguiente línea para no permitir conexiones anónimas:

```
#anonymous_enable
```

A continuación, se establecen los siguientes parámetros para enjaular a determinados usuarios, es decir, para evitar que vean otros directorios que no sean los que nosotros indicamos:

```
chroot_local_user=YES
```

```
chroot_list_enable=YES
```

```
chroot_list_file=/etc/vsftpd.chroot_list
```

Con esto se terminan de configurar las opciones básicas para el correcto funcionamiento. Lo siguiente es crear el fichero *chroot_list* que contendrá una lista de los usuarios a enjaular de la siguiente forma:

```
> sudo gedit /etc/vsftpd.conf
```

Y añadiendo la lista de usuarios del sistema a enjaular:

```
usuario1
```

```
usuario2
```

Simplemente se han de crear tantas líneas como usuarios vayan a tener acceso al servidor VSFTPD. En este caso, los usuarios del servicio FTP serán los usuarios del sistema.

Para crear un nuevo usuario, en primer lugar se crea el directorio:

```
> sudo mkdir [carpeta_usuario]
```

También se va a crear un *shell virtual* o fantasma, para que los usuarios no puedan conectarse a una sesión del sistema operativo:

```
> sudo mkdir /bin/ftp
```

Después hay que editar el siguiente fichero:

```
> sudo gedit /etc/shells
```

Donde se agrega al final del mismo: /bin/ftp. Y ya se pueden añadir usuarios:

```
> sudo useradd -g ftp -d [carpeta_usuario] -s /bin/ftp [usuario]
```

Y posteriormente se le añade una contraseña:

```
> sudo passwd [usuario]
```

De esta forma se habrá creado un nuevo usuario. Ahora se añade a la lista de usuarios enjaulados, anteriormente mencionada:

```
> sudo touch vsftpd.chroot_list (si no se ha creado antes)
> sudo echo [usuario] vftpd.chroot_list
> sudo /etc/init.d/vsftpd restart
```

Hay otra opción que consiste en la creación de usuarios ‘fantasma’, es decir, usuarios que no existan como tal en el sistema pero que sí están en un fichero de texto en formato usuario y contraseña.

El siguiente paso es la configuración del servidor para que acepte conexiones seguras. El protocolo FTP no codifica los datos, ni siquiera de los credenciales del usuario. El hecho de que alguien pudiera capturar un paquete en la ruta sería muy grave para la seguridad del sistema. Por tanto, la solución es utilizar un protocolo seguro, SFTP.

Para configurar el servidor VSFTPD de esta forma hay que generar un certificado, y para ello se da por supuesto que se tiene instalado OpenSSH. De lo contrario, se puede instalar desde el repositorio de Linux.

```
> sudo apt-get install openssh-server openssh-client
```

El certificado se puede generar fácilmente utilizando el siguiente comando, donde se generará un certificado con estructura X.509, algoritmo de cifrado RSA de 1024 Kbits (para este paso será necesario tener instalado OpenSSL):

```
> sudo apt-get install openssl
> sudo openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout /etc/ssl/private/vsftpd.key -out /etc/ssl/certs/vsftpd.pem
```

B. Manual de instalación

Lo siguiente será modificar el fichero de configuración */etc/vsftpd.conf* para añadir las líneas con la configuración de seguridad. Tal que, se añade al fichero:

```
# Secure FTP
#
# Activamos el soporte SSL
ssl_enable=YES

# No se obliga el establecimiento de conexiones encriptadas mediante SSL para
# usuarios locales.
force_local_logins_ssl=NO

# No se obliga a que las transferencias de datos locales sean encriptados con
# SSL. force_local_data_ssl=NO

require_ssl_reuse=NO

# Se habilitan los soportes para las diversas versiones de SSL
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO

# Se especifica la ubicación del Certificado Generado.
rsa_cert_file=/etc/ssl/certs/vsftpd.pem

rsa_private_key_file=/etc/ssl/private/vsftpd.key
```

Hay que comentar las líneas de los certificados RSA que aparecen por defecto para que se utilice el certificado nuevo generado. Finalmente reinicia el servicio. También se puede instalar:

```
> sudo apt-get install ftp-ssl
```

para así obligar, siempre que se pueda, a que se inicie la conexión FTP segura.

B.3. Servidor OpenLDAP

La instalación de OpenLDAP es sencilla puesto que estamos en un entorno Linux. Simplemente se instala desde el repositorio:

```
> sudo apt-get install slapd ldap-utils
```

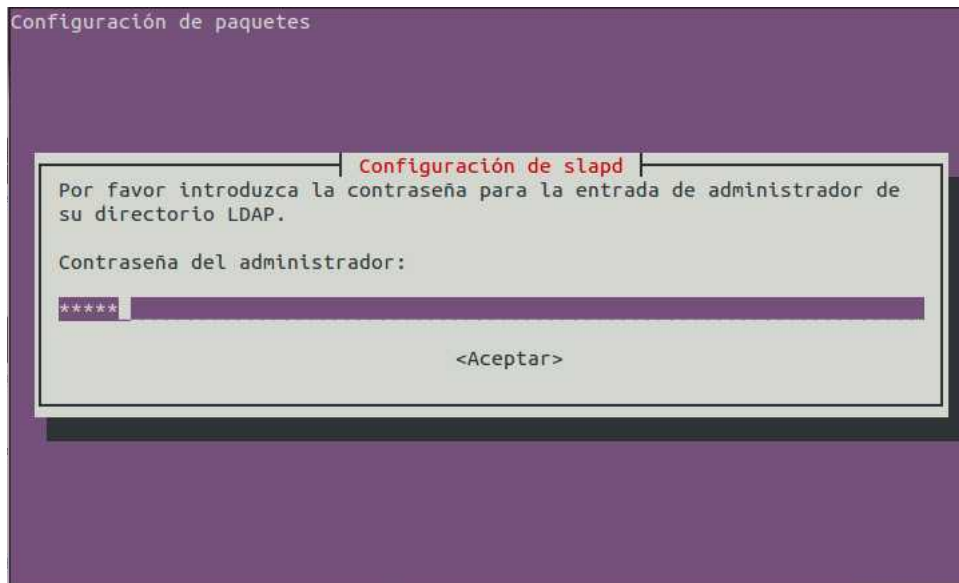



Figura B.4: Interfaz de configuración del servidor OpenLDAP

El proceso de instalación pedirá una contraseña. El caso mas sencillo es utilizar para la configuración de todos los servidores un usuario “admin” con contraseña “admin”.

Una vez la instalación es completada se procede a la configuración del servidor. Para realizar la configuración hay que ejecutar el siguiente comando:

```
> sudo dpkg-reconfigure slapd
```

Al ejecutar este comando aparecerá una interfaz como la de la Figura B.4. Mediante dicha interfaz se configurarán las opciones de la siguiente manera:

```
Desea omitir la configuración del servidor OpenLDAP? NO
DNS Domain Name: ugr.es (puede ser diferente)
Organization Name: ugr.es (puede ser diferente)
Database: BDB
Desea que se borre la base de datos cuando se purgue el paquete slapd?: NO
Desea mover la base de datos antigua? SI
Contraseña del administrador: admin
Verificar la contraseña: admin
Allow LDAPv2 protocol?: NO
```

Con estas modificaciones ya debería estar funcionando nuestro servidor de base de datos correctamente. Para comprobar que funciona se han de introducir las primeras entradas a una base de datos vacía. Como ya se sabe,

el servidor LDAP utiliza un tipo específico de ficheros para añadir nuevas entradas a la base de datos. Estos ficheros son los denominados *.ldif*. En este sentido, cabe diferenciar entre los archivos utilizados para crear grupos de personas o entradas y los utilizados para añadir dichas personas o entradas.

Para el caso que nos concierne se crean dos grupos: uno correspondiente a los profesores y otro para los alumnos. Además, se crea un usuario perteneciente al grupo de profesores a modo de prueba para comprobar el correcto funcionamiento de nuestro servidor OpenLDAP.

El fichero *grupos.ldif* contiene:

```
dn: ou=profesores, dc=ugr, dc=es
objectClass: top
objectClass: organizationalUnit
ou: profesores
description: Profesores

dn: ou=alumnos, dc=ugr, dc=es
objectClass: top
objectClass: organizationalUnit
ou: alumnos

description: Alumnos
```

Y el fichero *usuarios.ldif*:

```
dn: cn=prueba, ou=profesores, dc=ugr, dc=es
objectclass: top
objectclass: organizationalPerson
sn: apellido
cn: nombre
userPassword: prueba
description: Profesor
```

Cuando se tienen estos ficheros creados el proceso a seguir para añadir las nuevas entradas es el siguiente: en primer lugar, se detiene el servidor. Posteriormente, será necesario añadir en primer lugar los nuevos grupos y seguidamente los nuevos usuarios.

```
> sudo /etc/init.d/slaped stop
```

Para añadir los grupos:

```
> ldapadd -x -D "cn=admin,dc=ugr,dc=es" -w admin -f /opt/lampp/htdocs/
PiServer/Usuarios/grupos.ldif
```

```

root@linux:/home/juanra# ldapadd -x -W -D "cn=admin,dc=ugr,dc=es" -f /var/www/html
/PiServer/Usuarios/grupos.ldif
Enter LDAP Password:
adding new entry "ou=profesores, dc=ugr, dc=es"

adding new entry "ou=alumnos, dc=ugr, dc=es"

root@linux:/home/juanra# ldapadd -x -W -D "cn=admin,dc=ugr,dc=es" -f /var/www/html
/PiServer/Usuarios/usuarios.ldif
Enter LDAP Password:
adding new entry "cn=prueba, ou=profesores, dc=ugr, dc=es"

root@linux:/home/juanra# █

```

Figura B.5: Resultado de añadir nuevas entradas a la base de datos

Siendo la respuesta esperada:

```

added: "ou=profesores ,dc=ugr ,dc=es"
added: "ou=alumnos ,dc=ugr ,dc=es"

```

Para añadir los usuarios:

```

> ldapadd -x -D "cn=admin,dc=ugr,dc=es" -w admin -f /direccion/al/
archivo/usuarios.ldif

```

Siendo la respuesta esperada:

```

added: "ou=profesores ,dc=ugr ,dc=es"
added: "cn=prueba,ou=profesores ,dc=ugr ,dc=es"

```

El resultado obtenido de realizar estas operaciones debe ser algo similar a la Figura B.5. Finalmente se inicia de nuevo el servidor y se realiza una búsqueda para verificar que se han añadido las nuevas entradas con éxito:

```

> sudo /etc/init.d/slaped start
> ldapsearch -x -b "dc=ugr,dc=es"

```

Siendo el resultado esperado algo como la Figura B.6.

Si el resultado de la búsqueda es exitoso y similar a los mostrados se puede considerar que el servidor funciona correctamente y está configurado y listo para ser usado por el servidor RADIUS.

Hay que añadir que estos ficheros de configuración no sólo pueden utilizarse para la adición de nuevas entradas sino que además pueden modificar las entradas ya existentes o incluso borrarlas.

```
root@linux:/home/juanra# ldapsearch -x -b "dc=ugr,dc=es"
# extended LDIF
#
# LDAPv3
# base <dc=ugr,dc=es> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# ugr.es
dn: dc=ugr,dc=es
objectClass: top
objectClass: dcObject
objectClass: organization
o: ugr.es
dc: ugr
# admin, ugr.es
dn: cn=admin,dc=ugr,dc=es
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
```

Figura B.6: Resultado de realizar una búsqueda en la base de datos

B.4. Servidor FreeRADIUS

El siguiente paso es la instalación y configuración del servidor de autenticación RADIUS. Como se dijo en el capítulo 4, se utilizará el servidor FreeRADIUS. Este servidor se encuentra disponible de forma gratuita en los repositorios de cualquier distribución Linux. Además, también será necesario instalar el módulo ldap.

```
> sudo apt-get install freeradius freeradius-ldap
```

Para activar el módulo ldap se ha de editar en el fichero de configuración donde se habilitan estos módulos:

```
> sudo gedit etc/freeradius/sites-available/default
```

En este fichero hay que descomentar la información referente a ldap:

```
authorize {
    ldap
}

authenticate {
    Auth-Type LDAP {
        ldap
    }
}
```

```

root@linux:/etc/freeradius# radtest admin admin localhost 1812 testing123
Sending Access-Request of id 27 to 127.0.0.1 port 1812
  User-Name = "admin"
  User-Password = "admin"
  NAS-IP-Address = 127.0.1.1
  NAS-Port = 1812
  Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=27, length=20
root@linux:/etc/freeradius# █

```

Figura B.7: Comprobación del correcto funcionamiento del servidor FreeRADIUS

Una vez habilitado el módulo ldap hay que configurarlo para que el servicio FreeRADIUS pueda utilizarlo. Esto se hace en el fichero *etc/freeradius/modules/ldap*. De forma que se añadan las siguientes líneas:

```

server = 'ldap://localhost'
basedn = 'dc=ugr,dc=es'

filter = '(cn=% %Stripped-User-Name:- %User-Name)'

```

Además, en el fichero */etc/raddb/users*, donde se comprueban los usuarios autorizados se añade el método de autenticación basado en ldap:

```

#
# First setup all accounts to be checked against the UNIX /etc/passwd.
# (Unless a password was already given earlier in this file).
#
DEFAULT Auth-Type = System
Fall-Through = 1
#
# Defaults for LDAP
#
DEFAULT Auth-Type := LDAP
Fall-Through = 1

```

Con esto queda el servidor completamente configurado. Para comprobar que funciona correctamente se puede lanzar el siguiente comando. Lo que dará como resultado algo similar a la Figura B.7.

```
> sudo gedit etc/freeradius/sites-available/default
```

B.5. Servidor Apache

El último servidor que se ha de instalar es el servidor HTTP. En este caso se utiliza el servidor Apache. Este servidor es más sencillo de instalar y configurar que los anteriores. Será necesario instalar tanto el servidor como el lenguaje PHP utilizado en la creación de un entorno web dinámico.

```
> sudo apt-get install apache2 php5 libapache2-mod-php5
```

Será necesario además la instalación de los módulos LDAP y RADIUS para PHP.

```
> sudo apt-get install php5-ldap php5-radius
```

El directorio principal utilizado por el servidor Apache es `/var/www/html`. En ese directorio se debe añadir la carpeta *"SPMD"* anexa a este documento correspondiente al entorno web.

Además, se modificará el servidor de forma que se cambie la configuración HTTP por HTTPS. Para ello se ha de generar un certificado de forma similar al servidor VSFTPD.

```
> sudo mkdir /etc/apache2/ssl
> sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
```

Y finalmente, quedan todos los servidores configurados y preparados para el funcionamiento del sistema.

B.6. Servidor RPi

Finalmente, se añaden los ficheros correspondientes al servidor RPi (descargar.php y ServidorRPi.java) en una carpeta del sistema de la Raspberry Pi. Desde ese directorio se lanzará dicho servidor según lo explicado en la sección A.3 del Anexo A.

Bibliografía final

- [1] A. Andrade, A. M. Rodríguez, G. Chabur Sánchez and R. A. Almeida Delgado, “Las tics como estrategia didáctica en la docencia universitaria,” 2014.
- [2] **Microsoft Office**. Disponible en: <http://office.microsoft.com/es-es/>
- [3] M. Treleven, R. J. Penlesky, T. E. Callarman, C. A. Watts, D. J. Bragg, “The value of powerpoint animations in theaching specific operations and supply chain management techniques and concepts”. *Proceedings for the Northeast Region Decision Sciences Institure (NEDSI)*, Apr 01, 2013, pp. 777-784.
- [4] J. Salinas, “Innovación docente y uso de las TIC en la enseñanza universitaria,” *Revista Universidad y Sociedad Del Conocimiento*, vol. 1, pp. 1-16, 2004.
- [5] R. A. Berk, “Research on PowerPoint: From basic features to multimedia”. *International Journal of Technology in Teaching and Learning*, Jun 2011, vol. 7 (1), pp. 24-35.
- [6] F. Frommer and E. Julibert, *El Pensamiento PowerPoint: Ensayo Sobre Un Programa Que Nos Vuelve Estúpidos*. Barcelona: Península, 2011.
- [7] N. C. Lago, “La educación en las universidades: infraestructuras técnicas”.
- [8] A. Szabo and N. Hastings, “Using IT in the undergraduate classroom: should we replace the blackboard with PowerPoint?” *Comput. Educ.*, vol. 35, pp. 175-187, 11/1, 2000.
- [9] W. Blokzijl and B. Andeweg, “The effects of text slide format and presentational quality on learning in college lectures,” in *Professional Communication Conference*, 2005, pp. 288-299.

- [10] R. Palomo López, J. Sánchez Rodríguez and J. Ruiz Palmero, *Enseñanza Con TIC En El Siglo XXI: La Escuela 2.0*. Alcalá de Guadaira Sevilla: Mad, 2008.
- [11] D. J. G. Gil, M. L. Cacheiro-González and J. Dulac, “La pizarra digital interactiva como recurso docente, *Teoría De La Educación: Educación y Cultura En La Sociedad De La Información*”, vol. 10, pp. 127-145, 2009.
- [12] Rosario, Jimmy, 2005, “La Tecnología de la Información y la Comunicación (TIC). Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual”.
- [13] Projectores Sony disponibles en: <http://www.sony.es/pro/products/projectors-home>.
- [14] Projectores Mitsubishi disponibles en: <http://www.mitsubishielectric.es/proyectores/proyectores-multimedia/>.
- [15] Projectores Panasonic disponibles en: <http://business.panasonic.es/sistemas-visuales/productos-y-accesorios/proyectores/proyectores-de-instalacion-fija>.
- [16] Artículo de Sony sobre proyectores destinados a la universidad disponible en: <http://www.sony.es/pro/article/displays-projectors-halls-auditoriums-article>.
- [17] Wireless display system ofrecido por ipevo y disponible en: http://www.ipevo.com/prods/wireless_presentation_system.
- [18] MARTÍN IGLESIAS, J. P., La pizarra digital interactiva (PDi) en educación, Anaya Multimedia, Madrid, 2010. (Publicación septiembre 2010).
- [19] C. S. Mayayo and C. D. Á. Álvarez, “4-PDI (LA PIZARRA DIGITAL)”.
- [20] Web informativa sobre pizarras interactivas: <http://www.pizarrasinteractivas.com/>.
- [21] **Apple TV** sitio web oficial: <http://www.apple.com/es/appletv/>.
- [22] **Apple** sitio web oficial: <http://www.apple.com/es/>.
- [23] Centers, *Take Control of Apple TV*. Ithaca, NY: TidBITS Pub., 2014.
- [24] **AirPlay** sitio web oficial: <http://www.apple.com/es/appletv/airplay/>.

BIBLIOGRAFÍA FINAL

- [25] S. N. Aznar, “Aumentando la interacción en el aula ordinaria mediante el iPad y demás dispositivos de m-learning,” *Revista De Educación y Derecho*, 2014.
- [26] **GoClass** sitio web oficial: <https://www.goclass.com/>.
- [27] D. Frakes, “AirPlay Mirroring.” *Macworld*, vol. 29, pp. 54-54, 2012-10-01.
- [28] **Wireless Display** sitio web oficial: <http://www.wirelessdisplay.com/index.html>.
- [29] E. Savitz, “3 Wireless Technologies That Will Change Home Networking.” *Forbes.Com*, pp. 3-3, 2012-08-01.
- [30] **Wi-Fi Alliance’s** sitio web oficial: <http://www.wi-fi.org/discover-wi-fi/wi-fi-certified-miracast/>.
- [31] M. Stauffer, “Connectivity solutions for smart TVs,” in *Consumer Electronics-Berlin (ICCE-Berlin), 2012 IEEE International Conference*, 2012, pp. 245-249.
- [32] **Chromecast** sitio web oficial: <http://www.google.es/intl/es/chrome/devices/chromecast/>.
- [33] J. Newman, “Chromecast Revisited.” *Time.Com*, pp. 1-1, 2013-12-13.
- [34] S. OCHS, “Google’s Chromecast: The Internet on your TV.” *PC World*, vol. 31, pp. 85-87, 2013-10-01.
- [35] T. R. Weiss, “Google’s Chromecast Now Available in Canada, Europe.” *Eweek*, pp. 4-4, 2014-03-19.
- [36] Web oficial del servidor web **Apache**: <http://www.apache.org/>.
- [37] Web oficial del servidor de ficheros **VSFTP**: <https://security.appspot.com/vsftpd.html>.
- [38] Web oficial del servidor RADIUS, FreeRADIUS: <http://freeradius.org/>.
- [39] Web oficial del servidor de base de datos **OpenLDAP**: <http://www.openldap.org/>.
- [40] Web oficial de **Raspberry Pi**: <http://www.raspberrypi.org/>.
- [41] Sistema operativo **Ubuntu 14.04**. Disponible en: <http://www.ubuntu.com/>.

- [42] **Distribuciones oficiales para Raspberry Pi.** Disponibles en: <http://www.raspberrypi.org/downloads/>.
- [43] **Java.** Disponible en: <http://www.java.com/es/>.
- [44] Web oficial de **PHP:** <http://www.php.net/>.
- [45] Entorno de desarrollo **Netsbeans.** Disponible en: <https://netbeans.org/>.
- [46] Web oficial de **MuPDF.** Disponible en: <https://www.mupdf.com/>
- [47] Web oficial del **COIT:** <https://www.coit.es/>.
- [48] E. Upton and G. Halfacree, Raspberry Pi. Madrid: Anaya Multimedia, 2014.
- [49] C. Albanesius, "Order Limit for \$35 Raspberry Pi PC Lifted." PC Magazine, pp. 1-1, 2012-07-01.
- [50] Anonymous "Raspberry Pi goes professional." Electronics Weekly, pp. 6-6, 2014-04-16.
- [51] Descripción autenticación UGR disponible en: <http://csirc.ugr.es/informatica/ServiciosWeb/AutenticacionUsuariosUGR.html>
- [52] **INTECO** sitio web oficial: <http://www.inteco.es/>
- [53] Artículo de INTECO sobre servidores web disponible en: http://cert.inteco.es/cert/Notas_Actualidad/guia_basica_securizacion_servidor_web_apache_20120816?postAction=getLatestInfo
- [54] Web oficial de **FileZilla:** <https://filezilla-project.org/>
- [55] Web oficial de **OpenSSH:** <http://www.openssh.com/>
- [56] **RASPBIAN:** <http://raspbian.org/>
- [57] **OPENELEC:** http://wiki.openelec.tv/index.php?title=Raspberry_Pi_FAQ
- [58] **PIDORA:** <http://pidora.ca/>
- [59] **RASPBMC:** <http://www.raspbmc.com/>
- [60] **ARCH LINUX:** <http://archlinuxarm.org/platforms/armv6/raspberry-pi>
- [61] **RISC OS:** <https://www.riscosopen.org/wiki/documentation/show/Welcome%20to%20RISC%20S%20Pi>

BIBLIOGRAFÍA FINAL

- [62] Web oficial del software **FBIDA**: <https://www.kraxel.org/blog/linux/fbida/>
- [63] Web oficial del software **Green**: <https://github.com/schandinat/green/wiki>
- [64] M. Horowitz, S. Lunt, “FTP Security Extensions”, *RFC 2228, Internet Engineering Task Force*, Octubre 1997.
- [65] Web oficial **OpenSSL**: <http://www.openssl.org/>
- [66] E. Rescorla, “Transport Layer Security (TLS) Renegotiation Indication Extension”, *RFC 5746, Internet Engineering Task Force*, Febrero 2010.
- [67] S. Turner, “Prohibiting Secure Sockets Layer (SSL) Version 2.0”, *RFC 6176, Internet Engineering Task Force*, Marzo 2011.
- [68] A. Freier, P. Karlton, P. Kocher, “The Secure Sockets Layer (SSL) Protocol Version 3.0”, *RFC 6101, Internet Engineering Task Force*, Agosto 2011.
- [69] G. Carter, *LDAP System Administration*. Sebastopol, CA: O’Reilly, 2003.
- [70] C. Weider, A. Herron, A. Anantha, A. Howes, “LDAP Control Extension for Simple Paged Results Manipulation”, *RFC 2696, Internet Engineering Task Force*, Septiembre 1999.
- [71] Y. Fernández Hansen, A. A. Ramos Varón and J. García Morán, *AAA - RADIUS - 802.1x: Sistemas Basados En La Autenticación En Windows y Linux-GNU: Seguridad Máxima*. Madrid: Ra-Ma, 2008.
- [72] D. Van der Walt and I. ebrary, *FreeRADIUS Beginner’s Guide*. Birmingham U.K.: Packt Pub. Ltd., 2011.
- [73] J. Frans, “HTTP Authentication: Basic and Digest Access Authentication”, *RFC 2617, Internet Engineering Task Force*, Junio 1999.
- [74] E. Rescorla, A. Schiffman, “The Secure HyperText Transfer Protocol”, *RFC 2660, Internet Engineering Task Force*, Agosto 1999.
- [75] P. Housley, P. Hoffman, “Internet X.509 Public Key Infrastructure Operational Protocol FTP and HTTP”, *RFC 2585, Internet Engineering Task Force*, Mayo 1999.
- [76] Manual oficial de **PHP** disponible en: <http://www.php.net/manual/es/index.php>

- [77] Anonymous *LAMP: Desarrollo Web Con Linux, Apache, MySQL y PHP 5*. Madrid: Anaya Multimedia, 2006.
- [78] S. McCracken, “Curso de programación Web con HTML5, CSS, JavaScript, PHP 5.6 y MySQL” pp. 1148, 2011.
- [79] D. S. McFarland, “JavaScript” pp. 656, 2009.
- [80] D. W. Mercer, *Php 5*. Madrid: Anaya Multimedia, 2005.
- [81] Documentación oficial de **Java** disponible en: <http://docs.oracle.com/javase/7/docs/api/>
- [82] K. L. Calvert and M. J. Donahoo, *TCP/IP Sockets in Java*. Amsterdam; Boston: Elsevier/Morgan Kaufmann, 2008.
- [83] Herramienta de análisis de red **Wireshark** disponible: <http://www.wireshark.org/>
- [84] Tema **Piracast** del foro oficial de **Raspberry Pi** disponible en: <http://www.raspberrypi.org/forums/viewtopic.php?t=60636>

