



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Diseño e implementación de un Sistema de Información para una PYME

TRABAJO FIN DE MÁSTER

Máster Universitario en Ingeniería Informática

Autor: Álvaro Roig Coves

Tutor: Xavier Molero Prieto

4 de julio de 2017

*" Ad augusta
per angusta "*

Resumen

Mediante este proyecto planteamos el análisis de las necesidades de digitalización de procesos de una PYME, tras el cual pretendemos llevar a cabo el diseño e implementación de un Sistema de Información especialmente adaptado a dichas necesidades, dotando a la PYME en cuestión de la capacidad de digitalizar sus procesos de negocio con el fin de ser agilizados y obtener, en general, un mayor rendimiento en los mismos.

Entre otras mejoras dotaremos a la organización de un sistema fiable de almacenamiento de información mediante el cual podrán llevar a cabo tareas y consultas propias del día a día de una forma más eficiente y cómoda.

Entre los elementos de diseño que vamos a desarrollar podemos destacar un Sistema de Información web a medida, la instalación y configuración de un servidor de bajo coste, la puesta en marcha de servicios en dicho servidor con el fin de que sean usados por parte del propio Sistema de Información, y el diseño de una aplicación móvil multiplataforma mediante la que otorgar a los trabajadores de la PYME de la posibilidad de llevar a cabo consultas de forma ágil y sencilla en la base de datos de la empresa.

Palabras clave: Sistema de Información, servidor, servicios, digitalización, PYME, aplicación móvil multiplataforma, Raspberry Pi.

Resum

Mitjançant aquest projecte plantegem l'anàlisi de les necessitats de digitalització de processos d'una PIME, després del qual pretenem dur a terme el disseny e implementació d'un Sistema d'Informació especialment adaptat a aquestes necessitats, dotant a la PIME en qüestió de la capacitat de digitalitzar els seus processos de negoci amb la finalitat de ser agilitats i obtenir, en general, un major rendiment en els mateixos.

Entre altres millores dotarem a l'organització d'un sistema fiable d'emmagatzematge d'informació mitjançant el qual podrán dur a terme tasques i consultes pròpies del dia a dia d'una forma més eficient i còmoda.

Entre els elements de disseny que anem a desenvolupar podem destacar un Sistema d'Informació web a mesura, la instal·lació i configuració d'un servidor de baix cost, l'engegada de serveis en aquest servidor amb la finalitat de que siguin usats per part del propi Sistema d'Informació, i el disseny d'una aplicació mòbil multiplataforma mitjançant la qual atorgar als treballadors de la PIME la

possibilitat de dur a terme consultes de forma àgil i senzilla en la base de dades de l'empresa.

Paraules clau: Sistema d'Informació, servidor, serveis, digitalització, PIME, aplicació mòbil multiplataforma, Raspberry Pi.

Abstract

Through this project we propose the analysis of the needs of the process of digitization of an SME, after which we intend to carry out the design and implementation of an Information System specially adapted to these needs, giving the SME in question the ability to digitize their business processes in order to be speeded up, and generally, get a higher performance of them.

Among other improvements we will provide the organization with a reliable system of information storage through which they can carry out tasks and queries of their own day-to-day in a more efficient and comfortable way.

Among the elements of design that we are going to develop we can highlight a tailor-made Web Information System, the installation and configuration of a low cost server, the commissioning of services in said server to be used by the Information System itself, and the design of a cross-platform mobile application through which the SME workers could consult in a simple and agile way the company's database.

Keywords: Information System, server, services, digitalization, SME, cross-platform mobile application, Raspberry Pi.

Índice general

1. Motivación y objetivos del trabajo	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Estructura de la memoria	3
1.4. Uso de la bibliografía	4
2. Necesidades de digitalización de información de la PYME	5
2.1. Análisis de requerimientos	5
2.2. Servicios a desplegar	6
2.2.1. Introducción a Raspberry Pi	8
3. Diseño e implementación del Sistema de Información	13
3.1. Computador central	13
3.1.1. Sistema operativo	14
3.1.2. Configuración base del servidor	15
3.2. Gestión remota	19
3.2.1. Diseño del portal de administración remoto	19
3.2.2. Instalación de servidores en la RaspberryPi	38
3.2.3. Servicios de reinicio y apagado del servidor	42
3.3. Consideraciones de seguridad	43
3.3.1. Mecanismos de seguridad en la empresa	43
3.3.2. Mecanismos de seguridad implantados	44
4. Diseño de la aplicación móvil de usuario	45
4.1. Introducción a Xamarin	45
4.2. Diseño de la aplicación	47

4.3. Comunicación con el servidor	53
4.3.1. Configuración del servidor	53
4.3.2. Diseño del servicio RESTful	54
4.3.3. Publicación de las aplicaciones	56
5. Presupuesto del proyecto	61
5.1. Material necesario	61
5.2. Estimación de horas de trabajo	62
6. Estudios de consumo y valoración de los usuarios	65
6.1. Consumo energético del servidor	65
6.2. Valoración por parte de los usuarios	66
6.2.1. Formularios de consulta	66
6.2.2. Resultados obtenidos	67
7. Conclusiones	71
7.1. Objetivos Alcanzados	71
7.2. Contratiempos y problemas sufridos	72
7.3. Posibles mejoras	72
Bibliografía	72
Apéndice	74
A. Códigos del portal de administración telemática	75
A.1. Página de acceso	75
A.2. Conexión a la base de datos	80
A.3. Cierre de sesión de usuario	80
A.4. Reinicio y apagado del servidor	81
A.5. Librería de funciones	82
A.6. Página de inicio	94
B. Códigos de la aplicación móvil	157
B.1. Página de inicio de la App	157
B.2. Página de información de la App	160
B.3. Página de listado de clientes de la App	163

B.4. Página de listado de productos de un cliente de la App	167
B.5. Página de información de un producto de la App	173
C. Códigos del API RESTful	179
C.1. Servicio RESTful en el servidor	179
D. Formularios de valoración de los usuarios	183
D.1. Formulario de valoración general	183

Índice de figuras

2.1. Logo de Embalajes Industriales Ramírez Tomás S.L.	5
2.2. Sistema de trabajo original	6
2.3. Sistema de trabajo modificado	7
2.4. Raspberry Pi 3 Modelo B	10
2.5. Logos de los principales OS para Raspberry Pi	11
2.6. Comparativa entre modelos de Raspberry Pi actuales	12
3.1. Logo del sistema operativo Raspbian	14
3.2. Descarga oficial de Raspbian	15
3.3. Montaje de la imagen de sistema operativo	16
3.4. Configuración de la dirección IP (interfaces)	17
3.5. Configuración de la dirección IP (dhcpcd.conf)	18
3.6. Configuración del hostname (hostnamectl)	18
3.7. Configuración del hostname (hosts)	18
3.8. Página de acceso al portal de trámites telemáticos	20
3.9. Página de inicio del portal de trámites telemáticos	21
3.10. Menú lateral del portal de trámites telemáticos	22
3.11. Menú superior del portal de trámites telemáticos	22
3.12. Acciones asociadas a la gestión de clientes	23
3.13. Listado de clientes registrados	24
3.14. Dar de alta un nuevo cliente en el sistema	25
3.15. Eliminar un cliente	25
3.16. Acciones asociadas a la gestión de la base de datos	26
3.17. Listado de entradas de productos	27
3.18. Alta de nueva entrada de producto	28

3.19. Confirmación de alta de nueva entrada	29
3.20. Modificación de entrada en la base de datos (selección de cliente)	29
3.21. Modificación de entrada en la base de datos (selección de entrada)	30
3.22. Modificación de entrada en la base de datos (edición de entrada)	30
3.23. Supresión de entrada de la base de datos (selección de cliente)	31
3.24. Supresión de entrada de la base de datos (selección de entrada)	31
3.25. Acciones asociadas a la gestión de copias de seguridad	32
3.26. Listar copias de seguridad	32
3.27. Crear nueva copia de seguridad	33
3.28. Supresión de copia de seguridad	33
3.29. Alta de nuevo usuario	34
3.30. Acciones asociadas a la gestión de usuarios	35
3.31. Listado de usuarios	35
3.32. Modificación de usuarios (selección de usuario)	36
3.33. Modificación de usuarios (edición de datos)	36
3.34. Eliminar un usuario	37
3.35. Interfaz para reinicio y apagado del servidor	42
4.1. Logo de Xamarin	45
4.2. Arquitectura de librerías en Xamarin	46
4.3. Entorno de desarrollo integrado de Xamarin	47
4.4. Páginas de inicio de la App	49
4.5. Páginas de información de la App	50
4.6. Páginas de listado de clientes de la App	51
4.7. Páginas de listado de productos de la App	51
4.8. Páginas de características de producto de la App	52
4.9. Logo de Flask	53
4.10. Publicación de la App para Android	57
4.11. Publicación de la App para iOS	58
4.12. Publicación de la App en la web	59

5.1. Servidor Raspberry Pi	62
5.2. Presupuesto final del proyecto	63
6.1. Valoración del gerente	67
6.2. Valoración de un encargado	68
6.3. Valoración de un operario	68
6.4. Notas de los formularios de valoración de usuario	69
D.1. Formulario de valoración de usuario	184

Índice de tablas

3.1. Órdenes para crear y dar permisos al grupo de trabajo que utiliza Apache	38
3.2. Órdenes para actualizar los repositorios e instalar Apache	38
3.3. Órdenes para instalar MySQL y PHPMyAdmin	39
3.4. Órdenes para crear ficheros de configuración	40
3.5. Tabla de MySQL para guardar credenciales web	40
3.6. Tabla de MySQL para guardar credenciales App	41
3.7. Tabla de MySQL para guardar el registro de clientes	41
3.8. Instalación e inicio del CGI	42
4.1. Instalación de Flask	54
4.2. Autoejecución del servicio RESTful	56

CAPÍTULO 1

Motivación y objetivos del trabajo

En este primer capítulo exponemos las razones que nos han conducido a la elaboración de este trabajo, así como los objetivos que pretendemos alcanzar y una pequeña descripción de los capítulos que forman la memoria del trabajo realizado.

1.1 Motivación

Tal como afirma Fernández Alarcón en [5], un sistema es un conjunto de componentes que interactúan entre sí para lograr un objetivo común. Aunque existe una gran variedad de sistemas, la mayoría de ellos pueden ser representados mediante un conjunto de cinco bloques básicos: elementos de entrada, elementos de salida, sección de transformación, mecanismos de control, y objetivos; además, podemos destacar que la sociedad actual está llena de ejemplos de sistemas: una máquina expendedora, una fábrica de productos manufacturados, un automóvil, un archivador de documentos, etc.

En la actualidad, la expresión *Sistema de Información* (SI) se utiliza de forma común y habitual en organizaciones de toda índole; sin embargo, existen tantas definiciones y matices para ella como escuelas o autores del tema. Aun así, y basándonos en la definición de sistema, se podría realizar una pequeña aproximación definiéndola como un conjunto de componentes que interactúan entre sí para lograr un objeto común: satisfacer las necesidades de información de una organización; de esta forma, podemos considerar los datos de dicha organización como la materia prima que usará el sistema, el cual, gracias a un conjunto de técnicas desarrolladas en el ámbito de las Tecnologías de la Información y de las Comunicaciones (TIC), podrán ser procesados de forma digital a través de todos los elementos o fases correspondientes a la definición de sistema.

1.2 Objetivos

Una vez hemos obtenido una idea clara de lo que entendemos por un Sistema de Información, establecemos que el objetivo principal de este proyecto será el diseño y puesta en marcha de un Sistema de Información para una PYME¹.

Entre los diferentes componentes que formarán parte de este proyecto, podemos destacar la necesidad de poner en marcha un servidor destinado al procesamiento y almacenaje de la información tratada por el sistema. Además, nos centraremos en el diseño de dos aplicativos integrantes del sistema de información que nos permitan operar con el mismo, de los cuales diferenciamos la necesidad de diseñar y programar un aplicativo de gestión administrativa tanto del servidor como del propio sistema de información, así como otro aplicativo más ligero que permita la consulta ágil dentro del propio sistema de información.

Una vez definidos los objetivos de este proyecto, destacamos que la PYME con la que vamos a colaborar para la consecución del mismo es la empresa Embalajes Industriales Ramírez Tomás S.L.², la cual nos ha expuesto la necesidad de digitalización y automatización de su sistema de referenciado y catalogación de sus productos; además, será dicha empresa la que asumirá los gastos económicos del proyecto, y se beneficiará del sistema resultante tras la finalización de este trabajo.

Con respecto al desarrollo de los aplicativos mencionados anteriormente, podemos diferenciar el diseño y programación de un portal web mediante el cual podremos llevar a cabo las tareas de administración tanto del servidor como del Sistema de Información. Para el caso del aplicativo ligero de consulta, desarrollaremos una aplicación móvil multiplataforma con el fin de que los operarios de la empresa que hagan uso del Sistema de Información puedan realizar consultas al mismo desde dispositivos móviles que porten consigo en sus labores cotidianas.

Finalmente, cabe la pena destacar que este proyecto, al estar orientado a organizaciones que no disponen de grandes medios ni operacionales ni económicos, tratará de realizar desarrollos basados en tecnologías *Open Source*³, así como el despliegue de dichas tecnologías en sistemas hardware acorde a las necesidades y posibilidades de la organización.

Una vez explicitados los objetivos de este proyecto, podemos sintetizarlos de tal forma que queden resumidos en los siguientes puntos:

¹La pequeña y mediana empresa (conocida también por el acrónimo PyME, lexicalizado como pyme, o por la sigla PME) es una empresa con características distintivas, la cual tiene dimensiones con ciertos límites ocupacionales y financieros.

²Embalajes Industriales Ramírez Tomás S.L. es una empresa dedicada a la elaboración y manufacturación de embalajes de tipo industrial basados en plataformas y estructuras de madera. Página web: <https://www.ramireztomas.es/>.

³El código abierto (*Open Source*) es el software distribuido y desarrollado de forma libre.

- Estudio de las necesidades de digitalización de información de la empresa.
- Montaje de un servidor para el almacenamiento de la información.
- Diseño de una aplicación web para la administración y gestión tanto del servidor como del Sistema de Información.
- Diseño de una aplicación móvil de consulta ágil para el Sistema de Información.

1.3 Estructura de la memoria

El presente trabajo se compone de seis capítulos, y en esta sección nos disponemos a realizar una breve descripción del trabajo realizado en cada uno de ellos.

- ★ **Capítulo 1, Motivación y objetivos del trabajo:** En este capítulo se expone la motivación por la cual se ha llevado a cabo este trabajo, así como los objetivos esenciales a abarcar.
- ★ **Capítulo 2, Necesidades de digitalización de información de la PYME:** Este capítulo estará orientado a llevar a cabo una valoración de las necesidades de digitalización de la empresa con el fin de determinar qué servicios y recursos poner en marcha para la correcta consecución de los objetivos planteados. Además definiremos los elementos hardware que formarán parte de nuestro proyecto, de los cuales realizaremos una breve introducción.
- ★ **Capítulo 3, Diseño del sistema de información:** En este capítulo llevaremos a cabo tanto la definición del presupuesto del proyecto, como la puesta en marcha del servidor y del portal web de administración del mismo, el cual utilizarán los usuarios y trabajadores encargados de la administración de este sistema de información.
- ★ **Capítulo 4, Diseño de aplicación móvil de usuario:** Este capítulo está orientado al diseño y programación de una aplicación móvil multiplataforma mediante la cual los usuarios y trabajadores de la empresa podrán realizar consultas a las bases de datos de nuestro servidor.
- ★ **Capítulo 5, Presupuesto del proyecto:** Este capítulo está orientado a confeccionar un presupuesto global del proyecto teniendo en cuenta tanto los gastos del material necesario, como el desempeño en horas de trabajo realizadas.
- ★ **Capítulo 6, Estudios de consumo y valoración de los usuarios:** En este capítulo llevaremos a cabo una serie de estudios mediante los cuales trataremos de realizar un análisis tanto del consumo como de la capacidad de

procesamiento de nuestro servidor, una vez esté operativo, y en el entorno de trabajo en el cual será utilizado tras la finalización de este proyecto.

- ★ **Capítulo 7, Conclusiones:** El último capítulo recoge las principales conclusiones obtenidas, así como un breve resumen del conjunto de funcionalidades implementadas a lo largo del proyecto, incluyendo además comentarios acerca de los problemas sufridos, o el planteamiento de futuras mejoras.

Finalmente, cabe destacar la presencia de varios anexos al final de la memoria, en los cuales se incluyen los códigos de más relevancia implementados durante la consecución del trabajo, entre los cuales podemos diferenciar tanto los códigos pertenecientes al portal web de administración del servidor y del Sistema de Información, como los códigos pertenecientes a la aplicación móvil multiplataforma de consulta ágil del Sistema de Información.

1.4 Uso de la bibliografía

En esta sección realizamos una breve descripción de los materiales bibliográficos utilizados, así como su uso y relación en cada uno de los apartados del trabajo.

- Al comienzo del proyecto hemos realizado algunas definiciones básicas sobre el concepto de un sistema y de un sistema de información, para lo que nos hemos apoyado en las definiciones y explicaciones encontradas en [5].
- Uno de los apartados iniciales del proyecto es la definición de los elementos hardware de los que haremos uso, de los cuales llevamos a cabo una pequeña introducción, para lo que hemos hecho uso de las referencias [6, 13, 15].
- Entre los objetivos de este proyecto encontramos la instalación y configuración de un servidor de bajo presupuesto para contener las bases de datos en las que se almacenarán los datos de nuestro sistema de información, para lo cual hemos hecho uso de las referencias bibliográficas [3, 4, 8, 11, 12].
- Por otro lado, durante la consecución del proyecto diseñamos y programamos un portal web de administración remota, para el cual hacemos uso de las referencias [2, 9, 14].
- Finalmente hemos diseñado una aplicación móvil multiplataforma para consultar información del servidor, en cuyo proceso hemos consultado las referencias [1, 7, 10].

CAPÍTULO 2

Necesidades de digitalización de información de la PYME

En este capítulo nos disponemos a llevar a cabo una valoración de las necesidades de digitalización de la empresa con el fin de determinar que servicios y recursos poner en marcha para la correcta consecución de los objetivos planteados.

2.1 Análisis de requerimientos

Como hemos comentado anteriormente, este proyecto se va a basar en el diseño e implementación de un Sistema de Información para una PYME, en concreto para la PYME Embalajes Industriales Ramírez Tomás S.L., cuyo logotipo podemos observar en la Figura 2.1, y la cual se dedica a la elaboración y manufacturación de embalajes de tipo industrial basados en plataformas y estructuras de madera.

Embalajes Industriales Ramírez & Tomás, S.L.

Figura 2.1: Logo de Embalajes Industriales Ramírez Tomás S.L.

Actualmente, en cuanto al procedimiento de trabajo de esta empresa, tal como podemos observar en la Figura 2.2, nos encontramos con la situación de que esta misma tiene una bolsa de clientes que de forma periódica se ponen en contacto con ellos para realizar pedidos de diferentes productos cuyas referencias de manufacturación (planos, guías de construcción, medidas, referencias, etc.) se encuentran en formato físico, y comúnmente en papel impreso en los archivadores de la oficina de administración. Suele ocurrir que para cada nuevo pedido los trabajadores encargados de comenzar con el diseño del producto solicitado deben

acudir a los archivos físicos y buscar las referencias del producto entre una gran cantidad de documentos.

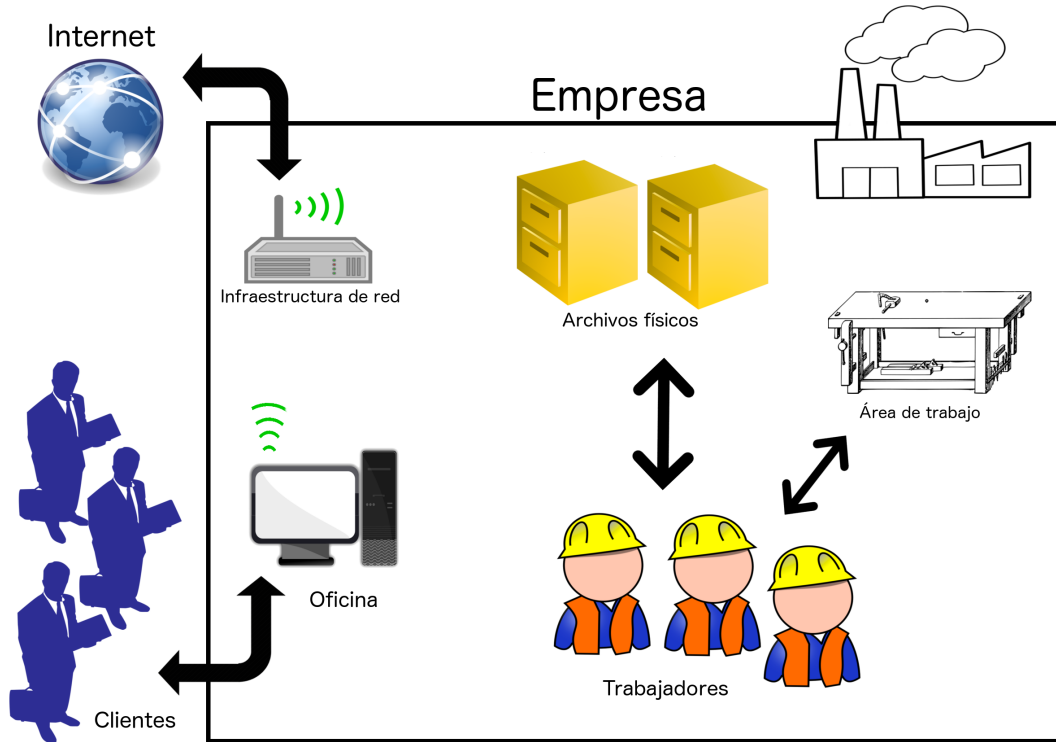


Figura 2.2: Sistema de trabajo original

Es por ello que se establece la necesidad de implantación de un sistema digital de almacenamiento de esta información, de tal forma que además de ahorrar una gran cantidad de tiempo a los trabajadores a la hora de buscar las referencias de un producto, seremos capaces de trabajar con las mismas de forma dinámica, pudiendo modificarlas de forma sencilla en cuanto nos sea requerido, y quedando estas siempre accesibles para los trabajadores.

2.2 Servicios a desplegar

De esta forma nos disponemos a diseñar un Sistema de Información especialmente adaptado a las necesidades de esta empresa, entre las cuales destacamos, tal como hemos mencionado en apartados anteriores, la capacidad de tener un registro de clientes y productos, en los que asociaremos a cada cliente los productos que le son referidos, y en los cuales podremos guardar la información de dicho tipo de producto.

Para llevar a cabo el desarrollo de este Sistema de Información hemos realizado una valoración de las necesidades de digitalización de la empresa, tras la

cual determinamos que las necesidades de cómputo de la misma no son muy elevadas debido a que no existe una gran cantidad de trabajadores que harán uso de este Sistema de Información, estando estos en todo caso por debajo de la decena, y requiriendo tareas únicamente de consulta ligera a bases de datos en su mayoría. Es por ello que nos disponemos a habilitar, tal como muestra la Figura 2.3, un servidor físico de bajo coste y bajo consumo. En concreto haremos uso de una Raspberry Pi, que sin lugar a dudas será suficiente para cumplir con las necesidades de digitalización de esta organización. Este servidor será implantando en un computador de bajo coste que hará uso de un sistema operativo Linux, que además contará con servicios básicos para el funcionamiento del Sistema de Información que implantaremos más adelante, entre los que podemos destacar algunos como un servicio de base de datos, un servicio de FTP, o un servicio web.

Además, atendiendo a la propuesta de la empresa consistente en habilitar un mecanismo de consulta ágil y rápida para los operarios de la misma, nos disponemos a diseñar una aplicación móvil multiplataforma para otorgar la posibilidad de que estos instalen en sus terminales dicha aplicación para la consulta ágil de las bases de datos del servidor.

Finalmente cabe la pena destacar que este servidor que vamos a desplegar será situado de forma física en las oficinas de la empresa, estando este siempre accesible en la red interna de la misma, y al alcance físico de los administradores del Sistema de Información.

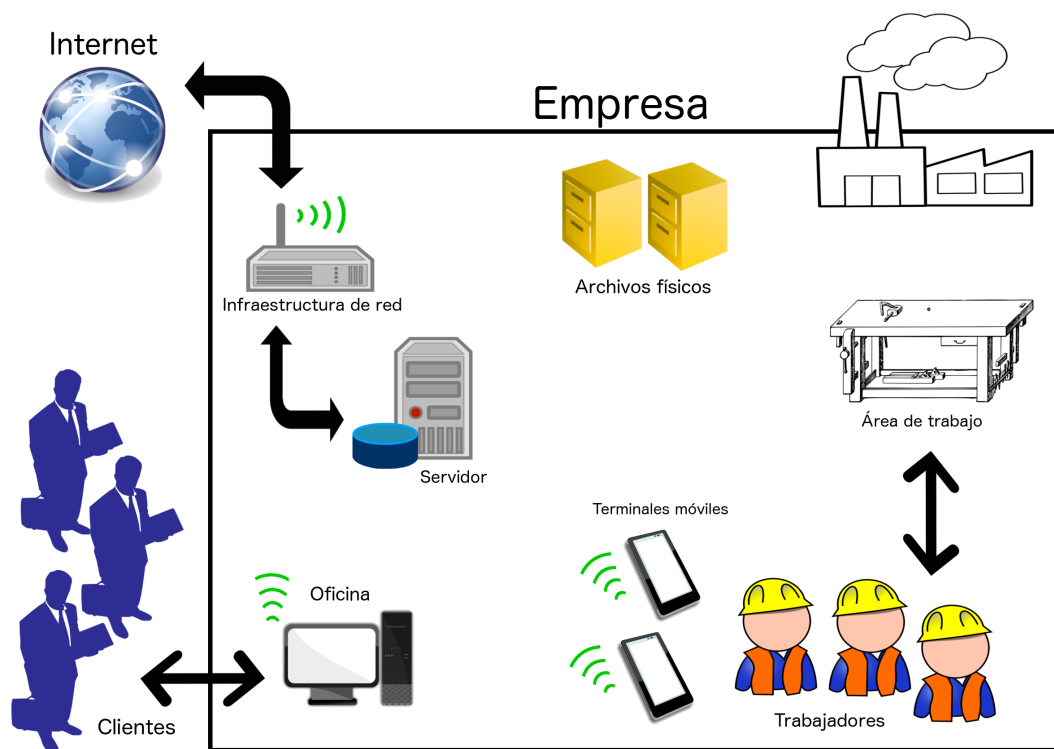


Figura 2.3: Sistema de trabajo modificado

2.2.1. Introducción a Raspberry Pi

La Raspberry Pi es una placa computadora (SBC¹) de bajo coste desarrollada desde 2012 en Reino Unido por la Raspberry Pi Foundation, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas. De esta forma fue diseñada con el fin de ser lo más barata posible y poder llegar al máximo número de usuarios.

Con unas dimensiones de 86 mm de largo x 56 mm de ancho x 20 mm de alto, un peso de 45 g, y un precio aproximado de 40€ en su versión más completa, la Raspberry Pi se presenta como una revolución dentro del mundo de los dispositivos empujados. Dentro del repertorio de posibilidades que encontramos con este dispositivo podemos destacar algunas como usarlo a modo de NAS, de servidor de aplicaciones distribuidas, o de servidor de correo. También hace las veces de computador de sobremesa, por lo que podemos realizar con él tareas tales como reproducir vídeos en alta definición, reproducir música, utilizar aplicaciones de ofimática, o navegar por la red, eso sí, debido a su limitada potencia no podemos esperar un gran rendimiento. Además, uno de los usos más extendidos y que más éxito han tenido con este dispositivo ha sido su integración en sistemas robóticos a modo de controlador central, o usarlo para emular sistemas recreativos arcade.

La popularidad de Raspberry Pi con estos fines reside en la disminución de los costes de un proyecto en el que debido a su naturaleza, o a las necesidades tecnológicas del mismo, suelen acarrear grandes costes durante su desarrollo; además, su reducido tamaño es un punto a su favor, ya que actualmente uno de los objetivos de la tecnología no es solo la eficiencia de los productos, sino que además, es muy valorado el tamaño de los mismos.

Finalmente cabe destacar que la Raspberry Pi no viene con un manual de fábrica, ni con unas instrucciones de instalación ni con nada parecido, por lo que para el usuario final puede aparentar la vuelta a una era en la que cada uno debía apañárselas para administrar sus dispositivos, era en la cual se aprendía investigando, tocando ajustes y probando qué pasaba, cómo funcionaba mejor, o investigando cuáles eran las mejores soluciones para hacer que nuestras máquinas hicieran lo que nosotros pretendíamos. En definitiva, la Raspberry Pi nos remonta a otra época, en la que el aprendizaje y la investigación eran necesarios para conseguir un buen rendimiento de los dispositivos que construimos o administramos.

¹Un SBC (*Single Board Computer*) es un computador completo construido en una única placa de circuitos, con microprocesador, entradas y salidas de datos, y otras características propias de un computador funcional.

Hardware

En la Raspberry Pi 3 Modelo B, que es la que vamos a utilizar en nuestro proyecto, y que se muestra en la Figura 2.4, podemos encontrar unas características muy interesantes. En su corazón nos encontramos con un chip integrado Broadcom BCM2837, que contiene un procesador Quad Cortex A53 con varias frecuencias de funcionamiento y la posibilidad de subirla (*overclocking*) hasta 1,2 GHz, un procesador gráfico VideoCore IV a 400 MHz, y además, las últimas Raspberry Pi cuentan con 1 GB de memoria RAM. Todo ello equivale en la práctica a un ordenador con unas capacidades gráficas similares a la Xbox de Microsoft y con la posibilidad de reproducir vídeo en Full HD.

En la placa nos encontramos además con una salida de vídeo y audio a través de un conector HDMI, con lo que conseguiremos conectar la tarjeta tanto a televisores como a monitores que cuenten con dicha conexión. En cuanto a vídeo se refiere, también cuenta con un conector DSI para conectar una pantalla externa, y una salida de audio a través de un conector *minijack*. Posee una conexión Ethernet 10/100, así como conexión WiFi 802.11n y Bluetooth 4.1 BLE. Además disponemos de cuatro puertos USB en los que podremos conectar cualquier tipo de periférico compatible, y otro conector DSI en el que podremos conectar una cámara de vídeo.

En cuanto al tipo de unidades de almacenamiento de este dispositivo, hay que comentar que no tiene ninguno. Pero en su parte inferior, cuenta con un lector de tarjetas MicroSD, lo que abarata enormemente su precio y da la posibilidad de instalar un sistema operativo en una tarjeta de memoria de 2 GiB o más (clase 4 o mejor). De esta forma tenemos también la posibilidad de minimizar el espacio que necesitamos para tener todo un ordenador en un volumen mínimo. Para su alimentación podemos usar un cargador con conexión micro USB que pueda dar una corriente de al menos 750 mA.

Software

La Raspberry Pi utiliza mayoritariamente sistemas operativos basados en el núcleo Linux, aunque actualmente podemos instalar un núcleo reducido del sistema operativo Windows, denominado Windows IoT Core, o incluso meter con calzador cualquier sistema al que le baste 1 GB de memoria RAM para funcionar con fluidez. Entre las distribuciones más comunes podemos encontrar las de propósito general, de las cuales destacamos las siguientes, y cuyos logos se muestran en la Figura 2.5:

- Raspbian: quizás la distribución Linux más conocida para Raspberry Pi, un sistema operativo basado en Debian que nos ofrece un entorno tanto en modo consola como en escritorio, con el que podremos programar o jugar. A través de la Pi Store, la tienda de aplicaciones de Raspberry Pi, podremos

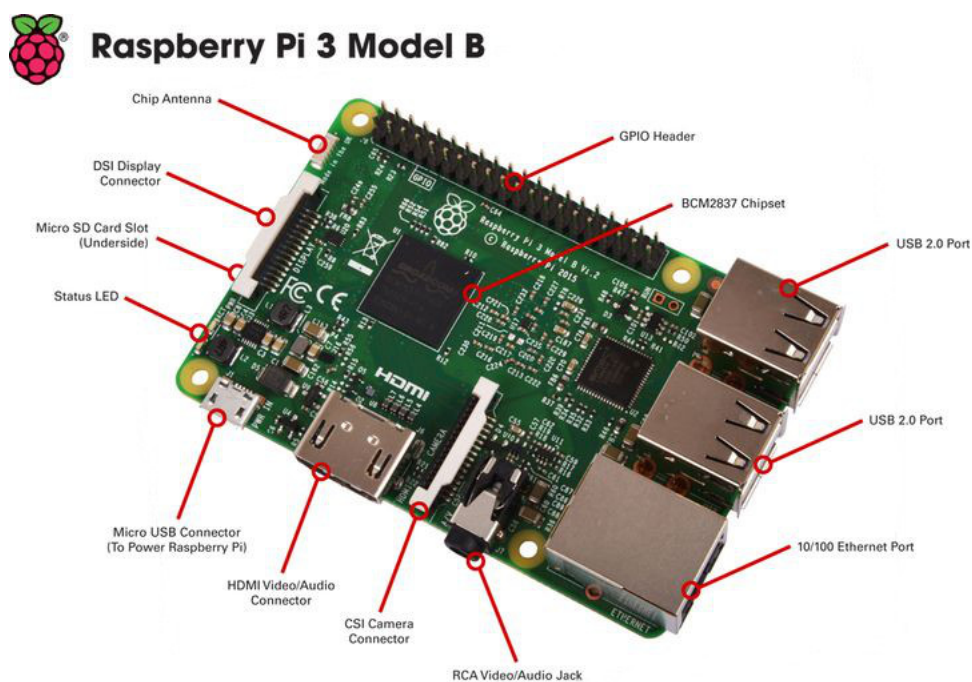


Figura 2.4: Raspberry Pi 3 Modelo B

acceder a múltiples aplicaciones tanto gratuitas como de pago y, por ejemplo, podremos instalar aplicaciones de ofimática tales como LibreOffice.

- Ubuntu Mate: una versión de escritorio Ubuntu optimizada para ser usada en la Raspberry Pi.
- Moebius: es una distribución que se caracteriza por ser extremadamente ligera, y por tener el servicio ssh activado por defecto.
- Arch Linux ARM: es otra opción simple y ligera que también podemos usar como sistema operativo base para nuestros proyectos.
- Windows IoT Core: esta versión reducida del sistema operativo Windows para la *Internet of Things* permite la integración de este dispositivo con las tecnologías de desarrollo de código propias de Windows, mediante las cuales podremos programar la placa, llevar a cabo su configuración interna, o incluso descargar aplicaciones de terceros ya desarrolladas.

Por otra parte, también podemos encontrar distribuciones orientadas a usos más concretos. De la misma forma que podemos encontrar distribuciones Linux que se han diseñado para acometer funciones concretas (*firewalls*, sistemas de *backup*, etc), en el mundo de este dispositivo empotrado también podemos encontrar distribuciones optimizadas para un entorno o ámbito de aplicación concreto, que nos permiten transformar nuestra Raspberry Pi en una máquina arcade, en un centro de seguridad, en un centro domótico, o incluso en un Media Center:

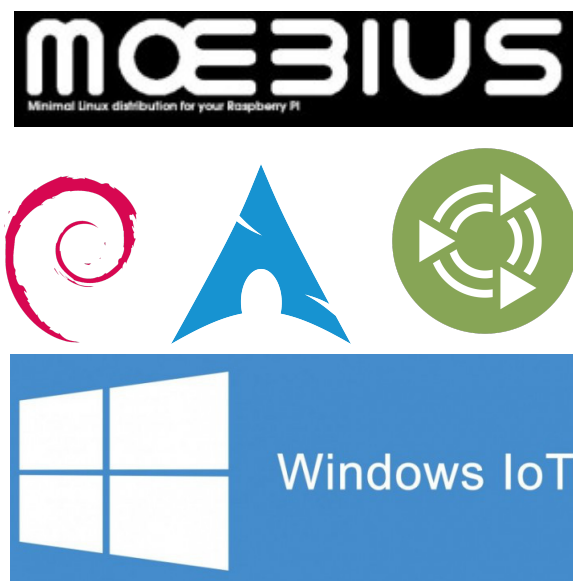


Figura 2.5: Logos de Moebius, Raspbian, ArchLinux, Ubuntu Mate, y Windows 10 IoT, respectivamente.

- ArkOS: es una distribución que ha nacido gracias al *crowdfunding*² con el objetivo de ofrecernos todas las herramientas necesarias para desplegar un servidor privado para nuestros archivos. Con ArkOS solamente necesitaremos una Raspberry Pi, una unidad de almacenamiento USB, y por supuesto, una tarjeta SD con la distribución. De esta forma, podemos disponer de un servidor privado que administramos según nuestras necesidades.
- OSMC (*Open Source Media Center*): es una distribución muy ligera con la que podremos implementar un Media Center sobre nuestra Raspberry Pi. Teniendo en cuenta que Raspberry Pi posee una salida HDMI, la idea es conectarla a nuestro televisor para usarla como reproductor multimedia de bajo coste con una distribución especialmente optimizada para este tipo de tareas y con un interfaz orientado también a hacernos sencillo su manejo.
- PiMAME: es una distribución que transformará nuestra Raspberry Pi en una máquina arcade; concretamente, esta distribución nos ofrece emuladores como MAME y algunos otros emuladores de consolas clásicas como Neo Geo, PlayStation, Super NES o Atari 2600. La distribución se apoya en Raspbian pero, cara al usuario, pone las cosas muy sencillas e implementa un servidor FTP para que nos sea fácil llevar las ROM de los juegos a la tarjeta SD del computador.

²*Crowdfunding*, microfinanciación colectiva: es la cooperación colectiva llevada a cabo por diferentes personas para conseguir dinero u otros recursos.

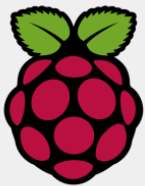
	Raspberry Pi 3 Model B	Raspberry Pi Zero	Raspberry Pi 2 Model B	Raspberry Pi Model B+
Introduction Date	2/29/2016	11/25/2015	2/2/2015	7/14/2014
SoC	BCM2837	BCM2835	BCM2836	BCM2835
CPU	Quad Cortex A53 @ 1.2GHz	ARM11 @ 1GHz	Quad Cortex A7 @ 900MHz	ARM11 @ 700MHz
Instruction set	ARMv8-A	ARMv6	ARMv7-A	ARMv6
GPU	400MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV
RAM	1GB SDRAM	512 MB SDRAM	1GB SDRAM	512MB SDRAM
Storage	micro-SD	micro-SD	micro-SD	micro-SD
Ethernet	10/100	none	10/100	10/100
Wireless	802.11n / Bluetooth 4.0	none	none	none
Video Output	HDMI / Composite	HDMI / Composite	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI	HDMI / Headphone	HDMI / Headphone
GPIO	40	40	40	40
Price	\$35	\$5	\$35	\$35

Figura 2.6: Comparativa entre modelos de Raspberry Pi actuales

Especificaciones técnicas

En la Figura 2.6 mostramos las diferencias en cuanto a características técnicas existentes entre los cuatro modelos de Raspberry Pi más comunes en la actualidad. Destacamos además la alta velocidad de mejora de estos dispositivos, ya que en apenas dos años hemos pasado de disponer de placas con 512 MB de memoria RAM y sin conectividad *wireless* integrada, a placas con el doble de memoria RAM y con WiFi y Bluetooth integrados, así como el aumento considerable en la frecuencia y capacidad de cómputo del procesador.

CAPÍTULO 3

Diseño e implementación del Sistema de Información

En este capítulo vamos a hablar acerca de los elementos que van a formar parte de nuestro Sistema de Información, entre los cuales distinguimos una Raspberry Pi, que realizará la función de computador central y servidor, un portal web destinado a la configuración interna del servidor, y una aplicación móvil mediante la cual los usuarios se podrán conectar al servidor para realizar las consultas propias de sus tareas laborales.

3.1 Computador central

Tal como hemos comentado anteriormente, vamos a hacer uso de una Raspberry Pi 3 Modelo B para que haga la función de computador central y servidor de nuestro sistema de información. Entre las funciones asociadas a dicho computador central encontramos el almacenamiento en las bases de datos que contendrán la información de los usuarios que estén autorizados a acceder al sistema de configuración, y la propia información introducida en el sistema por los mismos. Además, el computador central almacenará el software encargado de recibir y contestar las peticiones de usuario desde las aplicaciones móviles, y realizará la función de intermediario entre la base de datos y dichas aplicaciones.

Otra de las funciones del computador central será almacenar y ofrecer un sistema de administración telemático, mediante el cual, los usuarios que sean administradores del sistema, podrán acceder al mismo de forma remota, con el fin de realizar modificaciones en las bases de datos del sistema de información, así como para realizar cambios en la base de datos de usuarios, lo que les permitirá añadir o eliminar usuarios registrados para que estos tengan, o dejen de tener, los permisos necesarios para interactuar con el sistema. Dicha funcionalidad se explica con detenimiento en la sección 3.2.

3.1.1. Sistema operativo

Todas las funcionalidades descritas previamente precisan de la gestión y control de un SO¹, y como se ha indicado en secciones anteriores, la Raspberry Pi nos ofrece la posibilidad de integrar y trabajar con cualquier SO que requiera menos memoria de la integrada en la placa de la Raspberry Pi para poder funcionar. En este proyecto, hemos decidido trabajar una distribución muy popular de SO llamada Raspbian

Raspbian:

Raspbian, cuyo logo podemos observar en la Figura 3.1, es el nombre dado a una variante modificada de la popular distribución de Linux Debian. Debian es una de las más longevas distribuciones de Linux, y concentra una gran compatibilidad y un rendimiento excelente incluso con un hardware modesto, lo que lo convierte en una gran opción para la Raspberry Pi. Raspbian toma como base (o *distribución madre*) el SO Debian, y añade herramientas y software personalizado para conseguir trabajar con la Raspberry Pi de la forma más sencilla posible.

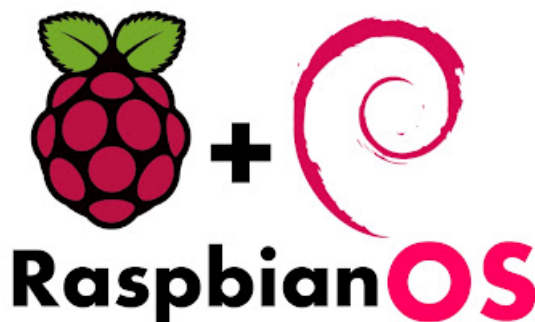


Figura 3.1: Logo del sistema operativo Raspbian

Para conseguir un consumo mínimo de recursos, la imagen de Raspbian para la Raspberry Pi (descargable, tal como podemos observar en la Figura 3.2, desde la página web oficial de Raspbian²) incluye únicamente una pequeña parte del software que encontraríamos en una versión regular de un SO para escritorio. Se incluyen herramientas para navegar por Internet, programar en Python, o una interfaz gráfica para trabajar de forma más amena. Adicionalmente, se ofrece la posibilidad de instalar más software de forma rápida mediante el gestor de paquetes de la distribución, bien a través de un entorno de consola (comando *apt*), o a través del programa de interfaz gráfica *Raspberry Pi Store*.

¹Un sistema operativo (SO) es un programa o conjunto de programas de un sistema informático que gestiona los recursos de hardware y provee servicios a los programas de aplicación, ejecutándose en modo privilegiado respecto de los restantes.

²Raspbian - descargas: <http://www.raspbian.org/RaspbianImages>

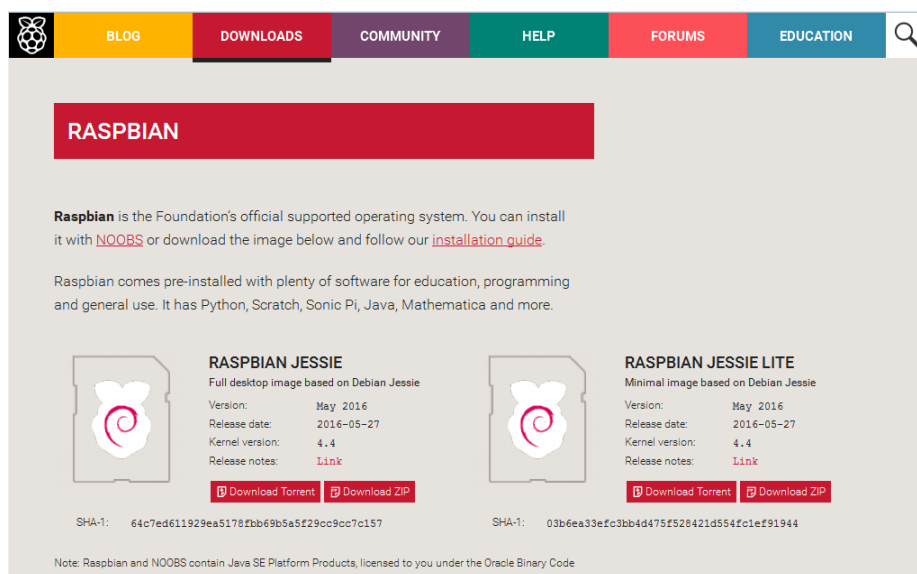


Figura 3.2: Descarga oficial de Raspbian

Raspbian incluye desde hace relativamente poco tiempo un nuevo entorno de escritorio conocido como el *Pixel Desktop Environment* el cual está diseñado para ofrecer una interfaz atractiva mediante un sistema de ventanas e interfaces similar al de otros sistemas operativos tales como Microsoft Windows u OSX, que ofrece un menor rendimiento y resultados que el anterior *Lightweighth X11 Desktop Environment*.

Para llevar a cabo la instalación del sistema operativo en nuestro dispositivo podemos hacer uso de cualquier tipo de herramienta que nos permita realizar el montaje de la imagen del sistema operativo sobre la tarjeta MicroSD que hará las veces de sistema de almacenamiento de nuestro sistema de información. En nuestro caso hemos hecho uso, tal como mostramos en la Figura 3.3, del software *Win32 Disk Imager*³

3.1.2. Configuración base del servidor

En esta sección llevaremos a cabo una serie de configuraciones base mediante las cuales dejaremos el servidor listo para proceder a la instalación del conjunto de servicios necesarios para ofrecer la funcionalidad del mismo, además realizaremos cambios en la configuración del sistema que contribuirán a la mejor accesibilidad al dispositivo cuando conectemos el servidor a nuestra red local.

³Web de descarga oficial de Win32 Disk Imager: <https://sourceforge.net/projects/win32diskimager/files/Archive/>

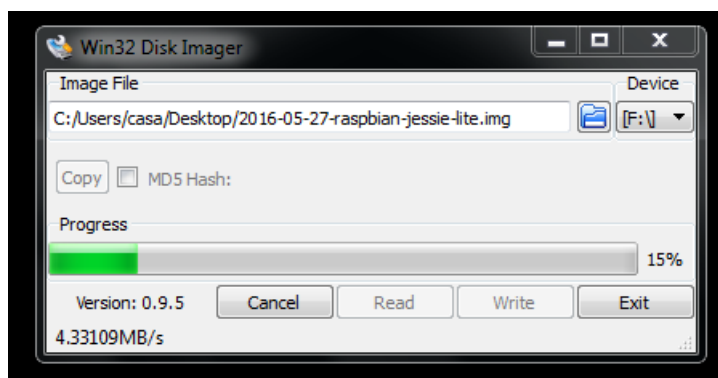


Figura 3.3: Montaje de la imagen de sistema operativo

Configuración de la dirección IP

Para mejorar el acceso al servidor, y con el fin de realizar de una forma más sencilla las comunicaciones entre este mismo y las aplicaciones que se desarrollarán de forma posterior, vamos a dotar a la Raspberry Pi de una dirección IP fija, de tal forma que al conectar el dispositivo a una red local este estará siempre accesible bajo esta dirección; siempre y cuando no haya un solape de dirección, por lo que llevaremos especial cuidado con la elección de esta misma, estableciendo una dirección de rango alto.

Basándonos en la dirección de la puerta de enlace del *router* en el que conectaremos el servidor (la cual es 192.168.1.1/24), hemos determinado hacer uso de la dirección de red 192.168.1.200/24. Para ello, tras haber procedido al montaje de la imagen del sistema operativo en la tarjeta MicroSD de almacenamiento de la Raspberry Pi, la conectaremos a la corriente eléctrica y esperaremos a que se inicie el sistema. Una vez hecho esto nos conectaremos al dispositivo, y tal como muestra la Figura 3.4, procederemos a editar el fichero `etc/network/interfaces`, marcando la dirección del adaptador de red `eth0` como *manual*.

Tras haber indicado que la configuración del adaptador de Ethernet sea local, procederemos a editar el fichero `etc/dhcpd.conf` en el cual, tal como podemos observar en la Figura 3.5, indicaremos que la dirección asociada a dicha interfaz de red sea estática con valor 192.168.1.200/24, que tanto la puerta de enlace como la dirección de nuestro DNS⁴ principal sea la 192.168.1.1.

Configuración del Hostname

Además de llevar a cabo la edición de la dirección de red de nuestro dispositivo, vamos a proceder a editar el nombre de red del mismo (*hostname*), de tal forma que facilitaremos el acceso al dispositivo en caso de que algún servicio o aplicación se disponga a buscar el servidor mediante el nombre de equipo en vez

⁴Un servidor DNS (*Domain Name System* - Sistema de nombres de dominio) es un servidor que traduce nombres de dominio a IPs y viceversa.


```
GNU nano 2.2.6 File: /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

auto eth0
allow-hotplug eth0
iface eth0 inet manual

allow-hotplug wlan0
iface wlan0 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Figura 3.4: Configuración de la dirección IP (interfaces)

de hacer uso de la dirección IP. Para ello, tal como muestra la Figura 3.6, haremos uso de la orden de consola `sudo hostnamectl set-hostname SISERVER` para modificar el nombre de red del servidor, acción que podremos comprobar mediante la orden de consola `hostnamectl`.

Además, para asegurar la correcta correlación entre el nombre de red y la dirección de red del dispositivo, editaremos tal como podemos observar en la Figura 3.7, el fichero `etc/hosts` indicando que el nombre asociado a la dirección de red de nuestra máquina local (*localhost*) será `SISERVER`.

```
GNU nano 2.2.6      File: /etc/dhcpd.conf      Modified
# Most distributions have NTP support.
option ntp_servers
# Respect the network MTU.
# Some interface drivers reset when changing the MTU so disabled by default.
#option interface_mtu

# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate Stable Private IPv6 Addresses instead of hardware based ones
slaac private

# A hook script is provided to lookup the hostname if not set by the DHCP
# server, but it should not be run by default.
nohook lookup-hostname

interface eth0
static ip_address=192.168.1.200/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
```

Figura 3.5: Configuración de la dirección IP (dhcpd.conf)

```
pi@raspberrypi:~$ sudo hostnamectl set-hostname SISERVER
pi@raspberrypi:~$ hostnamectl
  Static hostname: siserver
    Pretty hostname: SISERVER
      Icon name: computer
        Chassis: n/a
    Machine ID: 8c5604e85c924195aef34a3e4079998d
      Boot ID: 6c8a03934d1440a1b8e23beb8da1505b
  Operating System: Raspbian GNU/Linux 8 (jessie)
      Kernel: Linux 4.4.50-v7+
    Architecture: arm
pi@raspberrypi:~$
```

Figura 3.6: Configuración del hostname (hostnamectl)

```
GNU nano 2.2.6      File: /etc/hosts
127.0.0.1    localhost
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters

127.0.1.1   siserver
```

Figura 3.7: Configuración del hostname (hosts)

3.2 Gestión remota

Uno de los objetivos principales del proyecto es administrar el sistema de forma remota, de tal forma que una persona autorizada pueda acceder al sistema de gestión a través de cualquier dispositivo con conexión a Internet, y tras autenticarse con sus credenciales de acceso, sea capaz de realizar gestiones relativas a la administración del sistema de información, tareas tales como editar la base de datos de los usuarios a los que se permite utilizar el sistema, añadir o eliminar entradas en la base de datos de la organización, o realizar copias de seguridad de las bases de datos.

3.2.1. Diseño del portal de administración remoto

Para diseñar un portal de administración web seguro hemos optado por hacer uso de los lenguajes de programación HTML y PHP, de tal forma que mediante una combinación de ambos, obtendremos los recursos y la funcionalidad necesaria para diseñar un portal a medida que satisfaga nuestras necesidades. Además, en cuanto a la sección de programación HTML, haremos uso del *framework* de edición web Bootstrap⁵, el cual nos permite combinar las tecnologías HTML, CSS y Javascript, de una forma sencilla.

El punto de partida para la administración telemática, y la primera sección de nuestro portal de administración, será una página de *login*, es decir, una página en la que el usuario podrá indicar su nombre de usuario y contraseña con el fin de obtener el acceso autorizado al portal, desde el cual podrá realizar los trámites pertinentes. Cabe destacar que la base de datos que contendrá los datos de acceso de los usuarios autorizados a realizar trámites en el portal será una base de datos diferente a la utilizada para almacenar las credenciales de los usuarios autorizados a usar la máquina arcade. Esta es una medida de seguridad adicional, la cual nos permitirá, además, ampliar el abanico de tecnologías utilizadas en el proyecto, ya que dicha base de datos será implementada en MYSQL debido a la facilidad de interacción de este sistema de base de datos con los lenguajes de programación HTML y PHP, que van a ser los lenguajes mediante los cuales diseñaremos el portal de administración. De esta forma, para permitir al usuario autenticarse, crearemos una base de datos en MYSQL, y definiremos una tabla formada por dos campos, un nombre de usuario, y una contraseña asociada a él.

Una vez definidos los datos de acceso para el usuario autorizado a realizar trámites en el portal *online*, nos centramos en el diseño de la página de *login*, la cual contendrá, tal como se observa en la Figura 3.8, dos secciones identificadas: un formulario para que el usuario pueda introducir su nombre de usuario y contraseña con el fin de ser verificado en la base de datos, y una sección de información en la que situaremos un pequeño párrafo informativo acerca de las característi-

⁵<http://getbootstrap.com/>

cas del portal. Podemos ver el código completo de la página de acceso a nuestro portal en el Apéndice A.1.

The image shows a web page for user access. At the top left, it says 'SI Server EIR Servidor de administración'. At the top right is a logo for 'PALETS Y CAJAS DE MADERA Ramirez'. In the center, there is a login form with the title 'Acceso Usuarios'. The form contains two input fields: 'Usuario' and 'Contraseña', followed by a green button labeled 'Entrar'. To the right of the form, there is a text block that reads: 'Los datos de usuario para acceder a la sección de administración del servidor serán expedidos por el gerente de recursos informáticos de la empresa Embalajes Industriales Ramirez a los encargados de realizar dichos trámites.'

Figura 3.8: Página de acceso al portal de trámites telemáticos

Tras haber sido autenticado de forma satisfactoria en el sistema, el usuario accederá a la página de inicio del portal, la cual podemos observar en la Figura 3.9, y cuyo código podemos encontrar en el Apéndice A.6. Centrándonos en el diseño de la página de inicio del portal, encontramos un menú lateral con las diferentes acciones que puede realizar el usuario (Figura 3.10), un menú superior desde el cual el usuario podrá cerrar la sesión o volver en cualquier momento a la página de inicio (Figura 3.11), y una sección central en la que se nos muestra información personal del usuario.

Con respecto al conjunto de acciones que nos ofrece el menú lateral del portal, podemos encontrar acciones referidas a la gestión de clientes de la organización, acciones referentes a la administración de la base de datos, gestión de copias de seguridad, alta de nuevos usuarios del sistema, y a la gestión de usuarios tanto de la página web como de la aplicación móvil.

En cuanto a las acciones referidas a la gestión de clientes de la organización, tal y como podemos observar en la Figura 3.12, disponemos de las opciones de listar el registro de clientes existentes, añadir nuevos clientes a la base de datos, modificar la información referida a algún identificador de cliente, o eliminar uno de los mismos.

Al seleccionar la opción Registro de Clientes, tal y como podemos observar en la Figura 3.13, nos aparecerá un listado del conjunto de clientes introducidos en el sistema, así como toda la información asociada a cada uno de los identificadores existentes. Estos identificadores serán usados para asociar entradas de productos a cada uno de dichos clientes.

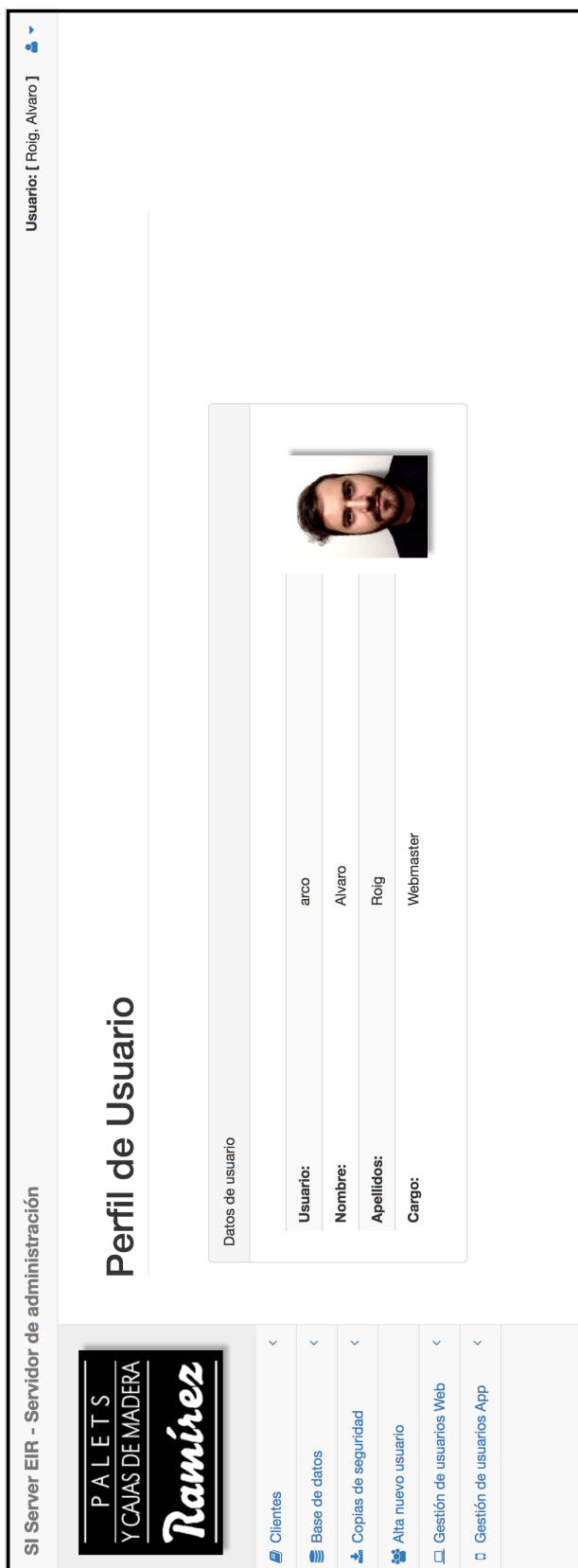


Figura 3.9: Página de inicio del portal de trámites telemáticos



Figura 3.10: Menú lateral del portal de trámites telemáticos

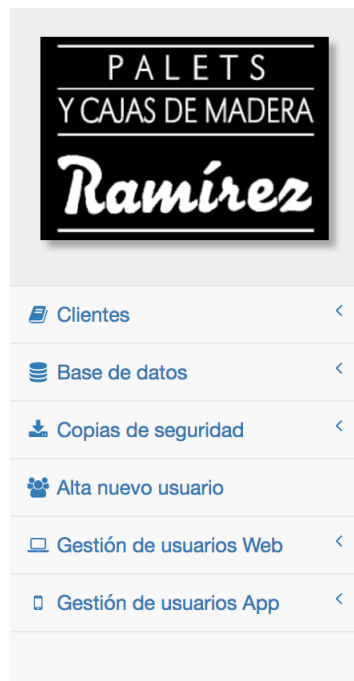


Figura 3.11: Menú superior del portal de trámites telemáticos

A su vez, con el fin de agregar nuevos clientes al sistema, seleccionaremos la opción *Alta Nuevo Cliente*, la cual nos conducirá a una página con un formulario en la que rellenar todos los campos necesarios para la inserción de un nuevo cliente en la base de datos. Podemos ver algunos de estos campos a rellenar en la Figura 3.14.

Por otra parte, el portal también nos da la posibilidad de eliminarlo del sistema, acción que realizaremos mediante la opción *Eliminar cliente* del menú lateral, en la que tal como observamos en la Figura 3.15, deberemos seleccionar uno de los que ya están presentes en el sistema, para que este sea eliminado posteriormente. Además destacar el hecho de que al eliminar un cliente de la base de datos también eliminaremos todas las entradas de la base de datos asociadas a dicho cliente.

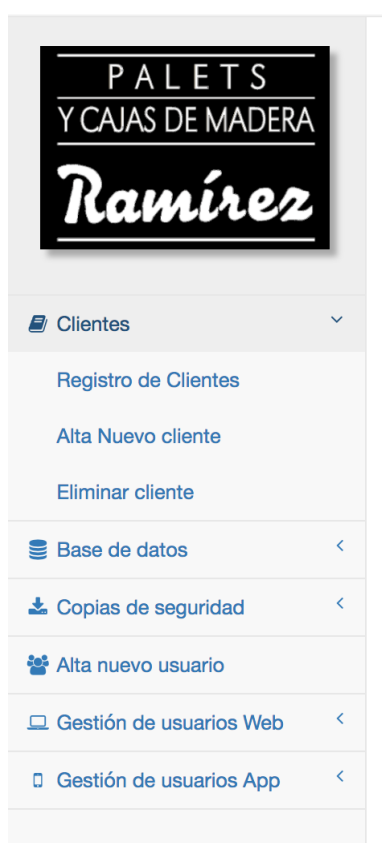


Figura 3.12: Acciones asociadas a la gestión de clientes

No solo disponemos de acciones referidas a la gestión de clientes en el sistema, ya tal y como podemos observar en la Figura 3.16, también disponemos de acciones que nos permiten gestionar las entradas de productos presentes en el sistema; entre ellas, diferenciamos las de listar el conjunto de entradas existentes, listar dichas entradas con un filtro por cliente, añadir una nueva entrada, así como modificar o eliminar una de las que ya disponemos en el sistema.

Para ver el conjunto de entradas de productos disponibles en el sistema nos dirigiremos a la sección *Listar* entradas del menú lateral, en la que tal y como muestra la Figura 3.17, encontraremos una lista de todos los productos registrados en el sistema. Además, se nos muestran algunos detalles tales como la fecha de alta de cada una de las entradas, el cliente al que está asociada cada una de ellas, así como su descripción, medidas, y una imagen o plano ilustrativo del producto que podremos descargar pulsando sobre su identificador de imagen. Adicionalmente disponemos de la posibilidad de realizar un listado de entradas aplicando un filtro por cliente, acción que podremos realizar desde la opción *Listar* entradas por cliente del menú lateral.

Otra de las acciones que nos permite el portal referente a la gestión de entradas de productos del sistema es la de añadir una nueva entrada. Esta acción la realizaremos desde la sección *Alta* nueva entrada del menú lateral; en dicha sec-

Registro de Clientes

Cientes registrados

Show entries Search:

ID	Cliente	Nº de entradas	Fecha de registro
13	Mulem	2	2016-08-04 13:07:58
14	Iberica	1	2016-08-04 13:08:01
15	Tecnicarton	0	2016-08-04 13:08:08
18	BombasIdeal	2	2016-08-04 13:09:41
23	NisperosPaco	2	2016-08-04 13:28:02
24	Ford	0	2016-08-24 09:55:30

Showing 1 to 6 of 6 entries Previous **1** Next

Figura 3.13: Listado de clientes registrados

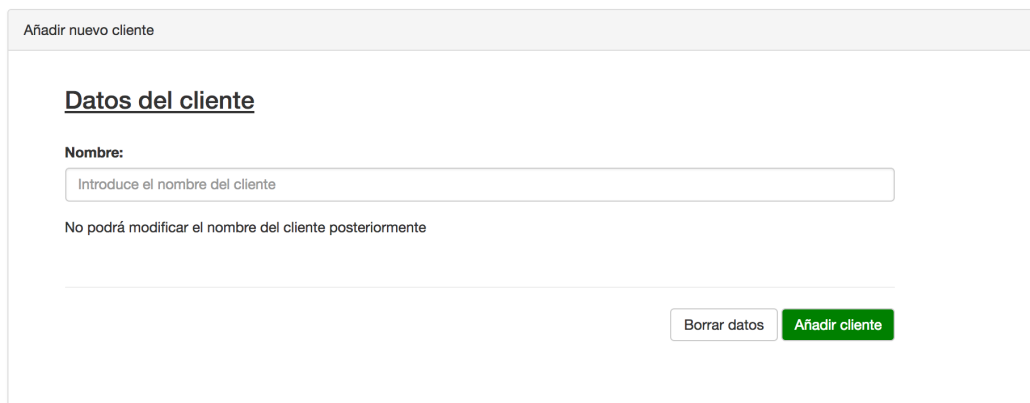
ción encontraremos un formulario como el presente en la Figura 3.18, mediante el cual podremos indicar la información referente al nuevo producto, entre la que destacamos el cliente al que irá asociado dicha entrada, su identificador, descripción, nombre, medidas, y de forma opcional, una imagen o plano del producto en cuestión. De esta forma, al indicar todos los datos mencionados anteriormente y pulsar el botón *Añadir entrada*, se nos mostrará, tal como podemos observar en la Figura 3.19, un mensaje de confirmación indicando si se ha dado de alta de forma satisfactoria, o no, la entrada indicada.

Destacamos además que la subida del plano o imagen la realizaremos mediante una conexión FTP; esto requerirá por nuestra parte la instalación y configuración de un servicio de FTP para realizar dichas transferencias a través del portal. De esta forma, el conjunto de instalaciones para disponer de los servicios necesarios los expondremos en la Sección 3.2.2.

Otra de las posibilidades que nos ofrece este portal de administración será la de modificar las entradas de productos ya registrados en el sistema; para ello nos dirigiremos a la sección *Modificar entrada* del menú lateral, donde se nos mostrará un formulario como el mostrado en la Figura 3.20, en el cual deberemos indicar el cliente al que pertenece la entrada que deseamos modificar. Una vez seleccionado el cliente, tal como observamos en la Figura 3.21, deberemos especificar la entrada del producto que queremos modificar. Finalmente, como podemos ver en la Figura 3.22, se nos mostrará toda la información asociada a dicha entrada, la cual podremos modificar y salvar de forma posterior.

Para finalizar con la sección de gestión de entradas de productos, cabe la pena destacar la posibilidad por parte de un administrador del portal, de eliminar cualquiera de las entradas de productos presentes en el servidor. Para realizar dicha

Alta Nuevo Cliente



Añadir nuevo cliente

Datos del cliente

Nombre:

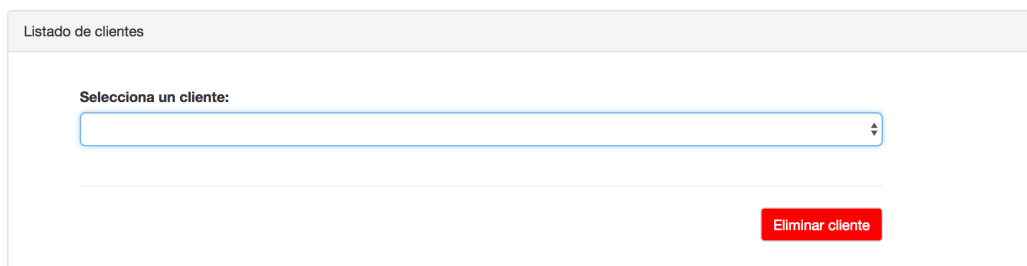
Introduce el nombre del cliente

No podrá modificar el nombre del cliente posteriormente

Borrar datos Añadir cliente

Figura 3.14: Dar de alta un nuevo cliente en el sistema

Eliminar Cliente



Listado de clientes

Selecciona un cliente:

Eliminar cliente

Figura 3.15: Eliminar un cliente

acción, nos dirigiremos a la sección **Eliminar** entrada del menú lateral, de forma que se nos conducirá a una página como la que se muestra en la Figura 3.23, en la que deberemos seleccionar un cliente presente en el sistema, para que de forma posterior, tal como observamos en la Figura 3.24, indiquemos la entrada asociada a dicho cliente que queremos eliminar.

Otra de las características del sistema de información será la gestión de copias de seguridad, de tal forma que podremos, como muestra la Figura 3.25, listar el conjunto de copias de seguridad realizadas y almacenadas en el servidor, crear una nueva copia de seguridad, o eliminar una existente.

Con respecto a la funcionalidad asociada al listado de copias de seguridad realizadas y presentes en el servidor debemos pulsar sobre la opción **Listar copias de seguridad** del menú lateral, de tal forma que se nos presentará un listado co-

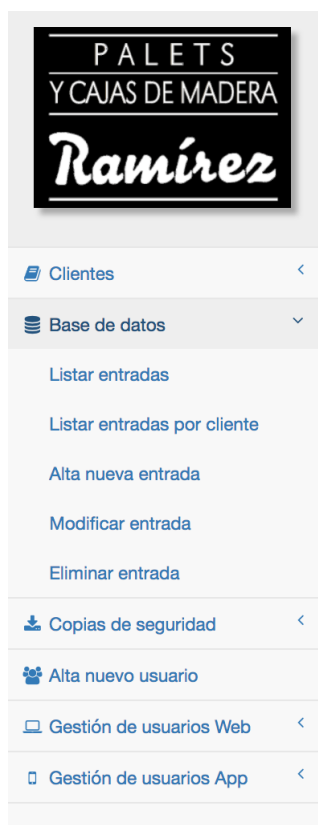


Figura 3.16: Acciones asociadas a la gestión de la base de datos

mo el mostrado en la Figura 3.26, donde se nos mostrará el conjunto de copias de seguridad realizadas, las cuales podremos descargar pulsando sobre su nombre.

Además, disponemos de la capacidad de realizar una copia de seguridad desde este mismo menú, pulsando sobre la opción *Crear copia de seguridad* del menú lateral, donde se nos mostrará, tal como podemos observar en la Figura 3.27, un resumen del estado actual de la base de datos, de la cual realizaremos una copia de seguridad completa al pulsar sobre el botón *Crear copia de seguridad*.

Por otro lado, para acabar con el conjunto de la funcionalidad asociada a la gestión de copias de seguridad, destacamos la posibilidad de eliminar cualquiera de las copias realizadas y almacenadas en el servidor. Para ello, nos dirigiremos a la sección *Eliminar copia de seguridad* del menú lateral, donde se nos mostrará un formulario como el mostrado en la Figura 3.28, en el que podremos indicar la copia de seguridad que deseamos eliminar del servidor.

Siguiendo con las opciones disponibles desde el portal de administración de nuestro servidor, destacamos la posibilidad de crear nuevos usuarios, bien sea para garantizar su acceso autenticado a la aplicación móvil, o al propio portal de administración web. De esta forma, pulsando sobre el botón *Alta nuevo usuario*, visualizaremos un formulario como el mostrado en la Figura 3.29, en el que po-

Listado de entradas en la base de datos

Entradas en la base de datos

Show entries Search:

Cliente ▲	ID ↕	Tipo ↕	Nombre ↕	Medidas ↕	Descripción ↕	Imagen ↕	Fecha de registro ↕
BombasIdeal	3	Típoooo	Nombreeee	12*12	Descripcioooooo oooooon	-	2016-08-08 11:54:07
BombasIdeal	B1	w	q	w	w	-	2016-08-04 14:14:29
Iberica	600	Jaula	Jaula-600	220x135x125	ddgdddf uhuuhm ihuhuiu yy	Jaula-600	2016-08-24 10:07:24
Mulem	M1	CajaMulem	CajaMulem1	2x2	Cajaaaaa	CajaMulem1	2016-08-09 12:04:22
Mulem	M2	CajaMulem	CajaMulem2	34x45x56	Descripcion caja	CajaMulem2	2016-08-17 16:05:03
NisperosPaco	N1	Caja Nisperos	Caja Nisperos 1	12*12	jnsdjanas aks xkas cj kasda á	Caja Nisperos 1	2016-08-04 13:30:27
NisperosPaco	N2	Caja Nisperos	Caja Nisperos 2	34x45x56	aaaaaaaaaaaaa bbbbbbbbbbb =>	Caja Nisperos 2	2016-08-05 14:00:10

Showing 1 to 7 of 7 entries Previous **1** Next

Figura 3.17: Listado de entradas de productos

dremos indicar todos los datos asociados a este nuevo usuario, tales como su nombre, apellidos, cargo dentro de la organización, nombre de usuario, contraseña, si se trata de un usuario para la aplicación móvil, para el portal de administración web, o para los dos casos; e incluso podremos adjuntar una imagen a modo de avatar para la personalización del perfil del usuario creado.

Para finalizar con la descripción de la funcionalidad del portal web de administración de nuestro servidor, destacamos la posibilidad de realizar una gestión completa de los usuarios que hayamos dado de alta, tanto de los usuarios de la aplicación móvil como los usuarios del portal web. En ambos casos, tal como observamos en la Figura 3.30, las opciones disponibles son las mismas, de las cuales diferenciamos el listado de usuarios de cada tipo, así como la modificación de cada uno de ellos, o su supresión del sistema, lo que conllevaría la pérdida y denegación del acceso a la plataforma en la que estuvieran dados de alta. Dichas opciones funcionales las podemos localizar en el menú lateral bajo las secciones Gestión de usuarios Web, y Gestión de usuarios App.

Tanto para obtener un listado de los usuarios dados de alta en la aplicación móvil como en el portal de administración web, podremos dirigirnos a la sección Listado de usuarios de ambas secciones indicadas anteriormente, desde las cuales se nos presentará una tabla como la mostrada en la Figura 3.31, en la que podremos observar el conjunto de los usuarios dados de alta para dicha sección, así como sus datos personales, avatar, y credenciales de acceso.

Otra de las posibilidades presentes en la funcionalidad asociada a la gestión de usuarios es la de modificar los datos de un usuario dado de alta en el sistema,

Nueva entrada en la base de datos

Añadir nueva entrada a la base de datos

Datos de la entrada

Selecciona un cliente:

Identificador (indicar solo si existe la referencia):

Tipo:

Nombre:

Medidas:

Descripción:

Plano/Imagen:

Seleccionar archivo Ningún archivo seleccionado

Figura 3.18: Alta de nueva entrada de producto

para ello nos dirigiremos a la opción *Modificar usuario* de su apartado correspondiente, donde tal como muestran las Figuras 3.32 y 3.33, podremos indicar el usuario deseado para su posterior edición y salvado de datos personales de usuario.

Para acabar con la funcionalidad asociada a la gestión de usuarios destacamos la posibilidad de eliminar usuarios, tanto de la aplicación móvil como del portal web de administración; recordar además que dicha acción conllevaría la pérdida del acceso a dichas aplicaciones. Para acceder a la sección desde la que podemos llevar a cabo la eliminación de usuarios del sistema habremos de pulsar sobre el botón *Eliminar usuario* de su apartado correspondiente, de tal forma que se nos mostrará un formulario como el presente en la Figura 3.34, en el que podremos indicar el usuario a ser eliminado del sistema.

Finalmente, vale la pena destacar el uso de tres ficheros PHP adicionales, los cuales son `conexion.php` (presente en el Anexo A.2), `funciones.php` (presente en el Anexo A.5), y `logout.php` (presente en el Anexo A.3). Dichos códigos se incluyen y asocian tanto en la página de acceso como en la página de inicio del portal, con el fin de gestionar el acceso a las bases de datos en el momento en que iniciamos una sesión en el portal, de aportar el conjunto de funciones desarrolladas

Nueva entrada en la base de datos

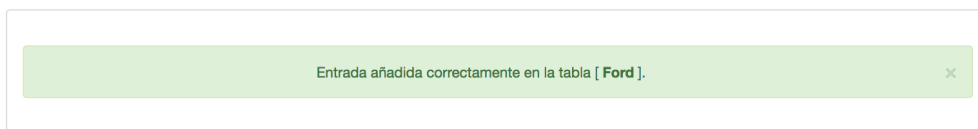


Figura 3.19: Confirmación de alta de nueva entrada

Modificar entrada de la base de datos

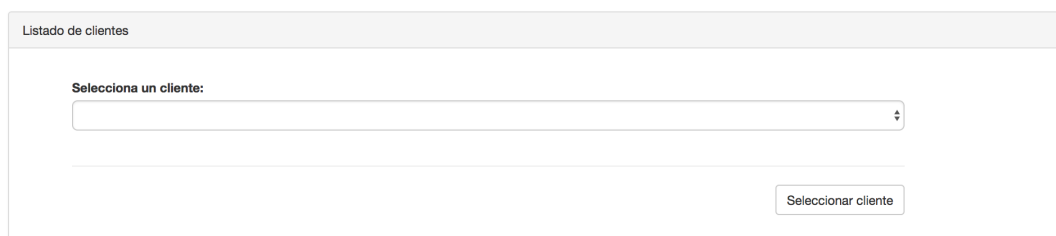
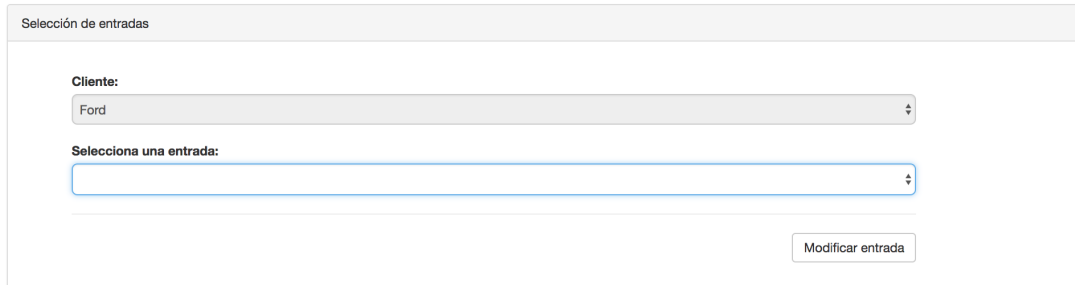


Figura 3.20: Modificación de entrada en la base de datos (selección de cliente)

para llevar a cabo todas las tareas descritas previamente, y para poder realizar un cierre de sesión seguro en el momento en que el usuario decide cerrar la sesión.

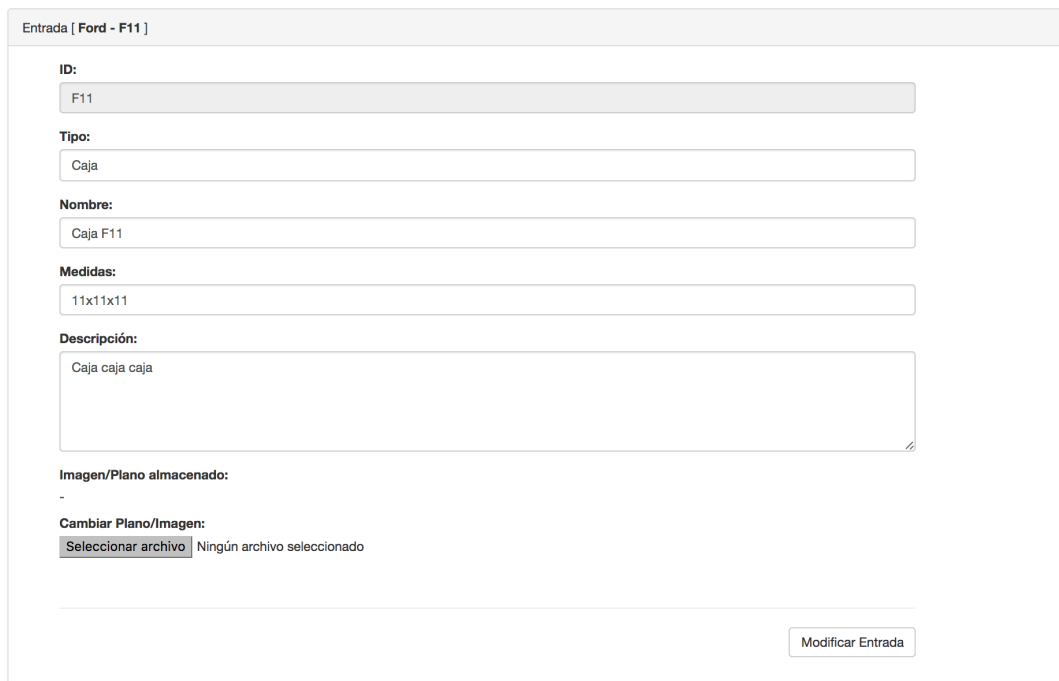
Modificar entrada de la base de datos



The screenshot shows a web form titled "Selección de entradas". It contains two dropdown menus. The first is labeled "Cliente:" and has "Ford" selected. The second is labeled "Selecciona una entrada:" and is currently empty. A "Modificar entrada" button is located at the bottom right of the form.

Figura 3.21: Modificación de entrada en la base de datos (selección de entrada)

Modificar entrada de la base de datos



The screenshot shows a web form titled "Entrada [Ford - F11]". It contains several input fields: "ID:" with "F11", "Tipo:" with "Caja", "Nombre:" with "Caja F11", and "Medidas:" with "11x11x11". The "Descripción:" field contains "Caja caja caja". Below these fields, there is a section for "Imagen/Plano almacenado:" with a "-" sign, and a "Cambiar Plano/Imagen:" section with a "Seleccionar archivo" button and the text "Ningún archivo seleccionado". A "Modificar Entrada" button is located at the bottom right of the form.

Figura 3.22: Modificación de entrada en la base de datos (edición de entrada)

Eliminar entrada de la base de datos



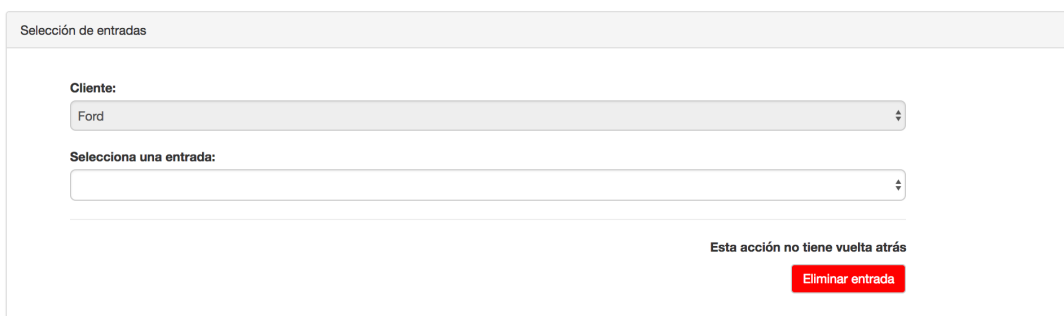
Listado de clientes

Selecciona un cliente:

Seleccionar cliente

Figura 3.23: Supresión de entrada de la base de datos (selección de cliente)

Eliminar entrada de la base de datos



Selección de entradas

Cliente:

Ford

Selecciona una entrada:

Esta acción no tiene vuelta atrás

Eliminar entrada

Figura 3.24: Supresión de entrada de la base de datos (selección de entrada)

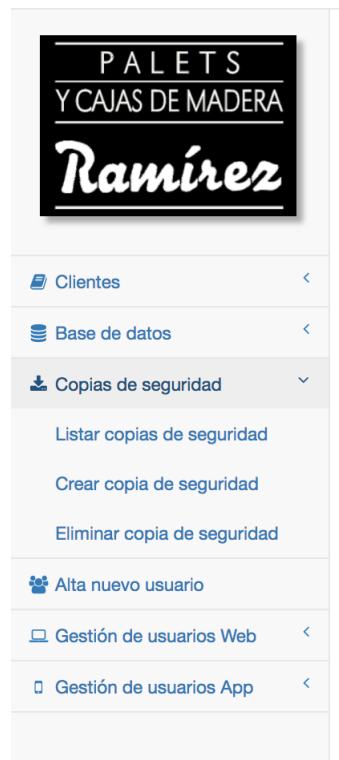


Figura 3.25: Acciones asociadas a la gestión de copias de seguridad

Listado de copias de seguridad

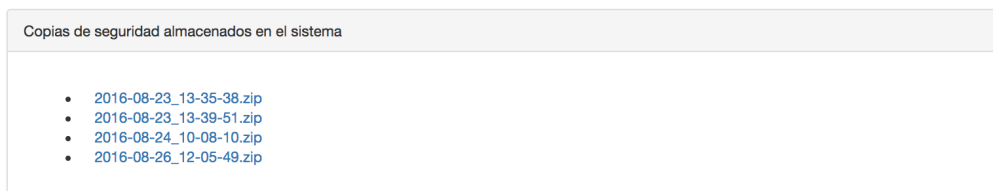


Figura 3.26: Listar copias de seguridad

Crear copia de seguridad

Estado de la base de datos

Número de tablas en la base de datos:
9

Tablas almacenadas en la base de datos:
BombasIdeal
Ford
Iberica
Mulem

Número de entradas totales en la base de datos:
19

Crear copia de seguridad

Figura 3.27: Crear nueva copia de seguridad

Eliminar copia de seguridad

BackUPS del sistema

Selecciona una copia de seguridad:

Eliminar copia de seguridad

Figura 3.28: Supresión de copia de seguridad

Alta nuevo usuario

Añadir nuevo usuario

Datos del usuario

Seleccione el tipo de usuario:

Usuario Web
 Usuario App

Nombre:

Apellidos:
Cargo:
Nombre de usuario:
Contraseña:
Avatar de usuario:

Ningún archivo seleccionado

Figura 3.29: Alta de nuevo usuario

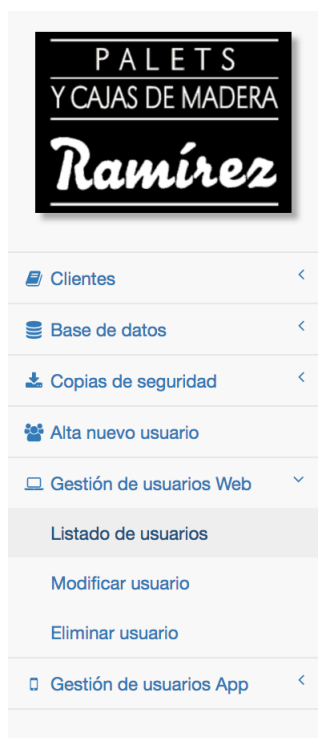


Figura 3.30: Acciones asociadas a la gestión de usuarios

Listado de usuarios Web

Usuarios con acceso al portal Web

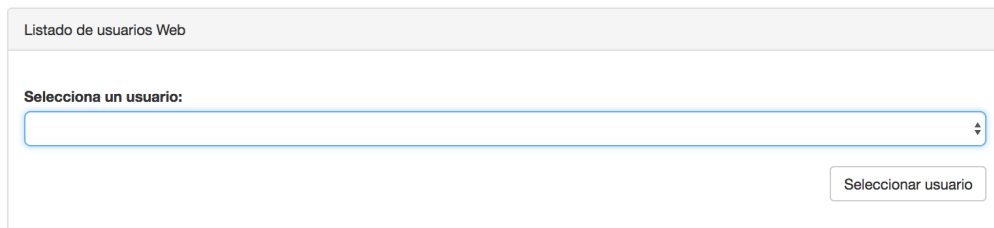
Show entries Search:

Nombre	Apellidos	Cargo	Avatar	Usuario	Contraseña
Alvaro	Roig	Webmaster	arco	arco	*****
Gabi	Roig Luján	Jefe de Producción y Personal	-	gabiroig	*****

Showing 1 to 2 of 2 entries Previous **1** Next

Figura 3.31: Listado de usuarios

Modificar usuario Web



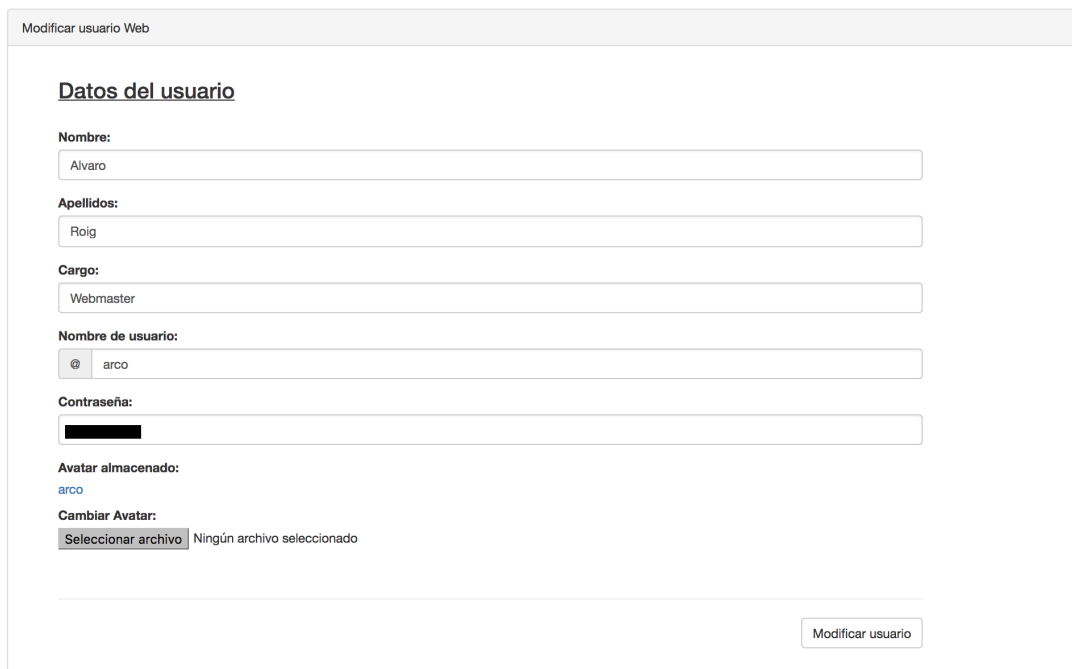
Listado de usuarios Web

Selecciona un usuario:

Seleccionar usuario

Figura 3.32: Modificación de usuarios (selección de usuario)

Modificar usuario Web



Modificar usuario Web

Datos del usuario

Nombre:

Apellidos:

Cargo:

Nombre de usuario:

Contraseña:

Avatar almacenado:
[arco](#)

Cambiar Avatar:
 Ningún archivo seleccionado

Figura 3.33: Modificación de usuarios (edición de datos)

Eliminar usuario Web

Listado de usuarios Web

Selecciona un usuario:

Figura 3.34: Eliminar un usuario

3.2.2. Instalación de servidores en la RaspberryPi

Con tal de ofrecer a los administradores del Sistema de Información un mecanismo de gestión fiable y sencillo desde el cual realizar el conjunto de operaciones necesarias para garantizar su correcto funcionamiento, hemos desarrollado un portal de administración desde el que realizar dichos trámites a través de Internet. De esta forma, debemos garantizar el acceso a dicho portal por parte de los usuarios. Consecuentemente, en esta sección nos disponemos a instalar todos los servidores necesarios para mantener operativo y accesible nuestro portal administrativo.

Debido a las necesidades del portal de gestión, precisamos hacer uso de un servicio web, un servicio FTP, y un servicio de base de datos que ofrezca una interacción sencilla con el resto de servicios. Dadas las necesidades anteriormente nombradas, vamos a hacer uso de un servidor Apache, un servidor FTP VSFTPD, y una base de datos MySQL.

Instalación del servidor Web

Como hemos comentado anteriormente, el servidor que vamos a instalar en la Raspberry Pi es un servidor web Apache. El primer paso para la instalación del mismo es crear (en caso de que no exista) y dar permisos al grupo de trabajo que usa Apache por defecto. Realizamos dichas acciones ejecutando las órdenes mostradas en la Tabla 3.1.

```
$ sudo addgroup www-data
$ sudo usermod -a -G www-data www-data
```

Tabla 3.1: Órdenes para crear y dar permisos al grupo de trabajo que utiliza Apache

Una vez hecho esto, nos disponemos a actualizar los repositorios, y a instalar Apache, así como la extensión de PHP que utilizará el servidor web para gestionar los recursos del servidor. Podemos encontrar las órdenes de consola necesarias para este propósito en la Tabla 3.2.

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install apache2 php5 libapache2-mod-php5
$ sudo apt-get install php5
```

Tabla 3.2: Órdenes para actualizar los repositorios e instalar *Apache*

Finalmente, como medida de seguridad para este tipo de instalaciones por defecto de Apache, procederemos a editar el fichero `/etc/apache2/apache2.conf`, en el cual buscaremos la referencia `<Directory /var/www/>` y cambiaremos la línea `Options Indexes FollowSymLinks` por `Options FollowSymLinks`, con el fin de prohibir el listado de elementos situados dentro de los directorios de nuestra ruta de publicación de contenidos en el servidor.

Tras haber instalado los servicios anteriores, reiniciamos el servidor web mediante la orden `sudo /etc/init.d/apache2 restart`.

Instalación del servidor FTP

El portal de gestión telemático nos ofrece, como uno de los principales elementos funcionales, la posibilidad de almacenar planos o imágenes de los productos registrados en el sistema. Para realizar las transacciones de dichos ficheros desde el equipo local del administrador hacia el servidor haremos uso de un servidor FTP, el cual nos permitirá de manera sencilla, copiar los archivos mencionados a los directorios del servidor destinados a contener dichos ficheros.

Consecuentemente, hemos elegido hacer uso del servidor FTP VSFTPD. Para su instalación ejecutaremos la orden de consola `sudo apt-get install vsftpd`, y tras la misma, deberemos modificar el fichero `/etc/vsftpd.conf` descomentando las líneas `local-enable=YES` y `write-enable=YES`; lo que permitirá la escritura de archivos en la Raspberry Pi por parte de los usuarios que administren el servidor. Además, editaremos los permisos de la carpeta de almacenamiento de los ficheros del portal web mediante la orden de consola `sudo chown -R pi /var/www` con el fin de permitir la subida mediante el servicio FTP de los archivos iniciales del portal por parte de nuestro usuario administrador del servidor.

Finalmente, para la correcta puesta en marcha del servidor FTP, ejecutaremos la orden `sudo service vsftpd restart`.

Instalación de la base de datos

Una vez hemos instalado el servidor web, nos disponemos a instalar una base de datos que nos garantice el acceso autenticado al portal y el almacenamiento de toda la información del sistema. La base de datos que vamos a instalar es MySQL, y además, instalaremos también el software de gestión de bases de datos PHPMyAdmin, desde el cual poder gestionar todo lo referente a las bases de datos MySQL. Para ello, ejecutaremos las órdenes mostradas en la Tabla 3.3.

```
$ sudo ifup lo
$ sudo apt-get install mysql-server mysql-client php5-mysql phpmyadmin
```

Tabla 3.3: Órdenes para instalar MySQL y PHPMyAdmin

Tras haber instalado los servidores anteriores, debemos modificar el fichero `php.ini` con el fin de indicar al servidor web que haremos uso de los servicios de MySQL. Dicha acción la realizamos insertando la línea `extension=mysql.so` después de la cabecera marcada como [Dynamic Extensions] del fichero situado en la ruta `/etc/php5/apache2/php.ini`.

Además, deberemos indicar en el fichero de configuración de Apache que hemos habilitado el gestor de bases de datos PHPMyAdmin. Dicha acción la realizamos añadiendo la línea `Include /etc/phpmyadmin/apache.conf` al final del fichero situado en la ruta `/etc/apache2/apache2.conf`, y reiniciando el servidor mediante la orden `sudo /etc/init.d/apache2 restart`.

Una vez hecho esto, ejecutamos las órdenes de consola situadas en la Tabla 3.4 para crear los ficheros de configuración necesarios para la asociación de todos los servidores que hemos instalado.

```
$ sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf.d/phpmyadmin.conf
$ sudo /etc/init.d/apache2 reload
```

Tabla 3.4: Órdenes para crear ficheros de configuración

Por último, situamos todos los códigos pertenecientes al portal web de gestión en la ruta `/var/www/html`, donde se sitúan los archivos públicos del servidor web; y mediante el uso de la herramienta de gestión de bases de datos PHPMyAdmin, creamos una base de datos, le agregamos tres tablas semejantes a las de las Tablas 3.5, 3.6, y 3.7, para guardar las credenciales de acceso de los administradores del portal, las credenciales de los usuarios de la aplicación móvil, y los datos de registro de clientes, y finalmente agregamos una entrada en la tabla de usuarios web con nuestras credenciales para poder acceder al portal con permisos de administración. Una vez creadas estas tablas y habiendo introducido de forma manual nuestro usuario de administrador en la base de datos, el propio programa será capaz de autogenerar los recursos y tablas necesarias para el correcto funcionamiento de toda su funcionalidad.

```
CREATE TABLE usuarios_web(
  id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  user VARCHAR(30) NOT NULL,
  pass VARCHAR(30) NOT NULL,
  nombre VARCHAR(30) NOT NULL,
  apellidos VARCHAR(50) NOT NULL,
  cargo VARCHAR(100),
  imagen VARCHAR(100),
  reg_date TIMESTAMP
);
```

Tabla 3.5: Tabla de MySQL para guardar credenciales web


```
CREATE TABLE usuarios_app(  
  id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  user VARCHAR(30) NOT NULL,  
  pass VARCHAR(30) NOT NULL,  
  nombre VARCHAR(30) NOT NULL,  
  apellidos VARCHAR(50) NOT NULL,  
  cargo VARCHAR(100),  
  imagen VARCHAR(100),  
  reg_date TIMESTAMP  
);
```

Tabla 3.6: Tabla de MySql para guardar credenciales App

```
CREATE TABLE clientes(  
  id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(50) NOT NULL,  
  reg_date TIMESTAMP  
);
```

Tabla 3.7: Tabla de MySql para guardar el registro de clientes

3.2.3. Servicios de reinicio y apagado del servidor

Debido al hecho de que este servidor ha sido planteado y diseñado como un sistema empotrado de tipo *headless*, es decir, será un dispositivo conectado a la red eléctrica y a la red local de la empresa, pero no dispondrá de un monitor desde el cual visualizar su actividad, ni de dispositivos de interacción físicos como pudieran ser un teclado o un ratón, nos encontramos con la necesidad de dar la posibilidad a los usuarios administradores de llevar a cabo tareas interactivas con el sistema operativo del servidor tales como las acciones de reinicio o apagado seguro del mismo.

Es por ello que necesitaremos habilitar la llamada a servicios del sistema desde una página web, de tal forma que procederemos a descargar e instalar la librería CGI⁶ de PHP5 mediante las órdenes de consola mostradas en la Tabla 3.8; de esta forma dispondremos de la capacidad de realizar llamadas al sistema operativo desde nuestro portal de administración.

```
$ sudo apt-get install php5-cgi
$ sudo systemctl daemon-reload
$ sudo service apache2 restart
```

Tabla 3.8: Instalación e inicio del CGI

Para dotar a los administradores del portal de esta funcionalidad, agregaremos al mismo la interfaz que nos permitirá llevar a cabo estas acciones, las cuales serán accesibles tal como muestra la Figura 3.35 desde el menú superior derecho del portal. Estos botones llamarán a las funciones presentes en el Anexo A.4, las cuales, dependiendo del tipo de acción indicada, harán uso de la librería CGI de PHP5 con el fin de realizar las llamadas al sistema que llevarán a cabo un reinicio o apagado seguro del servidor.

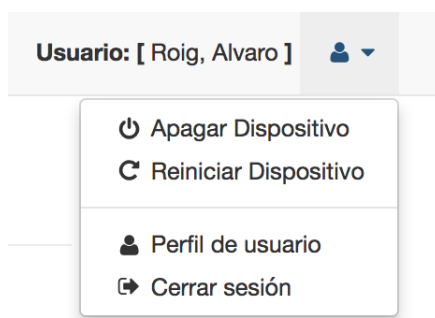


Figura 3.35: Interfaz para reinicio y apagado del servidor

⁶La interfaz de entrada común (en inglés *Common Gateway Interface*, abreviado CGI) es una importante tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web.

3.3 Consideraciones de seguridad

Hoy en día uno de los puntos más importantes a tener en cuenta a la hora de desplegar servicios digitales es el nivel de seguridad y privacidad del que disponen dichos servicios. En esta sección nos disponemos a abordar este punto tan fundamental, comentando los mecanismos de seguridad de los que dispone la empresa y los que vamos a instaurar en el Sistema de Información para asegurar la seguridad y privacidad del mismo.

3.3.1. Mecanismos de seguridad en la empresa

Actualmente, en el momento en que comenzamos a desarrollar este proyecto, hemos comprobado que la infraestructura de red de la empresa con la que vamos a colaborar dispone de una conexión a Internet de tipo ADSL, y el elemento principal de la misma es un *router* proporcionado por la propia compañía que les suministra dicho servicio.

Ya que pretendemos en este proyecto que diferentes aplicaciones se comuniquen con algunos elementos de red vía inalámbrica, la primera consideración de seguridad será proceder a verificar la correcta configuración del *router* de la empresa con el fin de evitar posibles intrusiones indeseadas en la red interna de la misma mediante esta vía.

De esta forma hemos comprobado que el canal Wi-Fi de dicho *router* está configurado mediante una interfaz 802.11 b/g/n, la cual trabaja en la banda de frecuencia de 2.4 GHz, y es capaz de alcanzar hasta 300 Mbit/s. Además hemos llevado a cabo un escaneo de los canales Wi-Fi presentes en las inmediaciones de la empresa con la finalidad de ver qué canales eran los menos utilizados, y así configurar el canal Wi-Fi del *router* de la empresa con el canal con menos uso detectado, de tal forma que evitaremos posibles solapes o interferencias en dicha banda de frecuencia.

Con respecto a la configuración de seguridad en el *router* con el que estamos trabajando hemos deshabilitado la funcionalidad WPS⁷ del mismo, ya que dicha funcionalidad presenta una importante vulnerabilidad de seguridad que ha sido ampliamente explotada en todo el mundo.

Otra de las características que hemos revisado con respecto a la configuración en este dispositivo ha sido la verificación de que los datos de acceso a su propio portal de configuración no eran los establecidos por defecto. Además hemos comprobado que la configuración de seguridad de la red Wi-Fi presenta un algoritmo WPA2-PSK con encriptación AES, lo cual consideramos suficientemente seguro en el caso con el que estamos trabajando.

⁷WPS (*Wi-Fi Protected Setup*) es un estándar de 2007, promovido por la Wi-Fi Alliance para facilitar la creación de redes WLAN. Más información en <http://www.wi-fi.org/>.

3.3.2. Mecanismos de seguridad implantados

Con respecto a las consideraciones de seguridad llevadas a cabo a la hora de realizar la configuración base de nuestro servidor, podemos destacar la elección de establecer el canal de comunicaciones con el *router* mediante Ethernet (a pesar de la posibilidad de hacer uso de las tarjetas de comunicación inalámbricas integradas en los nuevos modelos de Raspberry Pi) debido a la mayor seguridad que nos ofrece este medio, ya que en caso de que estableciéramos comunicaciones basadas en una red inalámbrica WiFi asumiríamos riesgos de seguridad que se consideran innecesarios en este proyecto.

Además, al establecer la comunicación de red vía Ethernet, podemos llevar a cabo configuraciones de red extra en caso de que fueran necesarias en un futuro, tales como la inclusión del servidor en una red virtual (VLAN⁸), o incluso llevar a cabo filtrados de acceso a recursos mediante un proxy. Por otro lado destacamos la posibilidad de hacer uso de certificados de seguridad para cifrar todo tipo de comunicaciones entre diferentes elementos de la red, ya sean clientes o servidores, pero esta será una propuesta de ampliación del proyecto debido a que en este punto de desarrollo del mismo no se trata de una característica fundamental.

Finalmente destacamos que con el fin de asegurar el correcto tratamiento de la información por parte del Sistema de Información, así como de su almacenamiento en la base de datos, al hacer uso del portal de gestión web comprobamos a nivel de capa de aplicación todos los datos que son introducidos por parte de un usuario, analizándolos y procediendo a su modificación en caso de que contengan sentencias o conjuntos de caracteres que aparenten conllevar el posible mal funcionamiento del propio Sistema de Información o de la base de datos.

⁸Una VLAN, acrónimo de *virtual LAN* (red de área local virtual), es un método para crear redes lógicas independientes dentro de una misma red física.

CAPÍTULO 4

Diseño de la aplicación móvil de usuario

En este capítulo abordaremos el diseño y desarrollo de una aplicación móvil multiplataforma para la consulta ágil de información almacenada en nuestro servidor. Esta aplicación estará orientada a facilitar la tarea de consulta de información de productos que deben realizar los operarios de la empresa al preparar un nuevo pedido para un cliente, y el objetivo principal del desarrollo que vamos a llevar a cabo será ofrecer la posibilidad de desplegar la aplicación en cualquier terminal móvil que funcione con los sistemas operativos Android o iOS.

4.1 Introducción a Xamarin

Con el fin de diseñar una aplicación móvil multiplataforma que nos permita llevar a cabo una instalación y despliegue de la misma sin tener en cuenta el sistema operativo sobre el cual está funcionando el terminal móvil, vamos a hacer uso de Xamarin.

Xamarin, cuyo logo podemos observar en la Figura 4.1, es una plataforma de desarrollo que nos permite diseñar aplicaciones nativas multiplataforma para una gran variedad de sistemas operativos, como Android, iOS, y Windows Phone, en lenguajes de programación como C# o C++.



Figura 4.1: Logo de Xamarin

Descendiente del proyecto de código libre Mono Project¹ que llevó .NET a Linux, la plataforma de Xamarin es un puerto para .NET hacia los sistemas operativos Android e iOS con soporte para Windows Phone. Teniendo en cuenta que Xamarin.Android es Mono para Android, y Xamarin.iOS es Mono para iOS, se generan, tal como podemos observar en la Figura 4.2, enlaces entre C# y las API nativas de Android e iOS para el desarrollo en móviles y tablets que funcionen con estos sistemas operativos. Esto nos ofrece la posibilidad de contar con la potente capacidad de diseñar interfaces nativas, gráficos, animaciones y características propias de Android o iOS, pero desarrollando el cuerpo principal de nuestros programas en el lenguaje de programación C#, pudiendo compartir este código entre las diferentes implementaciones del proyecto para cada sistema operativo.



Figura 4.2: Arquitectura de librerías en Xamarin

Con cada nueva actualización de los sistemas operativos Android o iOS se lanza una nueva versión de Xamarin con el fin de enlazar a las nuevas API para dichos sistemas, por lo que nos aseguramos poder disponer siempre de las últimas características de cada nueva versión de estos sistemas operativos. Además, el puerto de conexión con .NET que nos ofrece Xamarin nos permite trabajar con características propias de este entorno tales como tipos de datos, generalidades, métodos asíncronos, delegaciones, etc.

Todas estas librerías son organizadas por un enlazador interno de Xamarin que nos permite elegir en cada proyecto para qué sistemas queremos desarrollarlo, pudiendo elegir entre varios de ellos o uno solo, de tal forma que aparece el concepto de Xamarin.Forms, el cual vamos a utilizar en este trabajo, y que se trata de una capa de aplicación superior a Xamarin.Android y Xamarin.iOS, que nos permite desarrollar aplicaciones multiplataforma que compartan códigos desarrollados en C# con características propias de .NET, pero haciendo uso de interfaces nativas de Xamarin.Android o Xamarin.iOS según sea el caso de despliegue

¹Patrocinado por Microsoft, Mono es una implementación de código abierto del *framework* .NET basado en los estándares ECMA para C# y el *Common Language Runtime*. Más información en <http://www.mono-project.com/>.

en un sistema operativo o en otro. De esta forma disponemos de un entorno de desarrollo en .NET con enlaces a API nativas de Android e iOS que nos permite el diseño de aplicaciones multiplataforma para diferentes sistemas operativos.

4.2 Diseño de la aplicación

Una vez hemos llevado a cabo una breve introducción a Xamarin nos disponemos a comenzar con el diseño de la aplicación móvil, tarea que vamos a llevar a cabo, tal como mostramos en la Figura 4.3, mediante el propio IDE² de Xamarin, el cual es conocido como Xamarin Studio. Cabe la pena destacar que dicho IDE está disponible únicamente para los sistemas operativos MacOS, aunque disponemos de la posibilidad de integrarlo mediante un *plugin* en el IDE Visual Studio de los sistemas Windows. Además, haremos uso de emuladores software de los sistemas operativos Android e iOS para simular el comportamiento de nuestra aplicación en dichos sistemas.

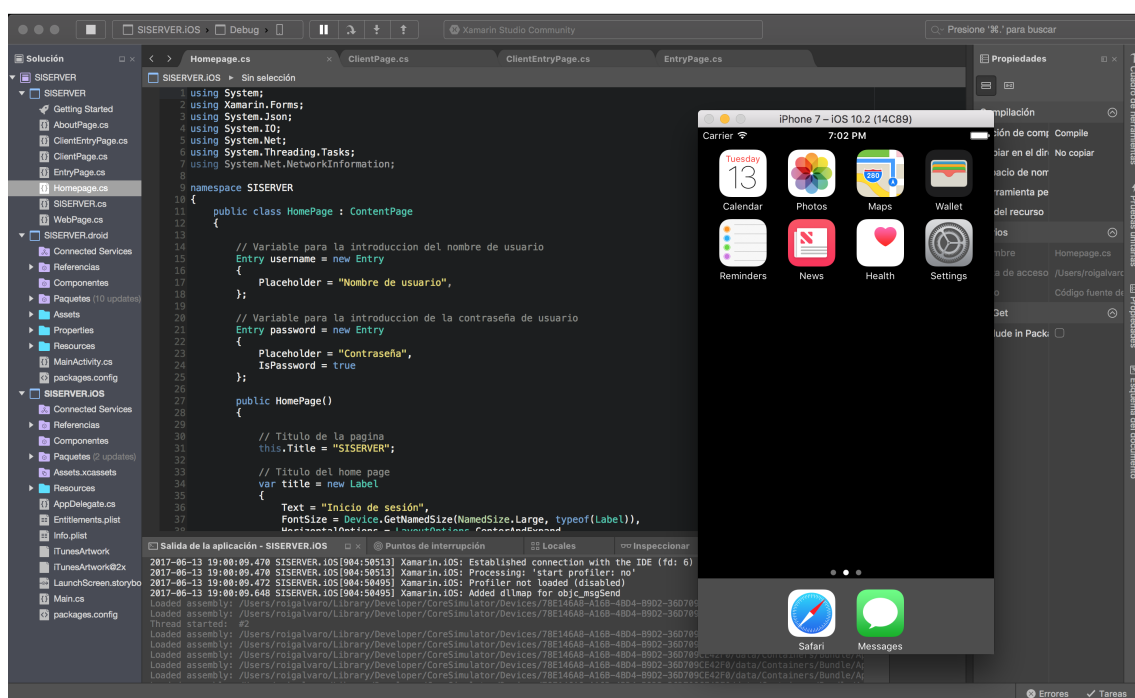


Figura 4.3: Entorno de desarrollo integrado de Xamarin

Tal como hemos comentado anteriormente, vamos a hacer uso de Xamarin.Forms para llevar a cabo un diseño multiplataforma de la aplicación para los Sistemas Operativos Android e iOS, en la que cada implementación para cada uno de ellos dispondrá de una interfaz gráfica nativa de forma autogenerada por Xamarin

²Un entorno de desarrollo integrado (en inglés *Integrated Development Environment*), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software

Studio, mientras que el código principal de la aplicación se escribirá en lenguaje C# y será compartido para las dos versiones de la aplicación.

Con respecto al planteamiento del patrón de diseño de la aplicación establecemos los siguientes apartados de los que dispondremos en la misma:

- Página de inicio de sesión para acceder al contenido.
- Página de información general de la aplicación.
- Página de listado de los clientes presentes en la base de datos.
- Página de listado de productos asociados a cada cliente de la base de datos.
- Página de enumeración de las características de un producto.

Una vez definidas las secciones que formarán parte de nuestra aplicación nos disponemos a comentar el funcionamiento de cada una de ellas, así como a mostrar las diferentes interfaces para cada una de las versiones generadas para los dos sistemas operativos en los que estamos trabajando. Por otra parte, vale la pena destacar que todas las comunicaciones realizadas entre la aplicación móvil y el servidor serán llevadas a cabo mediante el uso de una API RESTful que desarrollaremos en el Apartado 4.3, la cual permanecerá en ejecución en el servidor a la espera de recibir peticiones por parte de las aplicaciones.

Página de inicio de sesión para acceder al contenido

Esta será la página de inicio de nuestra aplicación, cuyo código podemos encontrar en el Anexo B.1, y en la cual, tal como podemos observar en las Figuras 4.4a y 4.4b, el usuario podrá introducir su nombre de usuario y contraseña para acceder a la información presente en el servidor, a la cual podrá acceder en el caso de que se trate de un usuario autorizado a acceder a dicha información. Además podremos pulsar sobre el botón *App Info* con el fin de acceder a la página de información de la aplicación que describimos en el siguiente punto.

Finalmente destacamos que estos datos de acceso que debe introducir el usuario habrán de ser establecidos de forma explícita por el administrador del Sistema de Información en la página web de gestión del mismo, ya que pueden coincidir, o no, con los datos de acceso para un mismo usuario en el propio portal web; es decir, disponemos de la posibilidad de permitir el acceso a un usuario únicamente a la aplicación móvil, únicamente a la página de gestión web, o bien a las dos aplicaciones.

Página de información general de la aplicación

Esta sección, cuyo código podemos encontrar en el Anexo B.2, será accesible desde la propia página de inicio de la aplicación, sin la necesidad de ha-

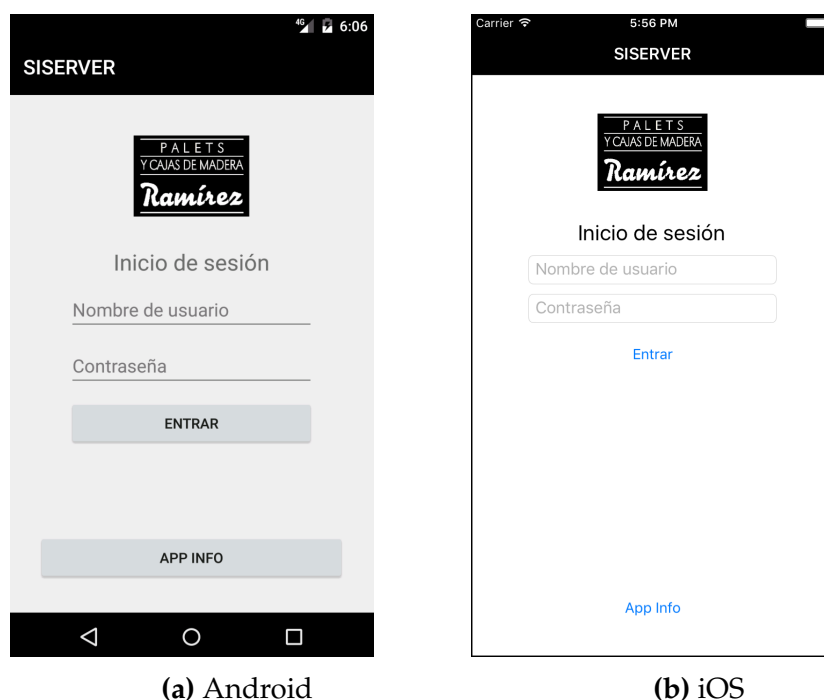


Figura 4.4: Páginas de inicio de la App

ber introducido unas credenciales de usuario, mediante la pulsación del botón App Info en la misma. En ella podremos visualizar, tal como muestran las Figuras 4.5a y 4.5b, un breve resumen acerca de la aplicación así como la información asociada a los desarrolladores de la misma.

Página de listado de los clientes presentes en la base de datos

Tras haber introducido los datos de acceso de usuario, en caso de una autenticación válida del mismo, seremos redirigidos a esta sección, cuyo código podemos encontrar en el Anexo B.3, y en la cual, tal como observamos en las Figuras 4.6a y 4.6b, se nos mostrará un listado del conjunto de clientes registrados en la base de datos del servidor, dándonos la posibilidad de elegir uno de ellos con el fin de proceder a la visualización de todas y cada una de las entradas de productos que tengan asociados dichos clientes.

Página de listado de productos asociados a clientes de la base de datos

Una vez hemos elegido un cliente, podremos visualizar la pantalla mostrada en las Figuras 4.7a y 4.7b, cuyo está disponible en el Anexo B.4, y en las cuales se nos muestra el conjunto de entradas de productos presentes en la base de datos del servidor para el cliente indicado por parte del usuario. En esta página podre-

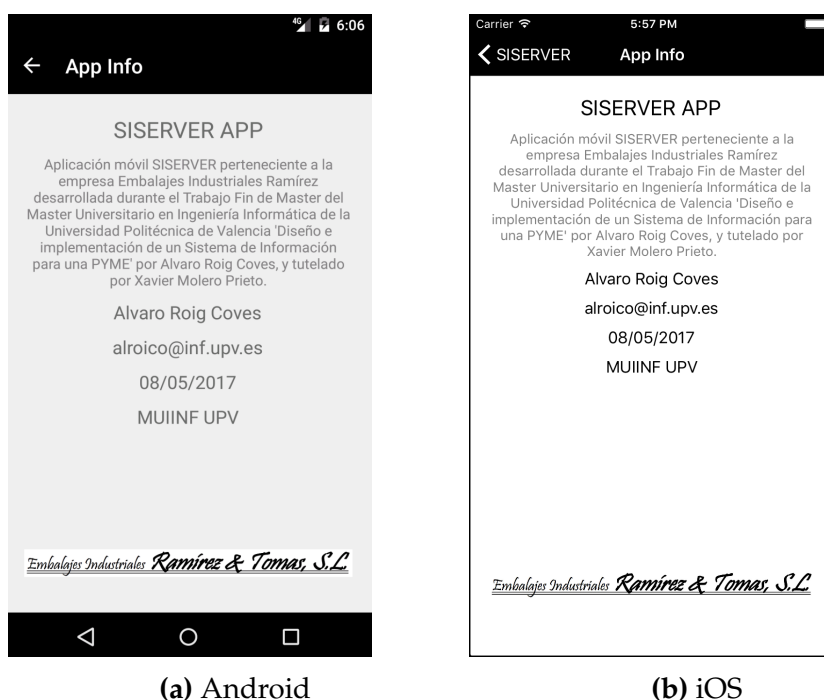
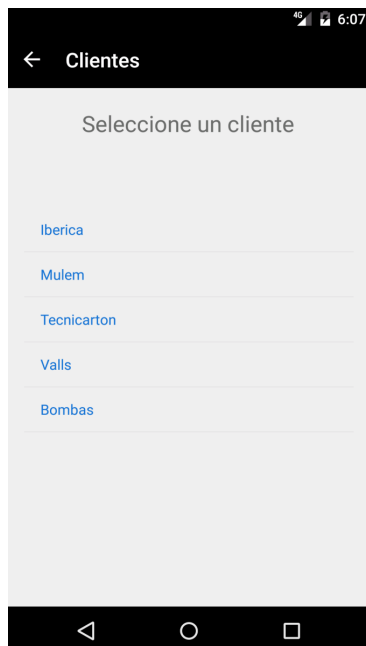


Figura 4.5: Páginas de información de la App

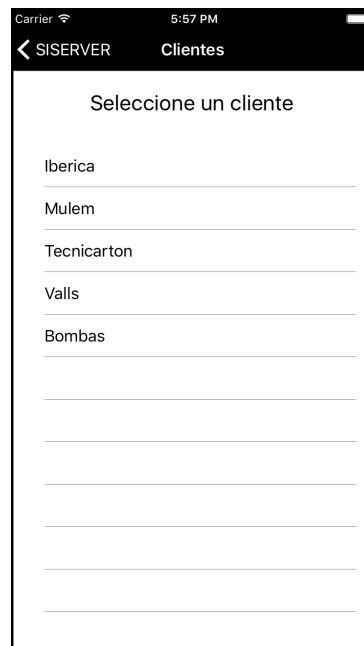
mos filtrar las entradas listadas mediante el cuadro de búsqueda presente en la parte superior de las mismas, y al seleccionar una de las entradas seremos redirigidos a la página que comentamos a continuación, en la que se nos mostrará la información asociada a dicha entrada.

Página de enumeración de las características de un producto

Tras haber seleccionado una entrada de producto seremos redirigidos a la página mostrada en las Figuras 4.8a y 4.8b, cuyo código podemos encontrar en el Anexo B.5, y en la cual se nos mostrará toda la información registrada en la base de datos para dicha entrada, entre la que podremos destacar datos tales como la referencia del producto, su nombre, tipo de producto, descripción, medidas, e incluso los planos o imágenes que hayamos adjuntado a esa entrada, los cuales serán abiertos por el software predeterminado para abrir esos tipos de ficheros (PDF, PNG, JPEG, etc.) por el sistema operativo en cada caso en particular, tras la pulsación sobre el botón correspondiente a cada plano o imagen.

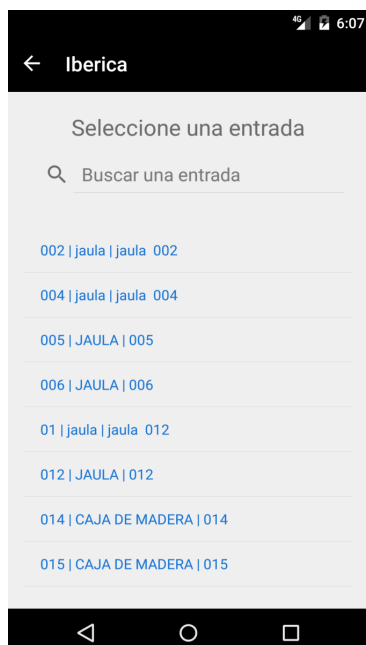


(a) Android

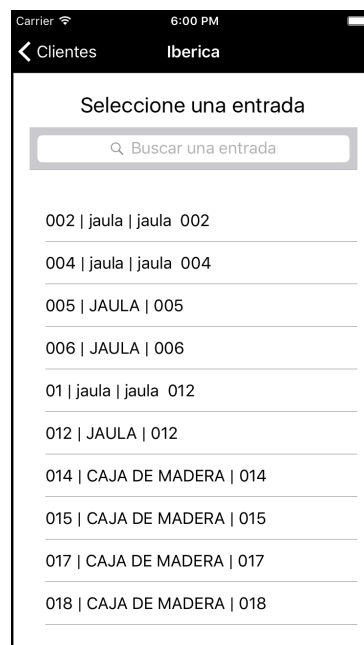


(b) iOS

Figura 4.6: Páginas de listado de clientes de la App



(a) Android



(b) iOS

Figura 4.7: Páginas de listado de productos de la App

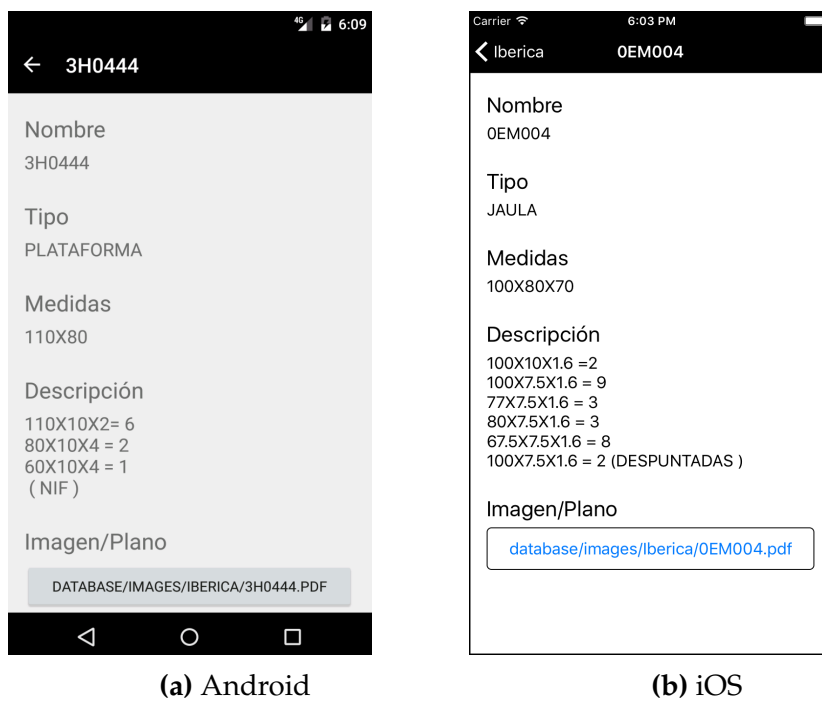


Figura 4.8: Páginas de características de producto de la App

4.3 Comunicación con el servidor

Con el fin de que las aplicaciones móviles que hemos desarrollado se puedan comunicar con el servidor de una manera ágil y sencilla nos disponemos a desarrollar un servicio RESTful mediante el cual facilitar dicha comunicación.

4.3.1. Configuración del servidor

REST (*REpresentational State Transfer*) es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP, el cual nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP, por lo que es increíblemente más simple y convencional que otras alternativas que se han usado en los últimos diez años como SOAP y XML-RPC. De esta forma REST es el tipo de arquitectura más natural y estándar para crear API para servicios orientados a Internet.

Una vez definidos estos conceptos nos disponemos a proceder a la instalación y configuración de las herramientas que nos permitirán llevar a cabo un despliegue de servicios RESTful en nuestro servidor. Para otorgar dicha funcionalidad haremos uso del *framework* Flask para Python.

Flask, cuyo logo observamos en la Figura 4.9, ha sido diseñado como un *framework* extensible, es decir, en su instalación inicial incorpora los paquetes básicos para funcionar, de forma que el resto de las posibles funcionalidades deben ser instaladas como módulos separados, lo que nos permite desarrollar aplicaciones de tamaño reducido incluyendo únicamente la funcionalidad que necesitemos en nuestros proyectos. Mediante este *framework* desarrollaremos una pequeña aplicación en Python que será ejecutada en nuestro servidor, y sobre la cual podremos realizar llamadas para consultar información almacenada en la base de datos, la cual nos será devuelta en formato JSON para su procesamiento en el conjunto de aplicaciones que llamen a esta interfaz RESTful.



Figura 4.9: Logo de Flask

En primer lugar comenzaremos por instalar la utilidad `pip`, mediante la cual procederemos a descargar e instalar los componentes de Flask; por otra parte, también necesitaremos el módulo de MySQL para Flask, por lo que también procederemos a su instalación. Para todo ello ejecutaremos en nuestro servidor las órdenes de consola mostradas en la Tabla 4.1.

```
$ sudo apt-get install python-pip
$ sudo pip install flask
$ sudo pip install flask-mysql
```

Tabla 4.1: Instalación de Flask

4.3.2. Diseño del servicio RESTful

En esta sección nos centraremos en el diseño de una aplicación que nos ofrezca la funcionalidad asociada a un servicio RESTful.

Desarrollo de la aplicación Flask

Una vez hemos instalado los componentes necesarios para la ejecución de programas que hagan uso de Flask, nos disponemos a desarrollar un programa que, basado en Python, y mediante el uso de dicho *framework*, se dedique a atender peticiones en un determinado puerto de nuestro servidor, con el fin de llevar a cabo consultas en la base de datos y responder al cliente que realice dicha solicitud.

Para ello, tal como podemos observar en el Anexo C.1, en la parte superior del fichero indicamos la importación explícita de los módulos Flask para trabajar con la base de datos MySQL, con tipos de datos JSON, y con sesiones de usuario, mediante las instrucciones `from flaskext.mysql import MySQL`, `from flask import json, jsonify`, y `from flask import session`, respectivamente.

Seguidamente, como puntos a destacar dentro de este código, crearemos un objeto MySQL, el cual configuraremos con los datos de conexión a la base de datos instalada en nuestro servidor en las líneas de la 22 a la 27 del código, indicaremos el *secretkey* de la aplicación en la línea 122, el cual es una clave para el cifrado de las comunicaciones, y estableceremos la ejecución y escucha de la aplicación en el puerto 200 mediante la instrucción presente en la línea 123. De esta forma, el procedimiento para llamar a una función de esta interfaz será hacer una petición a la dirección IP del servidor, en el puerto 200, indicando como parámetro la función a ejecutar o incluso los argumentos necesarios para su ejecución; además, es importante destacar que se han configurado las llamadas a las funciones del programa para que solo acepten métodos de llamada del tipo POST.

Por otra parte destacamos la funcionalidad de la aplicación distinguiendo entre las siguientes cinco funciones programadas:

- Función de inicio (/): Esta función nos ofrece información acerca del autor, fecha de creación, y un pequeño resumen de las funciones presentes en la interfaz y los argumentos necesarios o necesidades de las mismas para ser ejecutadas. Un ejemplo de llamada a la función podría ser:
`http://192.168.1.200:200/.`

- **Función de autenticación de usuario (/auth):** Mediante esta función, pasando como parámetros de la misma un nombre de usuario y una contraseña, se comprueba la existencia de dicho usuario en la base de datos. El resultado de la ejecución de esta función será la devolución de un valor booleano, el cual nos indicará si el usuario está dado de alta o no en la base de datos. Un ejemplo de llamada a esta función para la autenticación del usuario *user* con contraseña *password* sería la siguiente:
`http://192.168.1.200:200/auth?usr=user & pwd=password.`
- **Función de cierre de sesión de usuario (/clear):** Mediante la llamada a esta función se limpia la información de usuario de la sesión en curso. Esta función solo puede ser llamada una vez hecha la llamada a la función /auth habiendo resultado esta en una autenticación válida. Un ejemplo de llamada a la función podría ser el siguiente: `http://192.168.1.200:200/clear.`
- **Función para el listado de clientes (/listClients):** Mediante la llamada a esta función se nos devuelve la lista de clientes dados de alta en la base de datos. La ejecución de esta función requerirá de una autenticación de usuario válida, y se nos devolverá una cadena JSON que contendrá el listado de todos los clientes dados de alta en la base de datos. Un ejemplo de llamada a la misma podría ser: `http://192.168.1.200:200/listClients.`
- **Función para el listado de entradas asociadas a un cliente (/listEntry):** La llamada a esta función requerirá de una autenticación de usuario válida de forma previa, y nos permitirá obtener toda la información de las entradas de la base de datos asociadas a productos de un cliente en concreto, el cual deberemos indicar como parámetro de la llamada. El resultado de ejecutar esta función será, en caso de que el identificador de cliente pasado como parámetro sea válido, la devolución de una cadena JSON que contenga todos los datos de las entradas de productos asociadas al cliente indicado. Un ejemplo de llamada a la función podría ser el siguiente:
`http://192.168.1.200:200/listEntry?clientID=2.`

Ejecución automática del servicio RESTful

Una vez diseñada la aplicación Flask nos disponemos a configurar el servidor para que dicha aplicación sea autoiniciada en cada arranque de la Raspberry Pi. Para ello hemos situado el programa en Python que hace uso del *framework* Flask en el directorio `home` del usuario administrador de la Raspberry Pi. Por otra parte, hemos generado un *script* en el directorio `/etc/init.d/` del servidor, cuyo código podemos observar en la Tabla 4.2, y el cual ejecutará la aplicación Python al arranque del sistema operativo.

Finalmente, con el fin de dar permisos de ejecución al *script* que hemos generado, y para actualizar el registro de arranque del SO, hemos de ejecutar las

órdenes de consola `sudo chmod 755 /etc/init.d/initRestFullApi.sh`, y `sudo update-rc.d initRestFullApi.sh defaults`, respectivamente.

```
#!/bin/sh
# initRestFullApi.sh
# Script to autorun the RESTful API
cd /home/pi/
sudo python restfulApi.py
```

Tabla 4.2: Autoejecución del servicio RESTful

4.3.3. Publicación de las aplicaciones

Una vez diseñada la aplicación móvil y desarrollado el servicio RESTfull de comunicaciones en el servidor, nos disponemos a publicar la aplicación en sus diferentes versiones para Android e iOS, de tal forma que estas estén accesibles para el conjunto de usuarios de la empresa.

De esta forma debemos distinguir entre dos casos para cada uno de los dos sistemas operativos mencionados anteriormente. Cabe la pena destacar que en ningún caso vamos a publicar las aplicaciones en los respectivos mercados de aplicaciones de cada uno de los sistemas operativos mencionados, ya que pretendemos que dichas aplicaciones estén únicamente disponibles para los trabajadores de la empresa.

En el caso de Android publicaremos un archivo apk, el cual es un tipo de fichero orientado a la instalación local de aplicaciones en dicho sistema operativo. Para ello, con el proyecto de Xamarin abierto, tal como observamos en la Figura 4.10, pulsaremos el botón derecho del ratón sobre el paquete de Android de nuestro proyecto, pulsando seguidamente la opción Archivo para publicar, la cual nos permitirá firmar el compilado de la aplicación y almacenarlo en formato apk, o bien, publicarlo en el Play Store de Google. En nuestro caso almacenaremos el fichero de instalación local apk, el cual guardaremos en los directorios del servidor web de nuestra Raspberry Pi con el fin de ofrecer su descarga desde la propia página web de gestión que hemos desarrollado.

Con respecto a la publicación de la aplicación para el sistema operativo iOS, vale la pena destacar que así como para la publicación de aplicaciones en Android, al no ser esta publicada en el Play Store de Google podíamos autofirmar el compilado; en el caso de iOS resulta algo más complejo, ya que precisamos de una licencia de desarrollo de Apple en caso de que queramos que dicha aplicación pueda ser utilizada fuera del plazo de desarrollo de la licencia gratuita de desarrollador, la cual únicamente nos permite mantener operativa la aplicación durante unos pocos días. Es decir, para poder compilar la aplicación para iOS y que esta esté siempre operativa debemos adquirir una licencia de desarrollador de Apple, lo cual no vamos a realizar por ahora en este proyecto ya que en

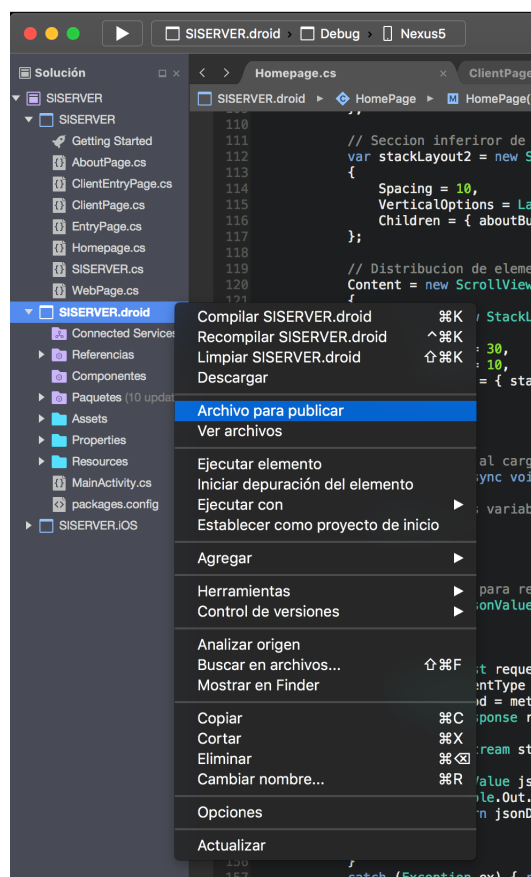


Figura 4.10: Publicación de la App para Android

un principio los terminales que serán utilizados en la empresa para consultar la información del servidor harán uso del sistema operativo Android.

A pesar de ello, y ante la posibilidad de una futura publicación de la aplicación para iOS, nos disponemos a describir el proceso de publicación del proyecto de Xamarin para dicho sistema operativo. Para ello, tal como hicimos en el caso de la publicación de la aplicación para Android, y como podemos observar en la Figura 4.11, pulsaremos con el botón derecho del ratón sobre el paquete de iOS del proyecto, pulsando además sobre la opción Archivo para publicar del submenú. Tras esta acción se procederá a la compilación de la aplicación con la configuración indicada, pudiendo firmar de forma posterior el compilado con nuestra firma de desarrollador premium de Apple, y dejándonos finalmente publicar la aplicación en el App Store de Apple, o bien guardar un fichero de instalación local en formato ipa, el cual de igual forma que en el caso de la publicación de Android, almacenaríamos en los directorios del servidor web para ofrecerlos de forma posterior desde la página web de gestión del servidor que hemos diseñado anteriormente.

Finalmente, tras haber procedido a la publicación de la aplicación móvil en sus dos versiones para los diferentes sistemas operativos en los que hemos tra-

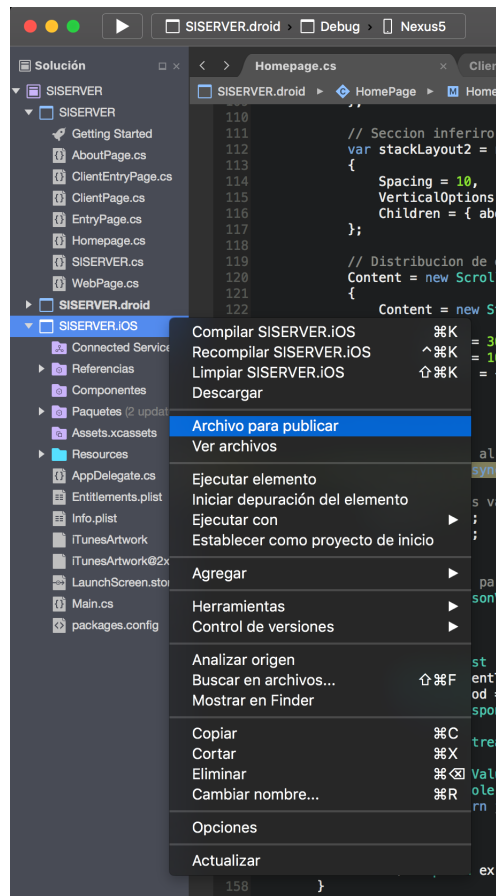
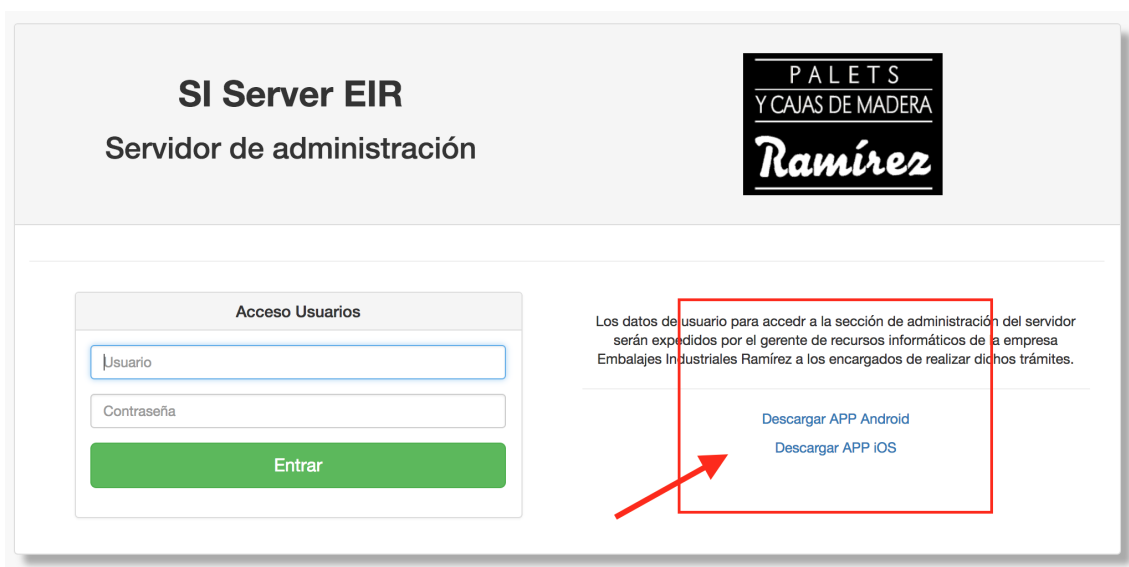


Figura 4.11: Publicación de la App para iOS

bajado, nos disponemos a hacerlas accesibles para los usuarios de la empresa, de tal forma que estas puedan ser descargadas e instaladas en los terminales de los trabajadores de la empresa destinados a operar con ellas.

Para ello nos disponemos a modificar el código presente en el Anexo A.1, en el cual añadiremos, tal como podemos observar en la Figura 4.12, dos enlaces a los ficheros de instalación local de cada una de las versiones publicadas de la aplicación. De esta forma, con el fin de instalar la aplicación en un terminal móvil, únicamente habremos de acceder a la página web desde el terminal en el que deseemos instalar la aplicación y pulsar sobre el enlace correspondiente a nuestro sistema operativo.



The screenshot displays the 'SI Server EIR' web interface. At the top left, the text reads 'SI Server EIR' and 'Servidor de administración'. At the top right is the logo for 'PALETS Y CAJAS DE MADERA Ramírez'. Below the header is a login section titled 'Acceso Usuarios' with input fields for 'Usuario' and 'Contraseña', and a green 'Entrar' button. To the right of the login form is a text block: 'Los datos de usuario para acceder a la sección de administración del servidor serán expedidos por el gerente de recursos informáticos de la empresa Embalajes Industriales Ramírez a los encargados de realizar dichos trámites.' Below this text are two links: 'Descargar APP Android' and 'Descargar APP IOS'. A red box highlights these links, and a red arrow points to the 'Descargar APP Android' link.

Figura 4.12: Publicación de la App en la web

CAPÍTULO 5

Presupuesto del proyecto

En este capítulo nos disponemos a realizar un presupuesto detallado tanto del conjunto de componentes que necesitaremos para llevar a cabo la construcción de nuestro sistema de información, como de la estimación de horas de trabajo que hemos invertido durante el desarrollo del proyecto.

5.1 Material necesario

Con el fin de implantar el servidor precisaremos de componentes tales como una Raspberry Pi 3 Modelo B, una tarjeta MicroSD de más de 4 GiB de capacidad, un alimentador de corriente MicroUSB, un cable de red RJ45, un disipador de calor, y una carcasa para la Raspberry Pi. Además, se incluye en el presupuesto una tablet con el fin de ser usada a modo de terminal móvil en el cual instalar la aplicación móvil de consulta para los operarios.

Vale la pena destacar que el coste en hardware de este proyecto no es muy elevado, ya que tal como comentamos en la Sección 3.2.3 nuestro servidor, el cual podemos observar en la Figura 5.1, será de tipo *headless*, de tal forma que al no precisar de otro tipo de componentes tales como un monitor o cualquier otro tipo de periférico para trabajar con el servidor, garantizamos una inversión mínima en componentes a la empresa. Por otra parte se suponen ya adquiridos anteriormente por parte de la empresa otros elementos necesarios para el desarrollo del proyecto, tales como una infraestructura de red inalámbrica en las instalaciones de la empresa mediante la cual se comunicarán nuestro servidor y las aplicaciones cliente de los usuarios.

Finalmente, destacamos que los costes asociados al material necesario para realizar este proyecto, han sido asumidos por parte de *Embalajes Industriales Ramírez Tomás S.L.*, la empresa que hará uso y se beneficiará de este proyecto tras su finalización.



Figura 5.1: Servidor Raspberry Pi

5.2 Estimación de horas de trabajo

Debemos tener en cuenta que la mayor parte del coste de este proyecto será el asociado al desarrollo de software y a la configuración de servicios en el servidor, por lo que el presupuesto queda abierto a un considerable incremento en caso de ser desarrollado por terceras personas, las cuales dispondrían cuál sería el precio por hora asociado a la dedicación en este proyecto.

De esta forma diferenciaremos el tiempo dedicado a el desarrollo de este proyecto entre horas dedicadas a la instalación y configuración de servicios en el servidor, horas dedicadas al diseño de la aplicación web de gestión, horas dedicadas al diseño de la aplicación móvil multiplataforma, y horas dedicadas al testeado del mismo Sistema de Información completo.

Destacamos además que con el fin de obtener un precio normalizado con el que valorar la hora de trabajo en labores de programación e instalación y configuración de servidores y servicios nos dirigimos al COIICV¹, desde el cual se nos comunicó que según el artículo 14 de la Ley sobre Colegios Profesionales, "Los Colegios Profesionales y sus organizaciones colegiales no podrán establecer baremos orientativos ni cualquier otra orientación, recomendación, directriz, norma o regla sobre honorarios profesionales", de tal forma que los precios correspondientes a la remuneración económica por hora de trabajo han sido estimados por nuestra parte en base a lo que consideramos un precio justo para la titulación que estamos cursando.

Finalmente, tal como muestra la Figura 5.2, para el desarrollo de este proyecto se ha invertido un total de 205,52 euros en la adquisición del material necesario para la implantación del servidor en el cual almacenaremos el propio Sistema de Información. Por otro lado se han estimado de forma aproximada un total de trescientas noventa horas de trabajo distinguiendo entre las diferentes tareas mencionadas anteriormente, entre las cuales, aproximando precios de entre 15 y 20 euros por hora de trabajo, suman un total de 6933,30 euros, quedando co-

¹Colegio Oficial de Ingenieros en Informática de la Comunidad Valenciana. Más información en <https://www.coiicv.org/>

mo importe final del presupuesto de este proyecto un total de 7138,82 euros de inversión económica.

Ítem	Precio Unitario	Unidades	Subtotal
Raspberry Pi 3 Model B	33,95 €	1	33,95 €
Fuente de alimentación MicroUSB	9,10 €	1	9,10 €
MicroSD Memory Card 16GB	12,00 €	1	12,00 €
Cabe de red RJ45	5,00 €	1	5,00 €
Kit disipador de calor de aluminio	2,00 €	1	2,00 €
Carcasa oficial para Raspberry	7,80 €	1	7,80 €
Tablet táctil 8"	100,00 €	1	100,00 €
Total sin IVA			169,85 €
IVA (21%)			35,67 €
TOTAL MATERIAL			205,52 €

Tarea	Precio/hora	Horas	Subtotal
Instalación y configuración de servicios	15,00 €	50	750,00 €
Diseño de la aplicación web de gestión	15,00 €	150	2.250,00 €
Diseño de la aplicación móvil multiplataforma	15,00 €	150	2.250,00 €
Testeo del mismo Sistema de Información	12,00 €	40	480,00 €
Total sin IVA			5.730,00 €
IVA (21%)			1.203,30 €
TOTAL TAREAS			6.933,30 €

IMPORTE FINAL	7.138,82 €
----------------------	-------------------

Figura 5.2: Presupuesto final del proyecto

CAPÍTULO 6

Estudios de consumo y valoración de los usuarios

En este capítulo nos disponemos a llevar a cabo tanto un estudio del consumo energético de nuestro servidor con el fin de compararlo con los consumos aproximados de un servidor real, como un estudio de valoración por parte de los usuarios del Sistema de Información con el fin de obtener información directa acerca de la opinión de los trabajadores de la empresa al implantar y utilizar los elementos desarrollados durante este proyecto.

6.1 Consumo energético del servidor

Uno de los puntos a tener en cuenta a la hora de proceder a la adquisición e instalación de un servidor en cualquier empresa se trata del consumo energético del mismo, ya que generalmente el uso de servidores con cierta potencia repercute en un incremento considerable en el coste anual de consumo eléctrico.

Vale la pena destacar que debido a las necesidades de la empresa hemos hecho uso de un servidor de bajo coste y consumo, por lo que seremos capaces de asegurar a la empresa un incremento del coste por consumo eléctrico mínimo.

Dicho esto nos disponemos a llevar a cabo el cálculo teórico del coste que supondría para la empresa hacer uso del servidor en cuanto a consumo eléctrico se refiere, para el cual hemos aproximado un consumo medio de 3,5 W por parte de la Raspberry Pi, un precio aproximado de 0,15 €/kWh, las ocho horas de una jornada laboral completa, y un uso del servidor durante todos los días del año.

Una vez realizados los cálculos, tal como podemos observar en la Ecuación 6.1, obtenemos como resultado del coste anual correspondiente al consumo eléctrico que supondría trabajar con el servidor que hemos implementado un total de

1,533 €/año, de tal forma que conseguimos poner en marcha los servicios necesarios para la implantación del Sistema de Información por un coste mínimo.

$$(3,5 \div 1000)\text{kW} \times 0,15\text{€/kWh} \times 8\text{horas} \times 365\text{días} = 1,533 \text{ €/año} \quad (6.1)$$

6.2 Valoración por parte de los usuarios

Llegados a este punto ya hemos desarrollado tanto el Sistema de Información web como la aplicación móvil multiplataforma, así como la instalación y configuración del servidor web y sus servicios, por lo que nos disponemos a probar *in situ* el conjunto de funcionalidades y servicios desarrollados.

6.2.1. Formularios de consulta

De esta forma hemos procedido a implantar el servidor en la empresa, otorgando a un número reducido de empleados sus diferentes claves de acceso a las aplicaciones en función de sus roles, con el fin de permitir su uso a un nivel real de trabajo, pero de forma controlada para analizar su rendimiento en busca de posibles errores o características a modificar.

Tras haber pasado dos semanas trabajando con los elementos desarrollados en este proyecto, hemos hecho llegar a varios de los trabajadores de la empresa que han estado utilizando estos elementos un formulario mediante el cual podían manifestar su opinión acerca de los servicios desplegados, así como reflejar sus opiniones de mejora, o indicar características que no les hayan gustado.

Dicho formulario lo podemos encontrar en el Anexo D.1, y en él distinguimos una primera sección en la que preguntamos por el nombre del empleado, su rol dentro de la empresa y qué aplicaciones ha utilizado; una segunda sección formada por seis preguntas de valoración subjetiva en una escala de 0 a 5 puntos en la que pretendemos obtener información acerca de la opinión de los trabajadores en aspectos tales como las interfaces de usuario, la usabilidad de las aplicaciones, o sobre la mejora que ha supuesto el uso de las mismas en su labor; y una sección final en la que el usuario puede proponer alguna mejora o indicar aspectos que no le hayan gustado del Sistema de Información, de tal forma que permitimos a los usuarios detectar posibles errores o incluso aportar nuevas ideas para mejorar el propio sistema.

6.2.2. Resultados obtenidos

El formulario fue enviado a tres trabajadores con diferentes roles dentro de la empresa con el fin de obtener una opinión global por parte de los mismos. Dicho formulario, tal como podemos observar en las Figuras 6.1, 6.2 y 6.3, fue cumplimentado por parte del gerente, por un encargado y por un operario, de esta forma obtenemos la valoración de personal que solo ha hecho uso de la aplicación móvil de consulta, de personal que solo ha hecho uso de la página web de gestión, y de personal que ha hecho uso de las dos aplicaciones que hemos desarrollado.

Pregunta	Valoración					
	0	1	2	3	4	5
¿Considera que el Sistema de Información (SI) desplegado le ayuda con sus labores diarias?						X
¿La forma de trabajar con el SI le parece sencilla?					X	
¿El tiempo que tarda en consultar información en la base de datos es relativamente corto?						X
¿El SI cumple con sus necesidades de obtención de información de productos?						X
¿Las interfaces de usuario de las aplicaciones del SI le parecen apropiadas?					X	
¿Le parece sencillo navegar entre diferentes secciones de las aplicaciones del SI?						X

	N/A	NO	SI
¿A detectado algún elemento del Sistema de Información que no sea correcto o se pueda mejorar?			X

En caso afirmativo, por favor, indíquelo a continuación:

SERÍA INTERESANTE QUE TODA LA INFORMACIÓN ESTE COLGADA EN LA NUBE PARA TENER ACCESO DESDE CUALQUIER LUGAR Y NO DEPENDER DE LA CONEXIÓN POR WIFI

Figura 6.1: Formulario de valoración cumplimentado por el gerente

Analizando estos formularios podemos observar, tal como mostramos en la Figura 6.4, cómo la valoración media de los servicios desarrollados es generalmente alta, siendo esta de un 8,7 sobre 10 tras la ponderación de todas las valoraciones de los encuestados. Por otra parte se han sugerido un par de mejoras en los servicios desarrollados; los cuales son, tal como podemos observar en la Figura 6.2, dotar de la posibilidad de realizar apagados y reinicios seguros del servidor por parte de los administradores del mismo, funcionalidad que hemos desarrollado tras esta proposición en el Apartado 3.2.3, y la propuesta, tal como podemos observar en la Figura 6.1, de hacer accesibles todos estos servicios desde una red externa a la empresa, sugerencia que vamos a posponer como posibilidad de mejora futura ante la falta de tiempo para el desarrollo de este proyecto.

Pregunta	Valoración					
	0	1	2	3	4	5
¿Considera que el Sistema de Información (SI) desplegado le ayuda con sus labores diarias?					X	
¿La forma de trabajar con el SI le parece sencilla?						X
¿El tiempo que tarda en consultar información en la base de datos es relativamente corto?					X	
¿El SI cumple con sus necesidades de obtención de información de productos?						X
¿Las interfaces de usuario de las aplicaciones del SI le parecen apropiadas?						X
¿Le parece sencillo navegar entre diferentes secciones de las aplicaciones del SI?				X		
		N/A	NO	SI		
¿A detectado algún elemento del Sistema de Información que no sea correcto o se pueda mejorar?						X

En caso afirmativo, por favor, indíquelo a continuación:

NECESITAMOS UN ELEMENTO PARA PODER APAGAR
y REINICIAR EL SERVIDOR

Figura 6.2: Formulario de valoración cumplimentado por un encargado

Pregunta	Valoración					
	0	1	2	3	4	5
¿Considera que el Sistema de Información (SI) desplegado le ayuda con sus labores diarias?					X	
¿La forma de trabajar con el SI le parece sencilla?					X	
¿El tiempo que tarda en consultar información en la base de datos es relativamente corto?						X
¿El SI cumple con sus necesidades de obtención de información de productos?				X		
¿Las interfaces de usuario de las aplicaciones del SI le parecen apropiadas?						X
¿Le parece sencillo navegar entre diferentes secciones de las aplicaciones del SI?				X		
		N/A	NO	SI		
¿A detectado algún elemento del Sistema de Información que no sea correcto o se pueda mejorar?			X			

En caso afirmativo, por favor, indíquelo a continuación:

Figura 6.3: Formulario de valoración cumplimentado por un operario

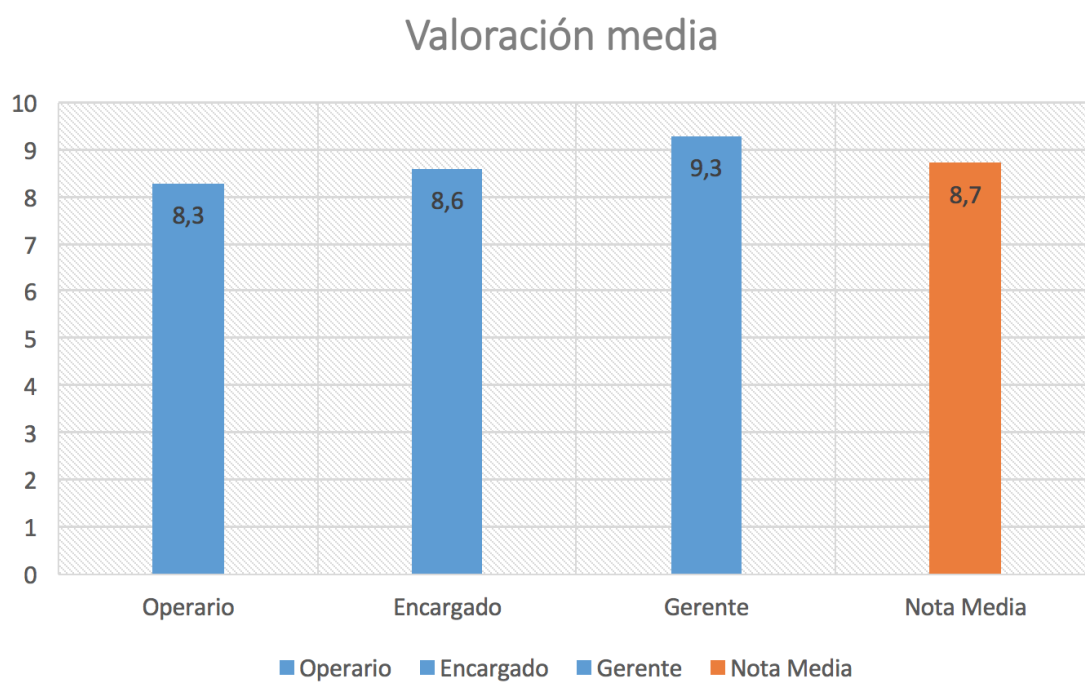


Figura 6.4: Notas de los formularios de valoración de usuario - Escala 0 a 10

CAPÍTULO 7

Conclusiones

En este último capítulo incluimos una sección en la que describimos los diferentes problemas y contratiempos que hemos sufrido durante el transcurso del trabajo. Además, se plantean las consideraciones finales obtenidas a partir del trabajo desarrollado, realizando un resumen final, y analizando el conjunto de funcionalidades implementadas.

7.1 Objetivos Alcanzados

Una vez llegamos a este punto podemos afirmar que hemos instalado y configurado un servidor Linux en una Raspberry Pi, al cual hemos dotado de la posibilidad de ofrecer servicios destinados al almacenamiento de información en bases de datos, servicios de transferencia de ficheros mediante el protocolo FTP, o incluso de la posibilidad de almacenar páginas web mediante un servicio web Apache.

Además, hemos abordado el diseño e implementación de un Sistema de Información digital programado de forma predominante en los lenguajes de programación PHP, HTML, y Java Script, el cual será administrado por los trabajadores de la empresa destinados a llevar a cabo este tipo de funciones, y residirá alojado en el servidor que hemos configurado.

Por otra parte hemos diseñado y desarrollado una aplicación móvil multiplataforma mediante el *framework* Xamarin.Forms de Xamarin, mediante el cual hemos podido implementar la aplicación de forma sencilla para sistemas operativos tanto Android como iOS, y mediante la cual ofrecemos a los trabajadores de la empresa la capacidad de llevar a cabo consultas a la base de datos de una forma ágil y sencilla instalando esta aplicación en sus dispositivos móviles.

Destacamos además que hemos desarrollado un servicio destinado a realizar las funciones propias de un API RESTful haciendo uso del *framework* Flask y el

lenguaje de programación Python, el cual ha sido implementado en el servidor, y mediante el uso del cual llevamos a cabo las comunicaciones entre las aplicaciones móviles desarrolladas y el propio servidor, de tal forma que las consultas a la base de datos serán realizadas por parte de dicha API RESTful, evitando que sean las aplicaciones de usuario las que realicen consultas de forma directa en la base de datos.

Finalmente, tras haber llevado a cabo tanto un estudio de consumo energético, como un estudio de valoración de uso por parte de los usuarios de la empresa, el cual, tal como analizamos en el Apartado 6.2.2, indicó una alta satisfacción por parte de los mismos, concluimos esta sección afirmando haber logrado alcanzar de forma satisfactoria los objetivos planteados al inicio del proyecto, quedando cubiertas las necesidades de digitalización requeridas por parte de la empresa, y habiendo obtenido en líneas generales un aumento de la producción en la misma debido al ahorro de tiempo que otorga el uso de un sistema digital de estas características, el cual ha resultado en un Sistema de Información diseñado a medida en base a las necesidades de la empresa, y que ha sido desarrollado mediante el despliegue e integración de diferentes tecnologías con el fin de lograr un objetivo específico.

7.2 Contratiempos y problemas sufridos

Con respecto a las dificultades encontradas durante el desarrollo del proyecto podemos destacar, por ejemplo, la necesidad identificada por parte de los usuarios de la empresa en los tramos finales de implantación del proyecto (presente en la Sección 6.2.2), consistente en la necesidad de realizar reinicios o apagados seguros del servidor; por lo que se hubieron de realizar, tal como podemos observar en el Apartado 3.2.3, modificaciones en el sistema de gestión web del servidor dotándolo de la funcionalidad de llevar a cabo dichos reinicios y apagados del servidor por parte de los administradores del Sistema de Información.

7.3 Posibles mejoras

Durante el estudio de valoración por parte de los usuarios de la empresa del conjunto de servicios y aplicaciones desarrolladas en este proyecto también se sugirió, tal como podemos observar en la Sección 6.2.2, la posibilidad de poder hacer uso de las aplicaciones desarrolladas desde una red externa a la empresa, es por ello que destacamos como posibilidad de futura mejora esta característica, la cual podríamos llevar a cabo realizando un despliegue de todos los servicios configurados a una plataforma *cloud*, o bien generando un servicio de VPN que nos permitiera realizar conexiones a la red de la empresa desde redes externas de una forma segura.

Bibliografía

- [1] Can Bilgin. *Mastering Cross-Platform Development with Xamarin. Community Experience Distilled*. Packt Publishing Ltd. Marzo, 2016.
- [2] Ángel Cobo. *PHP y MySQL. Tecnología para el desarrollo de aplicaciones web*. Ediciones Díaz de Santos, 2005.
- [3] Alejandro Esquiva Rodríguez. *Geeky Theory. Tutorial Raspberry Pi - 9. Servidor FTP*. Consultado en <https://geekytheory.com/tutorial-raspberry-pi-9-servidor-ftp>. Mayo, 2013.
- [4] Alejandro Esquiva Rodríguez. *Geeky Theory. Tutorial Raspberry Pi - 15. Instalación de Apache + MySql + PHP*. Consultado en <https://geekytheory.com/tutorial-raspberry-pi-15-instalacion-de-apache-mysql-php>. Agosto, 2013.
- [5] Vicenç Fernández Alarcón. *Desarrollo de sistemas de información: una metodología basada en el modelado*. Edicions UPC, Universitat Politècnica de Catalunya. Marzo, 2010.
- [6] Rick Golden. *Raspberry Pi Networking Cookbook. Quick answers to common problems*. Packt Publishing Ltd. Enero, 2013.
- [7] Miguel Grinberg. *Flask Web Development. Developing Web Applications with Python*. O'Reilly Media, Inc. Abril, 2014.
- [8] Alex Fornuto. *LAMP on Debian 8 (Jessie)*. Consultado en <https://www.linode.com/docs/websites/lamp/lamp-on-debian-8-jessie/>. Diciembre, 2015.
- [9] Olivier Heurtel. *PHP 5.5. Desarrollar un sitio web dinámico e interactivo*. Ediciones ENI, 2014.
- [10] Dan Hermes. *Xamarin Mobile Application Development. Cross-Platform C# and Xamarin.Forms Fundamentals*. Ediciones Apress. Julio, 2015.

- [11] Ministerio de Educación, Cultura, y Deporte. Gobierno de España. *Aulas en red, aplicaciones y servicios: Servidor DNS*. Consultado en http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m2/servidor_dns.html.
- [12] Doug Lowe. *Networking All-in-One For Dummies*. John Wiley & Sons Ed. Noviembre, 2012.
- [13] Matt Richardson y Shawn Wallace. *Getting Started with Raspberry Pi. Make: projects*. O'reilly and Associate Series. O'Reilly Media, 2012.
- [14] Álvaro Roig Coves. Trabajo Final de Grado de Ingeniería Informática: *Diseño e implementación de una máquina Arcade con Raspberry Pi y Arduino*. Riunet UPV, Julio 2015. Accesible desde <http://hdl.handle.net/10251/53962>.
- [15] Eben Upton y Gareth Halfacree. *Raspberry Pi User Guide*. John Wiley and Sons, 2014.

APÉNDICE A

Códigos del portal de administración telemática

A.1 Página de acceso

El siguiente código pertenece a la página de acceso al portal de administración telemática de nuestro servidor, desde la cual el usuario puede introducir su nombre de usuario y contraseña con el fin de verificar sus credenciales en la base de datos del portal de administración, para acceder de forma posterior a la página de gestión del servidor.

```
1
2 #####
3 ##### SISERVER index.php #####
4 #####
5 # Pagina index del servidor web
6 # @auth alroico@inf.upv.es
7 # @date 12/06/2017
8 <?php
9
10 session_start();
11 include_once "conexion.php";
12
13 function verificar_login($user,$password,&$result) {
14     $sql = "SELECT * FROM 'usuarios_web' WHERE user='$user' AND pass='
15         $password'";
16     $rec = mysql_query($sql);
17     $count = 0;
18
19     while($row = mysql_fetch_object($rec))
20     {
21         $count++;
22         $result = $row;
23     }
```

```
24     if ($count == 1)
25     {
26         return 1;
27     }
28
29     else
30     {
31         return 0;
32     }
33 }
34
35 $_SESSION[ 'firmado' ] = 0;
36
37 ?>
38
39 <html>
40 <head>
41     <!--
42         Vista movil desactivada.
43         <meta name="viewport" content="width=device-width, initial-scale
44             =1">
45     -->
46     <meta charset="utf-8">
47     <meta http-equiv="X-UA-Compatible" content="IE=edge">
48     <meta name="description" content="">
49     <meta name="author" content="">
50
51     <title>SI Server EIR – Servidor de administraci&oacute;n</title >
52
53     <!-- Bootstrap Core CSS -->
54     <link href="bower_components/bootstrap/dist/css/bootstrap.min.css"
55         rel="stylesheet">
56
57     <!-- MetisMenu CSS -->
58     <link href="bower_components/metisMenu/dist/metisMenu.min.css" rel=
59         "stylesheet">
60
61     <!-- Custom CSS -->
62     <link href="dist/css/sb-admin-2.css" rel="stylesheet">
63
64     <!-- Custom Fonts -->
65     <link href="bower_components/font-awesome/css/font-awesome.min.css"
66         rel="stylesheet" type="text/css">
67
68     <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and
69         media queries -->
70     <!-- WARNING: Respond.js doesn't work if you view the page via file
71         :// -->
72     <!--[if lt IE 9]>
73         <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/
74             html5shiv.js"></script>
```



```

108         </div>
109         <div class="form-group">
110             <input class="form-
                control"
                placeholder="
                Contraseña" name="
                pass" type="
                password">
111         </div>
112         <!-- Change this to a
                button or input when
                using this as a form
                -->
113         <div>
114             <input name="login"
                type="submit" value
                ="Entrar" class="
                btn btn-lg btn-
                success btn-block">
115         </div>
116     </fieldset>
117 </form>
118 </div>
119 </div>
120 </div>
121 <div class="col-lg-6" align="center" style
    ="margin-top:30px;">
122     <p>Los datos de usuario para acceder a
        la sección de administraci
        ón del servidor ser
        expedidos por el gerente de
        recursos inform
        ticos de la
        empresa Embalajes Industriales
        Ramirez a los encargados de
        realizar dichos tr
        mites.</p
    >
123 <hr>
124 <?php
125     if (!isset($_SESSION['userid']))
126     {
127         if (isset($_POST['login']))
128         {
129             if (verificar_login($_POST['
                user'], $_POST['pass'],
                $result) == 1)
130             {
131                 $_SESSION['userid'] =
                    $_POST['user'];
132                 $_SESSION['ses'] =
                    False;
133                 $_SESSION['firmado'] =
                    1;
134                 echo '<br><div align="
                    center" style="

```

```

135         color: green; "><
        strong>Usuario
        autenticado
        correctamente.</
        strong></div>';
136     echo '<div align="
        center">
        Redirigiendo ...</
        div><br>';
        echo "<meta http-equiv=
        'refresh' content='
        0;url=accesoAdmin.
        php?action=home'>";
137     }
138     else
139     {
140         echo '<br><div align="
        center" style="
        color:red;"><strong
        >Sus datos de
        acceso son
        incorrectos , int&
        eacute;ntelo
        nuevamente.</strong
        ></div><br>';
141         $_SESSION['firmado'] =
        0;
142     }
143     }
144     ?>
145     </div>
146 </div>
147 <p><a href="app/android/com.alroico.siserver.
        apk">Descargar APP Android</a></p>
148 <p><a href="itms-services://?action=download-
        manifest&url=app/ios/manifest.plist">
        Descargar APP iOS</a></p>
149 <!-- <p>Descargar APP iOS</p> -->
150 </div>
151 <!-- /.panel-body -->
152 </div>
153 <!-- /.panel -->
154 </div>
155 <!-- jQuery -->
156 <script src="bower_components/jquery/dist/jquery.min.js"></script>
157 <!-- Bootstrap Core JavaScript -->
158 <script src="bower_components/bootstrap/dist/js/bootstrap.min.js
        "></script>
159 <!-- Metis Menu Plugin JavaScript -->
160 <script src="bower_components/metisMenu/dist/metisMenu.min.js"></
        script>
161 <!-- Custom Theme JavaScript -->
162 <script src="dist/js/sb-admin-2.js"></script>
163

```

```

164
165 </body>
166 </html>
167
168 <?php
169 } else {
170     echo '<strong>Para continuar debe cerrar la sesion previa: </strong
171         >';
172     echo '<a href="logout.php">Cerrar Sesi&oacute;n.</a>';
173 }
?>

```

A.2 Conexión a la base de datos

Este código forma parte del fichero `conexion.php`, el cual se incluye tanto en la página de acceso, como en la página de inicio del portal, y es el encargado de gestionar la conexión con la base de datos en el momento en que el usuario inicia una sesión nueva en el portal.

```

1
2 #####
3 ##### SISERVER conexion.php #####
4 #####
5 # Pagina para conexion a base de datos en el servidor web
6 # @auth alroico@inf.upv.es
7 # @date 12/06/2017
8
9 <?php
10 // Datos para la conexion a mysql
11 define('DB_SERVER','localhost');
12 define('DB_NAME','tfm');
13 define('DB_USER','tfmUser');
14 define('DB_PASS','*****');
15 $con = mysql_connect(DB_SERVER,DB_USER,DB_PASS);
16 mysql_select_db(DB_NAME,$con);
17 // Query mysql para evitar errores en la lectura de caracteres
18 // especiales
19 mysql_query("SET NAMES 'utf8'", $con);
?>

```

A.3 Cierre de sesión de usuario

Este código forma parte del fichero `logout.php`, el cual se incluye en la página de inicio del portal, y es el encargado de cerrar la sesión de trabajo del usuario de una forma segura en el momento en el que el usuario decide cerrar la sesión.


```
1
2 #####
3 ##### SISERVER logout.php #####
4 #####
5 # Pagina de logout para el servidor web
6 # @auth alroico@inf.upv.es
7 # @date 12/06/2017
8
9 <?php
10     session_start();
11     session_destroy();
12
13     header('location: index.php');
14 ?>
```

A.4 Reinicio y apagado del servidor

Este código forma parte del fichero `specialLogout.php`, el cual se incluye en la página de administración del portal, y es el encargado de programar un reinicio u apagado del servidor en función de la opción elegida por parte del usuario desde el portal de administración.

```
1
2 #####
3 ### SISERVER specialLogout.php ###
4 #####
5 # Pagina para desconexiones especiales del servidor web
6 # @auth alroico@inf.upv.es
7 # @date 12/06/2017
8
9 <?php
10
11 session_start();
12
13 // Funcion para reiniciar el servidor
14 function reboot(){
15     session_destroy();
16     sleep(1);
17     system('sudo /sbin/reboot -h now');
18 }
19
20 // Funcion para apagar el servidor
21 function shutdown(){
22     session_destroy();
23     sleep(1);
24     system('sudo /sbin/shutdown -h now');
25 }
26
27 // Control de sesion para asegurar que el usuario se loguea
28 if (isset($_SESSION['firmado'])) {$firmado = $_SESSION['firmado'];}
```

```

29 if ($firmado==0||$firmado==''||is_null($firmado)){
30     header('Location: index.php');
31     die();
32 }
33 else{
34     if(isset($_GET['f'])&&$_GET['f']!=''){
35         $function = $_GET['f'];
36         switch ($function) {
37             case 'r':
38                 $_SESSION['firmado'] = 0;
39                 unset($_SESSION['firmado']);
40                 register_shutdown_function('reboot');
41                 break;
42             case 's':
43                 $_SESSION['firmado'] = 0;
44                 unset($_SESSION['firmado']);
45                 register_shutdown_function('shutdown');
46                 break;
47             default:
48                 echo 'Funcion no definida';
49                 break;
50         }
51     }
52     else{
53         echo 'Datos no definidos';
54     }
55 }
56
57 ?>

```

A.5 Librería de funciones

El siguiente código forma parte del fichero functions.php, el cual es incluido en el fichero accesoAdmin.php y hace las veces de una librería de funciones generada con el fin de contener una gran parte de las tareas más recurrentes realizadas por el programa de administración del servidor.

```

1
2 #####
3 ##### SISERVER functions.php #####
4 #####
5 # Libreria de funciones del servidor web
6 # @auth alroico@inf.upv.es
7 # @date 12/06/2017
8
9 <?php
10
11 // Funcion para eliminar un fichero fisico
12 function dropPic($imagen){
13     if(file_exists($imagen)){

```

```

14     unlink($imagen);
15 }
16 }
17
18 function uploadPicByImgNum($filename, $name, $tipo, $id, $table, $con,
19     $imgNum) {
20     // Si existe lo anyadimos
21     $imgNum = "imagen".$imgNum;
22     $sqlCheck = "SELECT * FROM ".$table." WHERE id='".$id."'";
23     $queryCheck = mysql_query($sqlCheck, $con);
24     if($queryCheck){
25         $rowCheck = mysql_fetch_array($queryCheck);
26         $img = $rowCheck[$imgNum];
27         list($tipo, $formato) = split('[/]', $tipo);
28         $currentdir = getcwd();
29         $filePath = $currentdir .'/database/images/'.$table.'/'.$
30             $name;
31         $filePathTemp = $filename.'.'.$formato;
32         $sqlAdd = "UPDATE ".$table." SET ".$imgNum."='database/
33             images/'.$table.'/'.$name.'" WHERE id='".$id."'";
34         $retvalAdd = mysql_query($sqlAdd, $con);
35         if($retvalAdd){
36             move_uploaded_file($filename, $filePath);
37         }
38         else {
39             echo '
40                 <div class="alert alert-danger alert-dismissable"
41                     align="center">
42                     <button type="button" class="close" data-
43                         dismiss="alert" aria-hidden="true">&times
44                     </button>
45                     Entrada anyadida pero imagen no cargada ,
46                     contacte con el administrador comunic&
47                     aacute;ndole el error.
48                 </div>
49             ';
50         }
51     }
52 }
53
54 // Funcion para guardar los avatares de los usuarios
55 function uploadPicUser($filename, $name, $tipo, $id, $user, $table,
56     $con){
57     list($tipo, $formato) = split('[/]', $tipo);
58     $currentdir = getcwd();
59     $filePath = $currentdir .'/database/images/users';
60     if(!file_exists($filePath)){
61         mkdir($filePath,0777,true);
62     }
63     $filePathTemp = $filename.'.'.$formato;
64     $sqlAdd = "UPDATE ".$table." SET imagen='database/images/users/"
65         $user.'.'.$formato.'" WHERE id='".$id."'";
66     $retvalAdd = mysql_query($sqlAdd, $con);
67     if($retvalAdd){

```

```

58     move_uploaded_file($filename, $filePath . '/' . $user . '/' . $formato)
59     ;
60 }
61 else {
62     echo '
63     <div class="alert alert-danger alert-dismissible" align="
64     center">
65     <button type="button" class="close" data-dismiss="alert
66     " aria-hidden="true">&times;</button>
67     Usuario anyadido pero imagen no cargada, contacte con
68     el administrador comunic&aacute;ndole el error.
69     </div>
70     ' ;
71 }
72 // Funcion para eliminar el directorio del proveedor junto con todo su
73 contenido
74 function deleteDir($path){
75     if (is_dir($path) === true){
76         $files = array_diff(scandir($path), array('.', '..'));
77         foreach ($files as $file){
78             deleteDir(realpath($path) . '/' . $file);
79         }
80         return rmdir($path);
81     }
82     else if (is_file($path) === true){
83         return unlink($path);
84     }
85     return false;
86 }
87 // Funcion para modificar elementos de una entrada en la base de datos
88 function modificaIDEntrada($table, $tipo, $nombre, $medidas, $descripcion,
89 $con){
90     echo '
91     <div class="panel panel-default">
92     <div class="panel-heading">
93     Anyadir nueva entrada a la base de datos
94     </div>
95     <div class="panel-body">
96     <div class="row">
97     <div class="col-lg-10" style="margin-left:40px;">
98     <form enctype="multipart/form-data" role="form"
99     action="accesoAdmin.php?action=altaentrada
100     " method="post">
101     <p><h3><u>Datos de la entrada</u></h3></p>
102     <br>
103     <div class="form-group">
104     <label>Selecciona un proveedor:</label>
105     <select class="form-control" name="
106     table" required>
107     <option value="' . $table . '" selected="
108     selected">' . $table . '</option>';

```

```
102 $sqlListaProveedor = "SELECT * FROM '
103     proveedores' ORDER BY 'id' ";
104 $queryListaProveedor = mysql_query(
105     $sqlListaProveedor, $con);
106 while ($rowListaProveedor =
107     mysql_fetch_array(
108     $queryListaProveedor)){
109     $nombreProv = $rowListaProveedor [ '
110     nombre' ];
111     if ($nombreProv != $table) {
112         echo '<option value="' .
113             $nombreProv . '"">' .
114             $nombreProv . '</option></a>'
115             ;
116     }
117 }
118 echo '
119 </select>
120 </div>
121 <div class="form-group">
122     <label><strong style="color:red;">***
123     </strong>Identificador </label> (
124     indicar solo si existe la
125     referencia) <strong></strong>
126     <input class="form-control" type="text"
127     placeholder="Introduce un
128     identificador v&aacute;lido" name="
129     id">
130 </div>
131 <div class="form-group">
132     <label>Tipo:</label>
133     <input class="form-control" type="text"
134     value="' . $tipo . '" name="tipo"
135     required>
136 </div>
137 <div class="form-group">
138     <label>Nombre:</label>
139     <input class="form-control" type="text"
140     value="' . $nombre . '" name="nombre"
141     required>
142 </div>
143 <div class="form-group">
144     <label>Medidas:</label>
145     <input class="form-control" type="text"
146     value="' . $medidas . '" name="medidas"
147     required>
148 </div>
149 <div class="form-group">
150     <label>Descripci&oacute;n:</label>
151     <textarea class="form-control" rows="5"
152     name="descripcion" required>' .
153     $descripcion . '</textarea>
154 </div>
155 <div class="form-group">
```



```
179         <th style="text-align:center;">
180             Avatar</th>
181         <th style="text-align:center;">
182             Usuario</th>
183         <th style="text-align:center;">
184             Contraseña</th>
185     </tr>
186 </thead>
187 <tbody>
188     ;
189 // Diferenciamos entre el tipo de usuario
190 // dependiendo de los parametros recibidos
191 if($table=='Web'){
192     $sqlListaUsers = "SELECT * FROM 'usuarios_web'
193     ORDER BY 'id'";
194 }
195 else if($table=='App'){
196     $sqlListaUsers = "SELECT * FROM 'usuarios_app'
197     ORDER BY 'id'";
198 }
199 $queryListaUsers = mysql_query($sqlListaUsers,$con)
200 ;
201 while ($rowListaUsers = mysql_fetch_array(
202     $queryListaUsers)){
203     echo '<tr>';
204     echo '<td class="center">'. $rowListaUsers[ '
205     nombre' ]. '</td>';
206     echo '<td class="center">'. $rowListaUsers[ '
207     apellidos' ]. '</td>';
208     echo '<td class="center">'. $rowListaUsers[ '
209     cargo' ]. '</td>';
210     if ($rowListaUsers[ 'imagen' ]!=NULL){
211         echo '<td class="center"><a target=_blank
212         href="'. $rowListaUsers[ 'imagen' ]. '>'.
213         $rowListaUsers[ 'user' ]. '</a></td>';
214     }
215     else{
216         echo '<td class="center">--</td>';
217     }
218     echo '<td class="center">'. $rowListaUsers[ 'user
219     ' ]. '</td>';
220     echo '<td class="center"> ***** </td>';
221     echo "</tr>";
222 }
223 echo '</tbody>
224 </table>
225 </div>
226 <!-- /.table-responsive -->
227 </div>
228 <!-- /.panel-body -->
229 </div>
230 <!-- /.panel -->
231 <br><br>
232 </div>
```

```

219     ' ;
220 }
221
222 // Funcion para modificar la entrada de un nuevo usuario
223 function modificaAltaUser($nombre,$apellidos,$user,$pass,$cargo,
224     $errorNick,$errorCheckList){
225     echo '
226     <div class="panel panel-default">
227     <div class="panel-heading">
228         Anyadir nuevo usuario
229     </div>
230     <div class="panel-body">
231     <div class="row">
232     <div class="col-lg-10" style="margin-left:40px;">
233     <form enctype="multipart/form-data" role="form"
234     action="accesoAdmin.php?action=altauser"
235     method="post">
236     <p><h3><u>Datos del usuario</u></h3></p>
237     <br>' ;
238     // Comprobacion error checklist para el
239     tipo de usuario
240     if($errorCheckList){
241     echo '
242     <div class="form-group">
243     <label><strong style="color:red
244     ;">*** </strong>Seleccione el
245     tipo de usuario: </label>
246     <div class="checkbox">
247     <label>
248     <input type="checkbox" name
249     ="check_list[]" value="
250     web">Usuario Web
251     </label>
252     </div>
253     <div class="checkbox">
254     <label>
255     <input type="checkbox" name
256     ="check_list[]" value="
257     app">Usuario App
258     </label>
259     </div>
260     </div>
261     ' ;
262     }
263     else{
264     echo '
265     <div class="form-group">
266     <label>Seleccione el tipo de
267     usuario: </label>
268     ' ;
269     if(!in_array('web', $_POST['check_list'
270     ])){
271     echo '
272     <div class="checkbox">

```



```
261         <label>
262             <input type="checkbox" name
                ="check_list[]" value="
                web">Usuario Web
263         </label>
264     </div>
265     ';
266 }
267 else {
268     echo '
269     <div class="checkbox">
270         <label>
271             <input type="checkbox" name
                ="check_list[]" value="
                web" checked>Usuario
                Web
272         </label>
273     </div>
274     ';
275 }
276 if (!in_array('app', $_POST['check_list'
277 ])) {
278     echo '
279     <div class="checkbox">
280         <label>
281             <input type="checkbox" name
                ="check_list[]" value="
                app">Usuario App
282         </label>
283     </div>
284     ';
285 }
286 else {
287     echo '
288     <div class="checkbox">
289         <label>
290             <input type="checkbox" name
                ="check_list[]" value="
                app" checked>Usuario
                App
291         </label>
292     </div>
293     ';
294 }
295 echo '</div>';
296 // Fin comprobacion error checklist
297 echo '
298 <div class="form-group">
299     <label>Nombre:</label>
300     <input class="form-control" type="text"
        value="'. $nombre. '" name="nombre"
        required>
301 </div>
```

```
302 <div class="form-group">
303 <label>Apellidos:</label>
304 <input class="form-control" type="text"
      value="'. $apellidos. '" name="
      apellidos" required>
305 </div>
306 <div class="form-group">
307 <label>Cargo:</label>
308 <input class="form-control" type="text"
      value="'. $cargo. '" name="cargo"
      required>
309 </div>
310 <br>;
311 // Comprobacion error user duplicado
312 if ($errorNick){
313     echo '
314     <label><strong style="color:red;">***
      </strong>Nombre de usuario:</label>
315     <div class="form-group input-group">
316     <span class="input-group-addon">@</
      span>
317     <input type="text" class="form-
      control" value="'. $user. '" name
      ="user" required>
318     </div>
319     '
320 }
321 else {
322     echo '
323     <label>Nombre de usuario:</label>
324     <div class="form-group input-group">
325     <span class="input-group-addon">@</
      span>
326     <input type="text" class="form-
      control" value="'. $user. '" name
      ="user" required>
327     </div>
328     '
329 }
330 // Fin comprobacion error user duplicado
331 echo '
332 <div class="form-group">
333 <label>Contrasena:</label>
334 <input class="form-control" type="text"
      value="'. $pass. '" name="pass"
      required>
335 </div>
336 <br>
337 <hr>
338 <div align="right">
339 <button type="reset" class="btn btn-
      default">Borrar datos</button>
340 <input type="submit" class="btn btn-
      default" style="background-color:
```



```

391         $row[$j] = ereg_replace("\n", "\\n", $row[$j]);
392         if (isset($row[$j])) { $return.= "'".$row[$j].'
393             "' ; } else { $return.= '""'; }
394         if ($j < ($num_fields-1)) { $return.= ','; }
395     }
396     $return.= ");\n";
397 }
398 $return.="\n\n\n";
399 }
400 // guardamos el fichero sql
401 $handle = fopen($folder.'/dbBackup_'.$reg_date.'_'.$reg_time.'.
402     sql', 'w+');
403 fwrite($handle, $return);
404 fclose($handle);
405 return true;
406 }
407 catch(Exception $e){
408     return false;
409 }
410 }
411 // Funcion para comprimir un directorio en formato zip
412 function planosZipBackup($source, $destination){
413     try{
414         if (!extension_loaded('zip') || !file_exists($source)) {
415             return false;
416         }
417         $zip = new ZipArchive();
418         if (!$zip->open($destination, ZIPARCHIVE::CREATE)) {
419             return false;
420         }
421         $source = str_replace('\\', '/', realpath($source));
422         if (is_dir($source) === true){
423             $files = new RecursiveIteratorIterator(new
424                 RecursiveDirectoryIterator($source),
425                 RecursiveIteratorIterator::SELF_FIRST);
426             foreach ($files as $file){
427                 $file = str_replace('\\', '/', $file);
428                 // Ignore "." and ".." folders
429                 if( in_array(substr($file, strrpos($file, '/')+1),
430                     array('.', '..')) )
431                     continue;
432                 $file = realpath($file);
433                 if (is_dir($file) === true){
434                     $zip->addEmptyDir(str_replace($source . '/', '',
435                         $file . '/'));
436                 }
437                 else if (is_file($file) === true){
438                     $zip->addFromString(str_replace($source . '/', '',
439                         $file), file_get_contents($file));
440                 }
441             }
442         }
443     }
444 }

```

```
438     else if (is_file($source) === true){
439         $zip->addFromString(basename($source), file_get_contents(
440             $source));
441     }
442     $zip->close();
443     return true;
444 }
445 catch(Exception $e){
446     return false;
447 }
448
449 // Funcion para eliminar un directorio
450 function deleteFolder($dir){
451     try{
452         $it = new RecursiveDirectoryIterator($dir,
453             RecursiveDirectoryIterator::SKIP_DOTS);
454         $files = new RecursiveIteratorIterator($it,
455             RecursiveIteratorIterator::CHILD_FIRST);
456         foreach ($files as $file) {
457             if($file->isDir()){
458                 rmdir($file->getRealPath());
459             }
460             else{
461                 unlink($file->getRealPath());
462             }
463         }
464         rmdir($dir);
465         return true;
466     }
467     catch(Exception $e){
468         return false;
469     }
470 }
471
472 // Funcion para crear un backup completo del sistema
473 function doBackup($reg_date,$reg_time){
474     $folder = 'database/sqlBackups/'. $reg_date.'_'.$reg_time;
475     if(!file_exists($folder)){
476         mkdir($folder,0777,true);
477     }
478     $retValBackUpSql = backupDB($folder,$reg_date,$reg_time);
479     $retValBackUpPlanos = planosZipBackup('database/images', $folder.'/
480         dbBackup_images_'.$reg_date.'_'.$reg_time.'.zip');
481     if($retValBackUpSql&&$retValBackUpPlanos){
482         // Creamos un fichero comprimido con el backup completo
483         $totalBackup = planosZipBackup($folder,$folder.'.zip');
484         if($totalBackup){
485             deleteFolder($folder);
486             return true;
487         }
488     }
489     else{
490         deleteFolder($folder);
491         return false;
492     }
493 }
```

```

488     }
489   }
490   else {
491     if (!$retValBackUpSql&& !$retValBackUpPlanos) {
492       return false;
493     }
494     if (!$retValBackUpSql) {
495       unlink($folder . '/dbBackup_images_' . $reg_date . '_' . $reg_time .
         '.zip');
496     }
497     if (!$retValBackUpPlanos) {
498       unlink($folder . '/dbBackup_' . $reg_date . '_' . $reg_time . '.sql')
         ;
499     }
500     return false;
501   }
502 }
503
504
505 ?>

```

A.6 Página de inicio

El siguiente código pertenece a la página de inicio del portal de administración telemática de nuestro servidor, desde la cual el usuario puede realizar gestiones tales como la gestión de usuarios de la aplicación móvil, la gestión de la base de datos del sistema, o la generación de copias de seguridad de la base de datos.

```

1
2 #####
3 ##### SISERVER accesoAdmin.php #####
4 #####
5 # Pagina home del servidor web
6 # @auth alroico@inf.upv.es
7 # @date 12/06/2017
8
9 <?php
10
11 // Incluimos el fichero auxiliar para la conexion a la BD
12 include_once "conexion.php";
13 // Incluimos el fichero auxiliar con las funciones del sistema
14 include_once "functions.php";
15 // Iniciamos la sesion de usuario
16 session_start();
17
18 // Control de sesion para asegurar que el usuario se loguea
19 $firmado = $_SESSION['firmado'];
20 if ($firmado==0){
21     header('Location: index.php');

```

```
22 }
23
24 // Damos forma al contenido HTML
25 echo '
26 <!DOCTYPE html>
27 <html lang="es">
28
29 <head>
30
31     <!--
32         Vista movil desactivada.
33         <meta name="viewport" content="width=device-width, initial-scale
34             =1">
35     -->
36
37     <meta charset="utf-8">
38     <meta http-equiv="X-UA-Compatible" content="IE=edge">
39     <meta name="description" content="">
40     <meta name="author" content="">
41
42     <title>SI Server EIR – Servidor de administraci&oacute;n</title >
43
44     <!-- Bootstrap Core CSS -->
45     <link href="bower_components/bootstrap/dist/css/bootstrap.min.css"
46         rel="stylesheet">
47
48     <!-- MetisMenu CSS -->
49     <link href="bower_components/metisMenu/dist/metisMenu.min.css" rel
50         ="stylesheet">
51
52     <!-- Timeline CSS -->
53     <link href="dist/css/timeline.css" rel="stylesheet">
54
55     <!-- Custom CSS -->
56     <link href="dist/css/sb-admin-2.css" rel="stylesheet">
57
58     <!-- Morris Charts CSS -->
59     <link href="bower_components/morrisjs/morris.css" rel="stylesheet">
60
61     <!-- Custom Fonts -->
62     <link href="bower_components/font-awesome/css/font-awesome.min.css"
63         rel="stylesheet" type="text/css">
64
65     <!-- DataTables CSS -->
66     <link href="bower_components/datatables-plugins/integration/
67         bootstrap/3/dataTables.bootstrap.css" rel="stylesheet">
68
69     <!-- DataTables Responsive CSS -->
70     <link href="bower_components/datatables-responsive/css/dataTables.
71         responsive.css" rel="stylesheet">
72
73     <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and
74         media queries -->
```

```

68 <!-- WARNING: Respond.js doesnt work if you view the page via file
    :// -->
69 <!--[if lt IE 9]>
70 <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/
    html5shiv.js"></script>
71 <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/
    respond.min.js"></script>
72 <![endif]-->
73
74 <link rel="shortcut icon" href="images/icon2.ico"/>
75 </head>
76
77 <body>
78
79 <div id="wrapper">
80
81 <!-- Navigation -->
82 <nav class="navbar navbar-default navbar-static-top" role="
    navigation" style="margin-bottom: 0">
83 <div class="navbar-header" style="margin-left:10px;">
84 <button type="button" class="navbar-toggle" data-toggle
    ="collapse" data-target=".navbar-collapse">
85 <span class="sr-only">Toggle navigation</span>
86 <span class="icon-bar"></span>
87 <span class="icon-bar"></span>
88 <span class="icon-bar"></span>
89 </button>
90 <a class="navbar-brand" href="accesoAdmin.php?action=
    home"><strong>SI Server EIR – Servidor de
    administraci&oacute;n</strong></a>
91 </div>
92 <!-- /.navbar-header -->
93
94 <ul class="nav navbar-top-links navbar-right" align="center
    " style="margin-right:50px;">';
95
96 // Consultamos e imprimimos los datos del usuario
97 $user = $_SESSION['userid'];
98 $sql = "SELECT nombre, apellidos, cargo, imagen FROM 'usuarios_web'
    WHERE user='$user'";
99 $query = mysql_query($sql);
100 $row = mysql_fetch_row($query);
101 $nombre = $row[0];
102 $apellidos = $row[1];
103 $cargo = $row[2];
104 $imagen = $row[3];
105 if ($imagen==NULL){
106 
107 }
108
109 echo '<strong>Usuario: [ </strong>';
110 echo $apellidos; echo ", "; echo $nombre;
111 echo '<strong> ]</strong>';
112

```



```

113     echo '
114         <li class="dropdown">
115             <a class="dropdown-toggle" data-toggle="dropdown"
116                 href="#">
117                 <i class="fa fa-user fa-fw"></i> <i class="fa
118                 fa-caret-down"></i>
119             </a>
120             <ul class="dropdown-menu dropdown-user">
121                 <li><a href="specialLogout.php?f=s"><i class="
122                 fa fa-power-off fa-fw"></i> Apagar
123                 Dispositivo</a></li>
124                 <li><a href="specialLogout.php?f=r"><i class="
125                 fa fa-repeat fa-fw"></i> Reiniciar
126                 Dispositivo</a></li>
127                 <li class="divider"></li>
128                 <li><a href="accesoAdmin.php?action=home"><i
129                 class="fa fa-user fa-fw"></i> Perfil de
130                 usuario</a></li>
131                 <li><a href="logout.php"><i class="fa fa-sign-
132                 out fa-fw"></i> Cerrar sesi&oacute;n</a></
133                 li>
134             </ul>
135             <!-- /.dropdown-user -->
136         </li>
137         <!-- /.dropdown -->
138     </ul>
139     <!-- /.navbar-top-links -->
140
141     <div class="navbar-default sidebar" role="navigation">
142         <div class="sidebar-nav navbar-collapse">
143             <ul class="nav" id="side-menu">
144                 <li align="center">
145                     <a href="accesoAdmin.php?action=home"></a>
149                 </li>
150                 <li>
151                     <a href="#"><i class="fa fa-book fa-fw"></i>
152                     <span class="fa arrow"></span>
153                     </a>
154                     <ul class="nav navbar-second-level">
155                         <li>'
156                             $_SESSION['ses'] = True;
157                             echo '<a href="accesoAdmin.php?
158                             action=listaclientes">Registro
159                             de Clientes</a>
160                         </li>
161                         <li>'
162                             $_SESSION['ses'] = True;
163                             echo '<a href="accesoAdmin.php?
164                             action=altacliente">Alta Nuevo
165                             cliente</a>
166                         </li>

```

```

148         <li>';
149         $_SESSION['ses'] = True;
150         echo '<a href="accesoAdmin.php?
151             action=eliminarcliente"></i>
152             Eliminar cliente</a>
153         </li>
154     </ul>
155     <!-- /.nav-second-level -->
156 </li>
157 <li>
158     <a href="#"><i class="fa fa-database fa-fw
159         "></i> Base de datos<span class="fa
160         arrow"></span></a>
161     <ul class="nav nav-second-level">
162         <li>';
163         $_SESSION['ses'] = True;
164         echo '<a href="accesoAdmin.php?
165             action=listaentrada">Listar
166             entradas</a>
167     </li>
168     <li>';
169     $_SESSION['ses'] = True;
170     echo '<a href="accesoAdmin.php?
171         action=listaentrcliente">Listar
172         entradas por cliente</a>
173     </li>
174     <li>';
175     $_SESSION['ses'] = True;
176     echo '<a href="accesoAdmin.php?
177         action=altaentrada">Alta nueva
178         entrada</a>
179     </li>
180     <li>';
181     $_SESSION['ses'] = True;
182     echo '<a href="accesoAdmin.php?
183         action=modificaentrada"></i>
184         Modificar entrada</a>
185     </li>
186     <li>';
187     $_SESSION['ses'] = True;
188     echo '<a href="accesoAdmin.php?
189         action=eliminaentrada"></i>
190         Eliminar entrada</a>
191     </li>
192     </ul>
193     <!-- /.nav-second-level -->
194 </li>
195 <li>
196     <a href="#"><i class="fa fa-download fa-fw
197         "></i> Copias de seguridad<span class="
198         fa arrow"></span></a>
199     <ul class="nav nav-second-level">
200         <li>';
201         $_SESSION['ses'] = True;

```

```
186         echo ' <a href="accesoAdmin.php?
           action=listacopia">Listar
           copias de seguridad</a>
187     </li>
188     <li>';
189         $_SESSION['ses'] = True;
190         echo ' <a href="accesoAdmin.php?
           action=altacopia">Crear copia
           de seguridad</a>
191     </li>
192     <li>';
193         $_SESSION['ses'] = True;
194         echo ' <a href="accesoAdmin.php?
           action=eliminacopia"></i>
           Eliminar copia de seguridad</a>
195     </li>
196 </ul>
197 <!-- /.nav-second-level -->
198 </li>
199 <li>';
200     $_SESSION['ses'] = True;
201     echo ' <a href="accesoAdmin.php?action=
           altauser"><i class="fa fa-users fa-fw
           "></i> Alta nuevo usuario</a>
202 </li>
203 <li>
204     <a href="#"><i class="fa fa-laptop fa-fw
           "></i> Gesti&oacute;n de usuarios Web<
           span class="fa arrow"></span></a>
205     <ul class="nav nav-second-level">
206         <li>';
207             $_SESSION['ses'] = True;
208             echo ' <a href="accesoAdmin.php?
                 action=listauserweb">Listado de
                 usuarios</a>
209         </li>
210         <li>';
211             $_SESSION['ses'] = True;
212             echo ' <a href="accesoAdmin.php?
                 action=modificauserweb">
                 Modificar usuario</a>
213         </li>
214         <li>';
215             $_SESSION['ses'] = True;
216             echo ' <a href="accesoAdmin.php?
                 action=eliminausersweb">
                 Eliminar usuario</a>
217         </li>
218     </ul>
219     <!-- /.nav-second-level -->
220 </li>
221 <li>
```

```

222         <a href="#"><i class="fa fa-mobile fa-fw
223             "></i> Gesti&ocute;n de usuarios App<
224             span class="fa arrow"></span></a>
225         <ul class="nav nav-second-level">
226             <li>';
227                 $_SESSION['ses'] = True;
228                 echo '<a href="accesoAdmin.php?
229                     action=listauserapp">Listado de
230                     usuarios</a>
231             </li>
232             <li>';
233                 $_SESSION['ses'] = True;
234                 echo '<a href="accesoAdmin.php?
235                     action=modificauserapp">
236                     Modificar usuario</a>
237             </li>
238             <li>';
239                 $_SESSION['ses'] = True;
240                 echo '<a href="accesoAdmin.php?
241                     action=eliminauserapp">Eliminar
242                     usuario</a>
243             </li>
244         </ul>
245         <!-- /.nav-second-level -->
246     </li>
247 </ul>
248 </div>
249 <!-- /.sidebar-collapse -->
250 </div>
251 <!-- /.navbar-static-side -->
252 </nav>
253
254 <!-- Page Content -->
255 <div id="page-wrapper">
256     <div class="container-fluid">
257         <div class="row">
258             <div class="col-lg-10">';
259
260 $ses = $_SESSION['ses'];
261
262 if ($ses!=False){
263     $action=$_GET['action'];
264
265     //Diferenciamos el contenido a mostrar dependiendo de la seleccion
266     //del usuario
267
268     //////////////////////////////////////
269     // Inicio de las acciones de usuario //
270     //////////////////////////////////////
271
272     if ($action == 'home'){
273         echo '<h1 class="page-header">Perfil de Usuario</h1>
274             <br><br>
275             <div class="col-lg-10">

```

```

267     <div class="panel panel-default">
268         <div class="panel-heading">
269             Datos de usuario
270         </div>
271         <!-- /.panel-heading -->
272         <div class="panel-body">
273             <div class="col-lg-10" align="left">
274                 <div class="table-responsive">
275                     <table class="table table-striped">
276                         <tbody>
277                             <br>
278                             <tr>
279                                 <td><strong>Usuario:</strong></td>
280                                 <td>'<echo $user;
281                                 <echo '
282                                 </td>
283                             </tr>
284                             <tr>
285                                 <td><strong>Nombre:</strong></td>
286                                 <td>'<echo $nombre;
287                                 <echo '
288                                 </td>
289                             </tr>
290                             <tr>
291                                 <td><strong>Apellidos:</strong></td>
292                                 <td>'<echo $apellidos;
293                                 <echo '
294                                 </td>
295                             </tr><tr>
296                                 <td><strong>Cargo:</strong></td>
297                                 <td>'<echo $cargo;
298                                 <echo '
299                                 </td>
300                             </tr>
301                         </tbody>
302                     </table>
303                 </div>
304             </div>
305             <br>
308             </div>
309         <!-- /.panel-body -->
310     </div>
311     </div>
312 }
313
314 // Listado de clientes

```

```

315 else if($action == 'listaclientes'){
316     echo '<h1 class="page-header">Registro de Clientes</h1>';
317     echo ' <br><br>
318         <div class="col-lg-12">
319             <div class="panel panel-default">
320                 <div class="panel-heading">
321                     Clientes registrados
322                 </div>
323                 <!-- /.panel-heading -->
324                 <div class="panel-body">
325                     <br>
326                     <div class="dataTable_wrapper table-
327                         responsive">
328                         <table class="table table-striped table
329                             -bordered table-hover" id="
330                             dataTables-example">
331                             <thead>
332                                 <tr>
333                                     <th>ID</th>
334                                     <th>Cliente</th>
335                                     <th>NÂ° de entradas</th>
336                                     <th>Fecha de registro</th>
337                                 </tr>
338                             </thead>
339                             <tbody>
340                                 ';
341                                 $sqlListaCliente = "SELECT * FROM 'clientes
342                                     ' ORDER BY 'id' ";
343                                 $queryListaCliente = mysql_query(
344                                     $sqlListaCliente,$con);
345                                 while ($rowListaCliente = mysql_fetch_array
346                                     ($queryListaCliente)){
347                                     $sqlNumRows = 'SELECT COUNT(*) FROM ' .
348                                         $rowListaCliente['nombre'];
349                                     $retvalNumRows = mysql_query(
350                                         $sqlNumRows,$con);
351                                     $rowNumRows = mysql_fetch_row(
352                                         $retvalNumRows);
353                                     $numRows = $rowNumRows[0];
354                                     echo '<tr>';
355                                     echo '<td class="center">'.
356                                         $rowListaCliente['id'].'</td>';
357                                     echo '<td class="center">'.
358                                         $rowListaCliente['nombre'].'</td>';
359                                     echo '<td class="center">'.$numRows.'</
360                                         td>';
361                                     echo '<td class="center">'.
362                                         $rowListaCliente['reg_date'].'</td>
363                                     '
364                                     echo "</tr>";
365                                 }
366                             echo '</tbody>
367                             </table>
368                         </div>
369                     </div>
370                 </div>
371             </div>
372         </div>
373     </div>
374 }

```

```

355         <!-- /.table-responsive -->
356     </div>
357     <!-- /.panel-body -->
358 </div>
359 <!-- /.panel -->
360 <br>
361 </div>
362 ';
363 }
364
365 // Alta nuevo Cliente
366 else if ($action == 'altacliente'){
367     echo '<h1 class="page-header">Alta Nuevo Cliente</h1>
368         <br><br>
369         <div class="col-lg-12">';
370
371     if (isset($_POST['nombre'])&&$_POST['nombre']!= ''){
372         $nombre = $_POST['nombre'];
373
374         // Comprobacion de seguridad para no repetir nombres de
375         // clientes
376         $sqlCheckTablename = "SELECT * FROM clientes WHERE nombre='
377         ".$nombre."'";
378         $queryCheckTablename = mysql_query($sqlCheckTablename, $con)
379         ;
380         $retVal = mysql_num_rows($queryCheckTablename);
381         if ($retVal!=0){
382             // Ya existen entradas con ese nombre
383             echo '
384                 <div class="alert alert-danger alert-dismissible"
385                 align="center">
386                 <button type="button" class="close" data-
387                 dismiss="alert" aria-hidden="true">&times;
388                 </button>
389                 Ya existe un cliente con el nombre indicado,
390                 por favor introduzca un nombre distinto.
391             </div>
392             ';
393         }
394         else{
395             // Limpiamos el string de caracteres especiales
396             $nombre = cleanString($nombre);
397             $datetime = date("Y-m-d H:i:s");
398             // No existe un cliente con el nombre indicado. Lo
399             // creamos.
400             $sqlAltaCliente = "INSERT INTO clientes (nombre,
401                 reg_date) VALUES ('$nombre', '$datetime')";
402             $queryAltaCliente = mysql_query($sqlAltaCliente, $con);
403             if ($queryAltaCliente){
404                 $sqlNewTable = 'CREATE TABLE '.$nombre.' (
405                     id VARCHAR(30) PRIMARY KEY,
406                     tipo VARCHAR(30) NOT NULL,
407                     nombre VARCHAR(30) NOT NULL,
408                     medidas VARCHAR(30) NOT NULL,

```

```

400         descripcion VARCHAR(10000) NOT NULL
401         ,
402         imagen1 VARCHAR(1000) ,
403         imagen2 VARCHAR(1000) ,
404         imagen3 VARCHAR(1000) ,
405         reg_date DATETIME
406     );
407 $queryNewTable = mysql_query($sqlNewTable,$con);
408 if($queryNewTable){
409     // Creamos una carpeta fisica para almacenar
410     // los planos de las entradas
411     $folder = 'database/images/'. $nombre;
412     if(!file_exists($folder)){
413         mkdir($folder,0777,true);
414     }
415     echo '
416     <div class="alert alert-success alert-
417     dismissable" align="center">
418     <button type="button" class="close" data-
419     dismiss="alert" aria-hidden="true">&
420     times;</button>
421     Cliente anyadido correctamente.
422     </div>
423     ';
424 }
425 else{
426     $sqlRedo = "DELETE FROM 'clientes' WHERE nombre
427     ='". $nombre. "'";
428     $retvalRedo= mysql_query($sqlRedo,$con);
429     echo '
430     <div class="alert alert-danger alert-
431     dismissable" align="center">
432     <button type="button" class="close" data-
433     dismiss="alert" aria-hidden="true">&times
434     ;</button>
435     Error en la base de datos. Tome nota del error
436     y contacte con el administrador del sistema
437     . Algun valor introducido en los datos no
438     es correcto: '. mysql_error();
439     echo '</div>
440     ';
441 }
442 }
443 else{
444     echo '
445     <div class="alert alert-danger alert-dismissable"
446     align="center">
447     <button type="button" class="close" data-dismiss="
448     alert" aria-hidden="true">&times;</button>
449     Error en la base de datos. Tome nota del error y
450     contacte con el administrador del sistema.

```



```

        Algun valor introducido en los datos no es
        correcto: '. mysql_error();
438         echo '</div>
439         ';
440     }
441 }
442 }
443 echo '
444     <div class="panel panel-default">
445         <div class="panel-heading">
446             Anyadir nuevo cliente
447         </div>
448         <div class="panel-body">
449             <div class="row">
450                 <div class="col-lg-10" style="margin-left:40px
451                     ;">
452                     <form role="form" action="accesoAdmin.php?
453                         action=altacliente" method="post">
454                         <p><h3><u>Datos del cliente </u></h3></p>
455                         <br>
456                         <div class="form-group">
457                             <label>Nombre:</label>
458                             <input class="form-control" type="
459                                 text" placeholder="Introduce el
460                                 nombre del cliente" name="
461                                 nombre" required>
462
463                             </div>
464                             <p>No podr&aacute; modificar el nombre
465                                 del cliente posteriormente</p>
466                             <br>
467                             <hr>
468                             <div class="form-group" align="right">
469                                 <button type="reset" class="btn btn
470                                     -default">Borrar datos</button>
471                                 <input type="submit" class="btn btn
472                                     -default" style="background-
473                                     color:green; color:white;"
474                                     value="Anyadir cliente">
475
476                             </div>
477                             <br><br>
478                         </form>
479                     </div>
480                     <!-- /.col-lg-10 (nested) -->
481                     <br><br>
482                 </div>
483                 <!-- /.row (nested) -->
484             </div>
485             <!-- /.panel-body -->
486         </div>
487         <!-- /.panel -->
488     </div>
489     ';
490 }

```

```

479
480 // Eliminar Cliente
481 else if ($action == 'eliminarcliente'){
482     echo '<h1 class="page-header">Eliminar Cliente</h1>
483         <br><br>
484         <div class="col-lg-12">';
485     if (isset($_POST['selectedCliente'])&&$_POST['selectedCliente']
486         != ''&&!isset($_POST['okDrop'])) {
487         // Confirmacion de seguridad
488         $selectedCliente = $_POST['selectedCliente'];
489         $sqlTableName = 'SELECT nombre FROM clientes WHERE id=' .
490             $selectedCliente;
491         $retvalTableName = mysql_query($sqlTableName, $con);
492         $rowTableName = mysql_fetch_row($retvalTableName);
493         $tableName = $rowTableName[0];
494         $sqlDataTable = 'SELECT COUNT(*) FROM ' . $tableName;
495         $retvalDataTable = mysql_query($sqlDataTable, $con);
496         $rowDataTable = mysql_fetch_row($retvalDataTable);
497         $numRows = $rowDataTable[0];
498         echo '
499         <div class="panel panel-default">
500             <div class="panel-heading">
501                 Confirmaci&ocaron de seguridad
502             </div>
503             <!-- /.panel-heading -->
504             <div class="panel-body">
505                 <br>
506                 <div class="col-lg-10" style="margin-left:40px
507                     ;">
508                     <label>Selecciona un cliente:</label>
509                     <form action="accesoAdmin.php?action=
510                         eliminarcliente" method="post">
511                         <select class="form-control" name="
512                             selectedCliente" required>
513                             <option value="' . $selectedCliente . '
514                                 " selected="selected">' .
515                                 $tableName . '</option>
516                         </select>
517                         <input type="hidden" name="tableName"
518                             value="' . $tableName . '">
519                         <br>
520                         <div class="form-group">
521                             <label>N&uacute;mero de entradas en
522                                 la base de datos del cliente
523                             </label>
524                             <input class="form-control" type="
525                                 text" name="okDrop" value="' .
526                                 $numRows . '" required>
527                         </div>
528                         <hr>
529                         <div class="form-group" align="right">
530                             <p><strong>Esta accion no tiene
531                                 vuelta atras</strong></p>

```

```

519         <input type="submit" class="btn btn
                    -default" style="background-
                    color:red; color:white;" value
                    ="Eliminar cliente">
520     </div>
521 </form>
522 </div>
523 </div>
524 <!-- /.table-responsive -->
525 </div>
526 <!-- /.panel-body -->
527 </div>
528 <!-- /.panel -->
529 <hr>
530 '
531 }
532 else if (isset($_POST['selectedCliente'])&&$_POST['
selectedCliente']!= ''&&isset($_POST['okDrop'])&&isset(
$_POST['tableName'])) {
533     $selectedCliente = $_POST['selectedCliente'];
534     $tableName = $_POST['tableName'];
535     $sqlDropCliente = "DELETE FROM 'clientes' WHERE id='".
        $selectedCliente."'";
536     $retvalDropCliente = mysql_query($sqlDropCliente,$con);
537     if($retvalDropCliente) {
538         $sqlDropTable = 'DROP TABLE '.$_POST['tableName'];
539         $retvalDropTable= mysql_query($sqlDropTable,$con);
540         if($retvalDropTable) {
541             // Eliminamos el directorio con los planos del
                    Cliente
542             $folderPath = "database/images/".$tableName;
543             deleteDir($folderPath);
544             echo '
545 <div class="alert alert-success alert-dismissable"
                    align="center">
546 <button type="button" class="close" data-dismiss="
                    alert" aria-hidden="true">&times;</button>
                    Cliente eliminado correctamente.
547 </div>
548 '
549 ;
550         }
551     } else {
552         echo '
553 <div class="alert alert-danger alert-dismissable"
                    align="center">
554 <button type="button" class="close" data-dismiss="
                    alert" aria-hidden="true">&times;</button>
                    Error en la base de datos. Cliente eliminado pero
                    tabla no removida. Tome nota del error y
                    contacte con el desarrollador de la aplicaci&
                    oacute;n.</div>
556 '
557 ;
558     }

```

```

559     else {
560         echo '
561         <div class="alert alert-danger alert-dismissible" align
562             ="center">
563         <button type="button" class="close" data-dismiss="alert
564             " aria-hidden="true">&times;</button>
565         Error en la base de datos. Tome nota del error y
566             contacte con el administrador del sistema. Algun
567             valor introducido en los datos no es correcto: '.
568         mysql_error();
569         echo '</div>
570         ';
571     }
572 }
573 else {
574     echo '
575     <div class="panel panel-default">
576     <div class="panel-heading">
577         Listado de clientes
578     </div>
579     <!-- /.panel-heading -->
580     <div class="panel-body">
581         <br>
582         <div class="col-lg-10" style="margin-left:40px
583             ;">
584             <label>Selecciona un cliente:</label>
585             <form action="accesoAdmin.php?action=
586                 eliminarcliente" method="post">
587             <select class="form-control" name="
588                 selectedCliente" required>
589             <option value="" selected="selected"></
590                 option>';
591             $sqlListaCliente = "SELECT * FROM 'clientes
592                 ' ORDER BY 'id' ";
593             $queryListaCliente = mysql_query(
594                 $sqlListaCliente,$con);
595             while ($rowListaCliente =
596                 mysql_fetch_array($queryListaCliente)){
597                 $nombre = $rowListaCliente ['nombre'];
598                 $id = $rowListaCliente ['id'];
599                 echo '<option value="'. $id. '"">'.
600                     $nombre. '</option></a>';
601             }
602             echo '
603             </select>
604             <br>
605             <hr>
606             <div class="form-group" align="right">
607                 <input type="submit" class="btn btn-
608                     default" style="background-color:
609                     red; color:white;" value="Eliminar
610                     cliente">
611             </div>
612             </form>

```

```

597         </div>
598     </div>
599     <!-- /.table-responsive -->
600 </div>
601 <!-- /.panel-body -->
602 </div>
603 <!-- /.panel -->
604 <hr>
605 ';
606 }
607 }
608
609 // Listar entradas de la base de datos
610 else if($action == 'listaentrada'){
611     echo '<h1 class="page-header">Listado de entradas en la base de
        datos</h1>
612     <br>
613     <div class="col-lg-12">
614         <br>
615         <div class="panel panel-default">
616             <div class="panel-heading">
617                 Entradas en la base de datos
618             </div>
619             <!-- /.panel-heading -->
620             <div class="panel-body">
621                 <div class="dataTable_wrapper table-responsive
622                     ">
623                     <table class="table table-striped table-
624                         bordered table-hover" id="dataTables-
625                             example">
626                         <thead>
627                             <tr>
628                                 <th>Cliente </th>
629                                 <th>ID</th>
630                                 <th>Tipo</th>
631                                 <th>Nombre</th>
632                                 <th>Medidas</th>
633                                 <th>Descripci&oacute;n</th>
634                                 <th>Imagen 1</th>
635                                 <th>Imagen 2</th>
636                                 <th>Imagen 3</th>
637                                 <th>Fecha de registro</th>
638                             </tr>
639                         </thead>
640                         <tbody>
641                             <tr>
642                                 <td>

```

```
643 $sqlListaEntradas = "SELECT * FROM
        ". $tabla. " ORDER BY id";
644 $queryListaEntradas= mysql_query(
        $sqlListaEntradas , $con);
645 while ( $rowListaEntradas =
        mysql_fetch_array(
        $queryListaEntradas)) {
646     echo '<tr>';
647     echo '<td class="center">'.
        $tabla. '</td>';
648     echo '<td class="center">'.
        $rowListaEntradas[ 'id' ]. '</
        td>';
649     echo '<td class="center">'.
        $rowListaEntradas[ 'tipo' ]. '
        </td>';
650     echo '<td class="center">'.
        $rowListaEntradas[ 'nombre'
        ]. '</td>';
651     echo '<td class="center">'.
        $rowListaEntradas[ 'medidas'
        ]. '</td>';
652     echo '<td class="center">'.
        $rowListaEntradas[ '
        descripcion' ]. '</td>';
653     if ( $rowListaEntradas[ 'imagen1'
        ] != NULL ) {
654         echo '<td class="center
        "><a target=_blank
        href="'.
        $rowListaEntradas[ '
        imagen1' ]. '">'.
        $rowListaEntradas[ '
        nombre' ]. ' - 1</a
        ></td>';
655     }
656     else {
657         echo '<td class="center
        ">-</td>';
658     }
659     if ( $rowListaEntradas[ 'imagen2'
        ] != NULL ) {
660         echo '<td class="center
        "><a target=_blank
        href="'.
        $rowListaEntradas[ '
        imagen2' ]. '">'.
        $rowListaEntradas[ '
        nombre' ]. ' - 2</a
        ></td>';
661     }
662     else {
663         echo '<td class="center
        ">-</td>';
```

```

664     }
665     if ($rowListaEntradas['imagen3'
666         ]!=NULL){
667         echo '<td class="center
668             "><a target=_blank
669                 href="' .
670                 $rowListaEntradas['
671                 imagen3' ]. '>' .
672                 $rowListaEntradas['
673                 nombre' ]. ' - 3</a
674                 ></td>';
675     }
676     else {
677         echo '<td class="center
678             ">-</td>';
679     }
680     echo '<td class="center">' .
681         $rowListaEntradas['reg_date
682         ']. '</td>';
683     echo "</tr>";
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }

```

```

// Listar entradas de la base de datos por cliente
else if ($action == 'listaentrcliente') {
    echo '<h1 class="page-header">Listado de entradas por cliente
    en la base de datos</h1>
    <br>
    <div class="col-lg-12">
    <br>
    <div class="panel panel-default">
    <div class="panel-heading">
    Entradas en la base de datos
    </div>
    <!-- /.panel-heading -->
    <div class="panel-body">

```

```

704 <div class="dataTable_wrapper table-
705 responsive">
706 <table class="table table-striped table
707 -bordered table-hover" id="
708 dataTableExample">
709 <thead>
710 <tr>
711 <th>Cliente</th>
712 <th>ID</th>
713 <th>Tipo</th>
714 <th>Nombre</th>
715 <th>Medidas</th>
716 <th>Descripci&oacute;n</th>
717 <th>Imagen 1</th>
718 <th>Imagen 2</th>
719 <th>Imagen 3</th>
720 <th>Fecha de registro</th>
721 </tr>
722 </thead>
723 <tbody>
724 <tr>
725 <td class="center">'.
726 $cliente.'</td>'.
727 <td class="center">'.
728 $rowListaEntradas['id'].'</
729 td>'.
730 <td class="center">'.
731 $rowListaEntradas['tipo'].'
732 </td>'.
733 <td class="center">'.
734 $rowListaEntradas['nombre'
735 ].'</td>'.
736 <td class="center">'.
737 $rowListaEntradas['medidas'
738 ].'</td>'.
739 <td class="center">'.
740 $rowListaEntradas['
741 descripcion'].'</td>'.
742
743 <td class="center">'.
744 <a href="#">'.
745 $rowListaEntradas['
746 imagen1' ].'>'.

```



```

735         $rowListaEntradas[ '
736         nombre' ]. ' - 1</a
737         ></td>';
738     }
739     else {
740         echo '<td class="center
741         ">-</td>';
742     }
743     if ($rowListaEntradas[ 'imagen2'
744     ]!=NULL) {
745         echo '<td class="center
746         "><a target=_blank
747         href="'.
748         $rowListaEntradas[ '
749         imagen2' ]. '">'.
750         $rowListaEntradas[ '
751         nombre' ]. ' - 2</a
752         ></td>';
753     }
754     else {
755         echo '<td class="center
756         ">-</td>';
757     }
758     if ($rowListaEntradas[ 'imagen3'
759     ]!=NULL) {
760         echo '<td class="center
761         "><a target=_blank
762         href="'.
763         $rowListaEntradas[ '
764         imagen3' ]. '">'.
765         $rowListaEntradas[ '
766         nombre' ]. ' - 3</a
767         ></td>';
768     }
769     else {
770         echo '<td class="center
771         ">-</td>';
772     }
773     echo '<td class="center">'.
774     $rowListaEntradas[ 'reg_date
775     ' ]. '</td>';
776     echo "</tr>";
777 }
778 echo '
779 </tbody>
780 </table>
781 </div>
782 <!-- /.table-responsive -->
783 </div>
784 <!-- /.panel-body -->
785 </div>';
786 }
787 else {
788     // Seleccionamos el Cliente
```

```

765     echo '
766         <div class="panel panel-default">
767             <div class="panel-heading">
768                 Listado de clientes
769             </div>
770             <!-- /.panel-heading -->
771             <div class="panel-body">
772                 <br>
773                 <div class="col-lg-10" style="margin-
774                     left:40px;">
775                     <label>Selecciona un cliente:</
776                         label>
777                     <form action="accesoAdmin.php?
778                         action=listaentrcliente" method
779                         ="post">
780                     <select class="form-control" name="
781                         selectedCliente" required>
782                     <option value="" selected="selected
783                         "></option>' ;
784                     $sqlListaCliente = "SELECT * FROM '
785                         clientes' ORDER BY 'id' ";
786                     $queryListaCliente = mysql_query(
787                         $sqlListaCliente,$con);
788                     while ($rowListaCliente =
789                         mysql_fetch_array(
790                             $queryListaCliente)){
791                         $nombre = $rowListaCliente ['
792                             nombre'];
793                         echo '<option value="'. $nombre.
794                             '"">' . $nombre. '</option></a
795                             >';
796                     }
797                     echo '
798                     </select>
799                     <br>
800                     <hr>
801                     <div class="form-group" align="
802                         right">
803                         <input type="submit" class="btn
804                             btn-default" value="
805                             Seleccionar cliente">
806                     </div>
807                     </form>
808                 </div>
809             </div>
810             <!-- /.table-responsive -->
811         </div>
812     <!-- /.panel-body -->
813 </div>
814 <!-- /.panel -->
815 <hr>
816 ' ;
817 }
818 echo '

```

```

803         <!-- /.panel -->
804     </div>
805     ';
806 }
807
808 // Anyadir nueva entrada a la base de datos
809 else if($action == 'altaentrada'){
810     echo '<h1 class="page-header">Nueva entrada en la base de datos
811     </h1>';
812     echo ' <br><br>
813     <div class="col-lg-12">';
814     if (isset($_POST['table'])&&isset($_POST['nombre'])&&isset(
815         $_POST['tipo'])&&isset($_POST['medidas'])&&isset($_POST['
816         descripcion'])&&$_POST['table']!= ''&&$_POST['nombre']!= ''&&
817         $_POST['tipo']!= ''&&$_POST['medidas']!= ''&&$_POST['
818         descripcion']!= ''){
819         $table = $_POST['table']; $nombre = cleanString($_POST['
820         nombre']); $tipo = cleanString($_POST['tipo']);
821         $medidas = cleanString($_POST['medidas']); $descripcion
822         = cleanString($_POST['descripcion']);
823         if (isset($_POST['id'])&&$_POST['id']!= ''){
824             $id = $_POST['id'];
825             // Si el id esta duplicado hay que modificarlo
826             $sqlComprobarIdDuplicado = "SELECT * FROM ".$table."
827             WHERE id='".$id.'" ";
828             $queryComprobarIdDuplicado = mysql_query(
829                 $sqlComprobarIdDuplicado,$con);
830             $retValComprobarIdDuplicado = mysql_num_rows(
831                 $queryComprobarIdDuplicado);
832             if ($retValComprobarIdDuplicado!=0){
833                 echo '
834                 <div class="alert alert-danger alert-dismissible"
835                 align="center">
836                 <button type="button" class="close" data-dismiss="
837                 alert" aria-hidden="true">&times;</button>
838                 Identificador duplicado en la base de datos de <
839                 strong>'.$table.'</strong>, no puede repetir el
840                 identificador de la entrada.
841                 </div>
842                 ';
843                 modificaIDEntrada($table,$tipo,$nombre,$medidas,
844                 $descripcion,$con);
845             }
846             else{
847                 $datetime = date("Y-m-d H:i:s");
848                 $sqlAltaEntrada = "INSERT INTO ".$table." (id,
849                 nombre, tipo, medidas, descripcion, reg_date)
850                 VALUES ('$id', '$nombre', '$tipo', '$medidas',
851                 '$descripcion', '$datetime')";
852                 $retvalAltaEntrada = mysql_query($sqlAltaEntrada,
853                 $con);
854                 if ($retvalAltaEntrada){
855                     //Si existe imagen/plano lo almacenamos en la
856                     base de datos

```

```

836     if (isset($_FILES['imagen1']['name'])&&$_FILES['
837         imagen1']['name']!=""){
838         if (!$FILES['imagen1']['error']){
839             uploadPicByImgNum($_FILES['imagen1']['
840                 tmp_name'], basename($_FILES['
841                 imagen1']['name']), $_FILES['
842                 imagen1']['type'], $id, $table,
843                 $con, '1');
844         }
845     }
846     if (isset($_FILES['imagen2']['name'])&&$_FILES['
847         imagen2']['name']!=""){
848         if (!$FILES['imagen2']['error']){
849             uploadPicByImgNum($_FILES['imagen2']['
850                 tmp_name'], basename($_FILES['
851                 imagen2']['name']), $_FILES['
852                 imagen2']['type'], $id, $table,
853                 $con, '2');
854         }
855     }
856     if (isset($_FILES['imagen3']['name'])&&$_FILES['
857         imagen3']['name']!=""){
858         if (!$FILES['imagen3']['error']){
859             uploadPicByImgNum($_FILES['imagen3']['
860                 tmp_name'], basename($_FILES['
861                 imagen3']['name']), $_FILES['
862                 imagen3']['type'], $id, $table,
863                 $con, '3');
864         }
865     }
866     echo '
867     <br>
868     <div class="col-lg-10">
869         <div class="panel panel-default">
870             <div class="panel-body">
871                 <br>
872                 <div class="alert alert-success alert-
873                     dismissable" align="center">
874                     <button type="button" class="close"
875                         data-dismiss="alert" aria-
876                         hidden="true">&times;</button>
877                     Entrada anyadida correctamente en
878                     la tabla [ <strong>'. $table. '</
879                     strong> ].
880                 </div>
881             </div>
882         </div>
883     </div>';
884 }
885 else{
886     echo '
887     <div class="alert alert-danger alert-
888         dismissable" align="center">

```

```
868     <button type="button" class="close" data-
      dismiss="alert" aria-hidden="true">&times
      </button>
869     Error en la base de datos. Tome nota del error
      y contacte con el administrador del sistema
      . Algun valor introducido en los datos no
      es correcto: '. mysql_error();
870     echo '</div>
871     '
872     }
873     }
874 }
875 else {
876     // Si no hay identificador anyadimos uno en base al
      total de entradas de la tabla
877     $sqlCuentaEntradas = "SELECT COUNT(*) FROM ".$table;
878     $queryCuentaEntradas = mysql_query($sqlCuentaEntradas ,
      $con);
879     $rowCuentaEntradas = mysql_fetch_row(
      $queryCuentaEntradas);
880     $retVal = $rowCuentaEntradas[0];
881     $id = $retVal+1;
882     $datetime = date("Y-m-d H:i:s");
883     $sqlAltaEntrada = "INSERT INTO ".$table." (id, nombre,
      tipo, medidas, descripcion, reg_date) VALUES ('".
      $id." ', '$nombre', '$tipo', '$medidas', '
      $descripcion', '$datetime')";
884     $retvalAltaEntrada = mysql_query($sqlAltaEntrada , $con);
885     if ($retvalAltaEntrada) {
886         //Si existe imagen/plano lo almacenamos en la base
      de datos
887         if (isset($_FILES['imagen1']['name'])&&$_FILES['
      imagen1']['name']!=""){
888             if (!$FILES['imagen1']['error']) {
889                 uploadPicByImgNum($_FILES['imagen1']['
      tmp_name'], basename($_FILES['imagen1'
      ]['name']), $_FILES['imagen1']['type'],
      $id, $table, $con, '1');
890             }
891         }
892         if (isset($_FILES['imagen2']['name'])&&$_FILES['
      imagen2']['name']!=""){
893             if (!$FILES['imagen2']['error']) {
894                 uploadPicByImgNum($_FILES['imagen2']['
      tmp_name'], basename($_FILES['imagen2'
      ]['name']), $_FILES['imagen2']['type'],
      $id, $table, $con, '2');
895             }
896         }
897         if (isset($_FILES['imagen3']['name'])&&$_FILES['
      imagen3']['name']!=""){
898             if (!$FILES['imagen3']['error']) {
899                 uploadPicByImgNum($_FILES['imagen3']['
      tmp_name'], basename($_FILES['imagen3'
```

```

[[ 'name' ]), $_FILES[ 'imagen3' ][ 'type' ],
    $id, $table, $con, '3');
900     }
901     }
902     echo '
903     <br>
904     <div class="col-lg-10">
905         <div class="panel panel-default">
906             <div class="panel-body">
907                 <br>
908                 <div class="alert alert-success alert-
909                     dismissable" align="center">
910                     <button type="button" class="close"
911                         data-dismiss="alert" aria-hidden="
912                         true">&times;</button>
913                     Entrada anyadida correctamente en la
914                     tabla [ <strong>'. $table. '</strong>
915                     ].
916                 </div>
917             </div>
918         </div>
919     </div>';
920 }
921 else {
922     echo '
923     <div class="alert alert-danger alert-dismissable"
924         align="center">
925     <button type="button" class="close" data-dismiss="
926         alert" aria-hidden="true">&times;</button>
927     Error en la base de datos. Tome nota del error y
928     contacte con el administrador del sistema.
929     Algun valor introducido en los datos no es
930     correcto: '. mysql_error();
931     echo '</div>
932     ';
933 }
934 }
935 }
936 }
937 else {
938     // Introducimos los datos de la nueva entrada
939     echo '
940     <div class="panel panel-default">
941         <div class="panel-heading">
942             Anyadir nueva entrada a la base de datos
943         </div>
944         <div class="panel-body">
945             <div class="row">
946                 <div class="col-lg-10" style="margin-left
947                     :40px;">
948                     <form enctype="multipart/form-data"
949                         role="form" action="accesoAdmin.php
950                         ?action=altaentrada" method="post">

```

```
938 <p><h3><u>Datos de la entrada</u></h3></p>
939 <br>
940 <div class="form-group">
941   <label>Selecciona un cliente:</label>
942   <select class="form-control" name="table" required>
943     <option value="" selected="selected"></option>' ;
944     $sqlListaCliente = "SELECT * FROM 'clientes' ORDER BY 'id' ";
945     $queryListaCliente = mysql_query($sqlListaCliente, $con);
946     while ($rowListaCliente = mysql_fetch_array($queryListaCliente)){
947       $nombre = $rowListaCliente['nombre'];
948       echo '<option value="' . $nombre . '>' . $nombre . '</option></a>' ;
949     }
950     echo '
951   </select>
952 </div>
953 <div class="form-group">
954   <label>Identificador</label> (indicar solo si existe la referencia) <strong></strong>
955   <input class="form-control" type="text" placeholder="Introduce el identificador" name="id">
956 </div>
957 <div class="form-group">
958   <label>Tipo:</label>
959   <input class="form-control" type="text" placeholder="Introduce el tipo" name="tipo" required>
960 </div>
961 <div class="form-group">
962   <label>Nombre:</label>
963   <input class="form-control" type="text" placeholder="Introduce el nombre" name="nombre" required>
964 </div>
965 <div class="form-group">
966   <label>Medidas:</label>
```

```

967         <input class="form-control"
          type="text" placeholder="
          Introduce las medidas" name
          ="medidas" required>
968     </div>
969     <div class="form-group">
970         <label>Descripci&oacute;n:</
          label>
971         <textarea class="form-control"
          rows="5" placeholder="
          Introduce la descripci&
          oacute;n" name="descripcion
          " required></textarea>
972     </div>
973     <div class="form-group">
974         <label>Plano/Imagen 1 (Opcional
          ):</label>
975         <input type="file" name="
          imagen1">
976     </div>
977     <div class="form-group">
978         <label>Plano/Imagen 2 (Opcional
          ):</label>
979         <input type="file" name="
          imagen2">
980     </div>
981     <div class="form-group">
982         <label>Plano/Imagen 3 (Opcional
          ):</label>
983         <input type="file" name="
          imagen3">
984     </div>
985     <br>
986     <hr>
987     <div align="right">
988         <button type="reset" class="btn
          btn-default">Borrar datos
          </button>
989         <input type="submit" class="btn
          btn-default" style="
          background-color: green;
          color: white;" value="
          Anyadir entrada">
990     </div>
991     <br>
992     </form>
993 </div>
994 <!-- /.col-lg-10 (nested) -->
995 <br><br>
996 </div>
997 <!-- /.row (nested) -->
998 </div>
999 <!-- /.panel-body -->
1000 </div>

```



```

1001         <!-- /.panel -->
1002         <br>
1003     </div>';
1004     }
1005 }
1006
1007 // Modificar entrada de la base de datos
1008 else if($action == 'modificaentrada'){
1009     echo '<h1 class="page-header">Modificar entrada de la base de
1010     datos</h1>
1011         <br><br>
1012         <div class="col-lg-12">';
1013     if(!isset($_POST['selectedCliente'])){
1014         // Seleccionamos el Cliente
1015         echo '
1016             <div class="panel panel-default">
1017                 <div class="panel-heading">
1018                     Listado de clientes
1019                 </div>
1020                 <!-- /.panel-heading -->
1021                 <div class="panel-body">
1022                     <br>
1023                     <div class="col-lg-10" style="margin-left:40px
1024                     ;">
1025                         <label>Selecciona un cliente:</label>
1026                         <form action="accesoAdmin.php?action=
1027                             modificaentrada" method="post">
1028                             <select class="form-control" name="
1029                                 selectedCliente" required>
1030                             <option value="" selected="selected"></
1031                                 option>';
1032                             $sqlListaCliente = "SELECT * FROM 'clientes
1033                                 ' ORDER BY 'id' ";
1034                             $queryListaCliente = mysql_query(
1035                                 $sqlListaCliente,$con);
1036                             while ($rowListaCliente = mysql_fetch_array
1037                                 ($queryListaCliente)){
1038                                 $nombre = $rowListaCliente ['nombre'];
1039                                 $id = $rowListaCliente ['id'];
1040                                 echo '<option value="' . $id . '">' .
1041                                     $nombre . '</option></a>';
1042                             }
1043                             echo '
1044                             </select>
1045                             <br>
1046                             <hr>
1047                             <div class="form-group" align="right">
1048                                 <input type="submit" class="btn btn-
1049                                     default" value="Seleccionar cliente
1050                                     ">
1051                             </div>
1052                             </form>
1053                         </div>
1054                     </div>
1055                 </div>
1056             </div>
1057         </div>
1058     </div>
1059 }

```



```

1082         echo '<option value="' .
           $rowEntradas['id'] . '"'>
           ' . $rowEntradas['id'] . '
           - ' . $rowEntradas['tipo'
           ]. ' - ' . $rowEntradas['
           nombre'] . ' - ' .
           $rowEntradas['medidas'
           ]. '</option></a>';
1083     }
1084     echo '
1085     </select>
1086 </div>
1087 <input type="hidden" name="
           selectedCliente" value="' .
           $selectedCliente . '"'>
1088 <hr>
1089 <div class="form-group" align="right">
1090     <input type="submit" class="btn btn
           -default" value="Modificar
           entrada">
1091     </div>
1092 </form>
1093 </div>
1094 </div>
1095 <!-- /.table-responsive -->
1096 </div>
1097 <!-- /.panel-body -->
1098 </div>
1099 <!-- /.panel -->
1100 <hr>
1101 ';
1102 }
1103 else if (isset($_POST['tableName'])&&$_POST['tableName']!= ''&&
1104         isset($_POST['entrada'])&&$_POST['entrada']!= ''&&!isset(
1105         $_POST['okDrop'])) {
1106     // Modificamos los datos
1107     $entrada = $_POST['entrada'];
1108     $tableName = $_POST['tableName'];
1109     $sqlEntrada = "SELECT * FROM " . $tableName . " WHERE id=" .
1110     $entrada . "'";
1111     $queryEntrada = mysql_query($sqlEntrada, $con);
1112     $rowEntrada = mysql_fetch_array($queryEntrada);
1113     echo '
1114     <div class="col-lg-12">
1115     <div class="panel panel-default">
1116     <div class="panel-heading">
1117     Entrada [ <strong>' . $tableName . ' - ' .
1118     $entrada . '</strong> ]
1119     </div>
1120     <!-- /.panel-heading -->
1121     <div class="panel-body">
1122     <div class="row">
1123     <div class="col-lg-10" style="margin-
1124     left:40px;">

```

```

1120 <form enctype="multipart/form-data"
      role="form" action="
      accesoAdmin.php?action=
      modificaentrada" method="post">
1121 <div class="form-group">
1122 <label>ID:</label>
1123 <input class="form-control"
      value="' . $rowEntrada[ '
      id ' ]. '" type="text"
      name="id" disabled>' . '
1124 </div>
1125 <div class="form-group">
1126 <label>Tipo:</label>
1127 <input class="form-control"
      value="' . $rowEntrada[ '
      tipo ' ]. '" type="text"
      name="tipo" required">'
      . '
1128 </div>
1129 <div class="form-group">
1130 <label>Nombre:</label>
1131 <input class="form-control"
      value="' . $rowEntrada[ '
      nombre ' ]. '" type="text"
      name="nombre" required
      ">' . '
1132 </div>
1133 <div class="form-group">
1134 <label>Medidas:</label>
1135 <input class="form-control"
      value="' . $rowEntrada[ '
      medidas ' ]. '" type="text"
      name="medidas"
      required">' . '
1136 </div>
1137 <div class="form-group">
1138 <label>Descripci&oacute;n
      :</label>
1139 <textarea class="form-
      control" rows="5" name
      ="descripcion" required
      >' . $rowEntrada[ '
      descripcion ' ]. '</
      textarea >
1140 </div>
1141 <label>Imagen/Plano
      almacenado:</label>';
1142 // imagen 1
1143 if ($rowEntrada[ 'imagen1 ' ]!=
      NULL) {
1144     echo '
1145         <td class="center">
1146             <br/>1 - <a
      target=

```

```
1147         _blank href
1148         ="' .
1149         $rowEntrada
1150         [ 'imagen1 '
1151         ]. "'> ' .
1152         $rowEntrada
1153         [ 'nombre' ].
1154         '</a><br/>
1155         Cambiar Plano/
1156         Imagen:
1157         <input type="
1158         file" name
1159         ="
1160         newImagen1
1161         ">
1162     </td>
1163     ' ;
1164 }
1165 else {
1166     echo '<td class="center
1167     "><br/>1 -<br/></td
1168     >
1169     Cambiar Plano/
1170     Imagen:
1171     <input type=" file "
1172     name="
1173     newImagen1">
1174     ' ;
1175 }
1176 // imagen 2
1177 if ($rowEntrada[ 'imagen2 ' ]!=
1178 NULL) {
1179     echo '
1180     <td class="center">
1181     <br/>2 - <a
1182     target=
1183     _blank href
1184     ="' .
1185     $rowEntrada
1186     [ 'imagen2 '
1187     ]. "'> ' .
1188     $rowEntrada
1189     [ 'nombre' ].
1190     '</a><br/>
1191     Cambiar Plano/
1192     Imagen:
1193     <input type="
1194     file" name
1195     ="
1196     newImagen2
1197     ">
1198     </td>
1199     ' ;
1200 }
```

```

1168         else {
1169             echo '<td class="center
1170                 "><br/>2 -<br/></td
1171                 >
1172                 Cambiar Plano/
1173                 Imagen:
1174                 <input type=" file "
1175                 name="
1176                 newImagen2">
1177                 '
1178             ;
1179         }
1180         // imagen 3
1181         if ($rowEntrada[ 'imagen3' ]!=
1182         NULL) {
1183             echo '
1184             <td class="center">
1185             <br/>3 - <a
1186                 target=
1187                 _blank href
1188                 ="' .
1189                 $rowEntrada
1190                 [ 'imagen3'
1191                 ]. "'> ' .
1192                 $rowEntrada
1193                 [ 'nombre' ].
1194                 '</a><br/>
1195                 Cambiar Plano/
1196                 Imagen:
1197                 <input type="
1198                 file" name
1199                 ="
1200                 newImagen3
1201                 ">
1202             </td>
1203             '
1204             ;
1205         }
1206         else {
1207             echo '<td class="center
1208                 "><br/>3 -<br/></td
1209                 >
1210                 Cambiar Plano/
1211                 Imagen:
1212                 <input type=" file "
1213                 name="
1214                 newImagen3">
1215                 '
1216             ;
1217         }
1218         echo '
1219         <br>
1220         <div class="form-group">
1221         <input type="hidden" name="
1222         tableName" value="' .
1223         $tableName. "'>

```

```

1194         <input type="hidden" name="
           entrada" value="' .
           $entrada. '">
1195         <input type="hidden" name="
           selectedCliente" value
          ="' . $_POST[ '
           selectedCliente ' ]. '">
1196         <input type="hidden" name="
           okDrop" value="okDrop">
1197     </div>
1198     <hr>
1199     <div align="right">
1200         <input type="submit" class
           ="btn btn-default"
           value="Modificar
           Entrada">
1201     </div>
1202     <br>
1203     </form>
1204     </div>
1205     </div>
1206     </div>
1207     </div>
1208     <!-- /.panel-body -->
1209 </div>
1210 <!-- /.panel -->
1211 </div>
1212 ;
1213 }
1214 else if (isset($_POST['tableName'])&&$_POST['tableName']!= ''&&
isset($_POST['entrada'])&&$_POST['entrada']!= ''&&isset(
$_POST['okDrop']))
1215     &&isset($_POST['tipo'])&&isset($_POST['nombre'])&&isset
($POST['medidas'])&&isset($_POST['descripcion'])
1216     &&$_POST['tipo']!= ''&&$_POST['nombre']!= ''&&$_POST[ '
medidas']!= ''&&$_POST['descripcion']!= ''){
1217     // Editamos la base de datos
1218     $tableName = $_POST['tableName'];
1219     $entrada = $_POST['entrada'];
1220     $reg_date = date("Y-m-d H:i:s");
1221     $sqlModificarEntrada = "UPDATE ".$tableName." SET tipo='".
cleanString($_POST['tipo'])."', nombre='".cleanString(
$_POST['nombre'])."', medidas='".cleanString($_POST[ '
medidas'])."', descripcion='".cleanString($_POST[ '
descripcion'])."', reg_date='".$reg_date."' WHERE id='".
.$entrada.'" ";
1222     $retvalModificarEntrada = mysql_query($sqlModificarEntrada ,
$con);
1223     if($retvalModificarEntrada){
1224         // Si se cambia el plano lo subimos al sistema de
ficheros
1225         if (isset($_FILES['newImagen1']['name'])&&$_FILES[ '
newImagen1']['name']!= ""){
1226             if (!$FILES['newImagen1']['error']){

```

```

1227     $sqlOldPic = "SELECT * FROM ".$tableName."
1228             WHERE id='".$entrada."'";
1229     $queryOldPic = mysql_query($sqlOldPic,$con);
1230     $rowOldPic = mysql_fetch_array($queryOldPic);
1231     $pathOldPic = $rowOldPic['imagen1'];
1232     if($pathOldPic!=NULL){
1233         list($db, $path) = split('tfm/',
1234             $pathOldPic);
1235         unlink($path);
1236     }
1237     uploadPicByImgNum($_FILES['newImagen1']['
1238         tmp_name'], basename($_FILES['newImagen1']['
1239         name']), $_FILES['newImagen1']['type'],
1240         $entrada, $tableName, $con, '1');
1241 }
1242 if(isset($_FILES['newImagen2']['name'])&&$_FILES['
1243     newImagen2']['name']!=""){
1244     if(!$_FILES['newImagen2']['error']){
1245         $sqlOldPic = "SELECT * FROM ".$tableName."
1246             WHERE id='".$entrada."'";
1247         $queryOldPic = mysql_query($sqlOldPic,$con);
1248         $rowOldPic = mysql_fetch_array($queryOldPic);
1249         $pathOldPic = $rowOldPic['imagen2'];
1250         if($pathOldPic!=NULL){
1251             list($db, $path) = split('tfm/',
1252                 $pathOldPic);
1253             unlink($path);
1254         }
1255         uploadPicByImgNum($_FILES['newImagen2']['
1256             tmp_name'], basename($_FILES['newImagen2']['
1257             name']), $_FILES['newImagen2']['type'],
1258             $entrada, $tableName, $con, '2');
1259     }
1260 }
1261 if(isset($_FILES['newImagen3']['name'])&&$_FILES['
1262     newImagen3']['name']!=""){
1263     if(!$_FILES['newImagen3']['error']){
1264         $sqlOldPic = "SELECT * FROM ".$tableName."
1265             WHERE id='".$entrada."'";
1266         $queryOldPic = mysql_query($sqlOldPic,$con);
1267         $rowOldPic = mysql_fetch_array($queryOldPic);
1268         $pathOldPic = $rowOldPic['imagen3'];
1269         if($pathOldPic!=NULL){
1270             list($db, $path) = split('tfm/',
1271                 $pathOldPic);
1272             unlink($path);
1273         }
1274         uploadPicByImgNum($_FILES['newImagen3']['
1275             tmp_name'], basename($_FILES['newImagen3']['
1276             name']), $_FILES['newImagen3']['type'],
1277             $entrada, $tableName, $con, '3');
1278     }
1279 }

```



```

1305     <select class="form-control" name="
1306         selectedCliente" disabled>
1307         <option value="'. $selectedCliente. '
1308             " selected="selected">'.
1309             $tableName. '</option>
1310     </select>
1311     <input type="hidden" name="tableName"
1312         value="'. $tableName. '">
1313     <br>
1314     <div class="form-group">
1315         <label>Selecciona una entrada:</
1316         label>
1317         <select class="form-control" name="
1318             entrada" required>
1319             <option value="" selected="
1320                 selected"></option>' ;
1321             $sqlEntradas = "SELECT * FROM "
1322                 . $tableName;
1323             $retvalEntradas = mysql_query(
1324                 $sqlEntradas, $con);
1325             while ($rowEntradas =
1326                 mysql_fetch_array(
1327                     $retvalEntradas)){
1328                 echo '<option value="'.
1329                     $rowEntradas['id']. '"">
1330                     '. $rowEntradas['id']. '
1331                     - '. $rowEntradas['tipo'
1332                     ]. ' - '. $rowEntradas['
1333                     nombre']. ' - '.
1334                     $rowEntradas['medidas'
1335                     ]. '</option></a>';
1336             }
1337             echo '
1338         </select>
1339     </div>
1340     <input type="hidden" name="okDrop"
1341         value="okDrop">
1342     <hr>
1343     <div class="form-group" align="right">
1344         <p><strong>Esta accion no tiene
1345             vuelta atras</strong></p>
1346         <input type="submit" class="btn btn
1347             -default" style="background-
1348             color:red; color:white;" value
1349             ="Eliminar entrada">
1350     </div>
1351 </form>
1352 </div>
1353 </div>
1354 <!-- /.table-responsive -->
1355 </div>
1356 <!-- /.panel-body -->
1357 </div>
1358 <!-- /.panel -->

```

```

1336         <hr>
1337     ';
1338 }
1339 else if (isset($_POST['tableName'])&&$_POST['tableName']!= ''&&
1340         isset($_POST['okDrop'])&&isset($_POST['entrada'])&&$_POST['
1341         entrada']!= ''){
1342     $tableName = $_POST['tableName'];
1343     $entrada = $_POST['entrada'];
1344     $sqlPlanoEntrada = "SELECT imagen FROM ".$tableName." WHERE
1345         id='".$entrada."'";
1346     $queryPlanoEntrada = mysql_query($sqlPlanoEntrada,$con);
1347     $rowPlanoEntrada = mysql_fetch_row($queryPlanoEntrada);
1348     $planoEntrada = $rowPlanoEntrada[0];
1349     $sqlDropEntrada = "DELETE FROM ".$tableName." WHERE id='".$
1350         $entrada."'";
1351     $retvalDropEntrada= mysql_query($sqlDropEntrada,$con);
1352     if($retvalDropEntrada){
1353         // Si existe plano asociado a la entrada lo eliminamos
1354         if($planoEntrada!=NULL){
1355             list($db,$path) = split('tfm/', $planoEntrada);
1356             unlink($path);
1357         }
1358         echo '
1359         <div class="alert alert-success alert-dismissable"
1360             align="center">
1361             <button type="button" class="close" data-dismiss="
1362             alert" aria-hidden="true">&times;</button>
1363             Entrada eliminada correctamente.
1364         </div>
1365         ';
1366     }
1367     else{
1368         echo '
1369         <div class="alert alert-danger alert-dismissable" align
1370             ="center">
1371         <button type="button" class="close" data-dismiss="alert
1372             " aria-hidden="true">&times;</button>
1373         Error en la base de datos. Tome nota del error y
1374         contacte con el administrador del sistema. Algun
1375         valor introducido en los datos no es correcto: '.
1376         mysql_error();
1377         echo '</div>
1378         ';
1379     }
1380 }
1381 else{
1382     echo '
1383     <div class="panel panel-default">
1384     <div class="panel-heading">
1385         Listado de clientes
1386     </div>
1387     <!-- /.panel-heading -->
1388     <div class="panel-body">
1389     <br>

```

```

1379         <div class="col-lg-10" style="margin-left:40px
1380         ;">
1381         <label>Selecciona un cliente:</label>
1382         <form action="accesoAdmin.php?action=
1383         eliminaentrada" method="post">
1384         <select class="form-control" name="
1385         selectedCliente" required>
1386         <option value="" selected="selected"></
1387         option>';
1388         $sqlListaCliente = "SELECT * FROM 'clientes
1389         ' ORDER BY 'id' ";
1390         $queryListaCliente = mysql_query(
1391         $sqlListaCliente,$con);
1392         while ($rowListaCliente =
1393         mysql_fetch_array($queryListaCliente)){
1394         $nombre = $rowListaCliente ['nombre'];
1395         $id = $rowListaCliente ['id'];
1396         echo '<option value="'. $id. '">'.
1397         $nombre. '</option></a>';
1398         }
1399         echo '
1400         </select>
1401         <br>
1402         <hr>
1403         <div class="form-group" align="right">
1404         <input type="submit" class="btn btn-
1405         default" value="Seleccionar cliente
1406         ">
1407         </div>
1408         </form>
1409         </div>
1410         </div>
1411         <!-- /.table-responsive -->
1412     </div>
1413     <!-- /.panel-body -->
1414 </div>
1415 <!-- /.panel -->
1416 <hr>
1417 ';
1418 }
1419 }
1420
1421 // Alta nuevo usuario
1422 else if($action == 'altauser'){
1423     echo '<h1 class="page-header">Alta nuevo usuario</h1>
1424     <br><br>
1425     <div class="col-lg-12">
1426     ';
1427     if (isset($_POST['nombre'])&&isset($_POST['apellidos'])&&isset(
1428     $_POST['user'])&&isset($_POST['pass'])&&isset($_POST['cargo
1429     '])
1430     &&$_POST['nombre']!= ''&&$_POST['apellidos']!= ''&&$_POST['
1431     user']!= ''&&$_POST['pass']!= ''&&$_POST['cargo']!= ''){

```

```
1419 $nombre = $_POST['nombre']; $apellidos = $_POST['apellidos '
1420 ];
1421 $user = $_POST['user']; $pass = $_POST['pass']; $cargo =
1422 $_POST['cargo'];
1423 ////////////////////////////////////////////////////////////////////
1424 // COMPROBACIONES DE SEGURIDAD //
1425 ////////////////////////////////////////////////////////////////////
1426 $errorNick = $errorCheckList = False;
1427 // Comprobacion de seguridad para los checklist
1428 // Si no esta seleccionado el tipo de usuario mostramos
1429 // mensaje de error
1430 if(empty($_POST['check_list'])) {
1431     $errorCheckList = True;
1432     echo '
1433         <div class="alert alert-danger alert-dismissable"
1434             align="center">
1435             <button type="button" class="close" data-dismiss="
1436                 alert" aria-hidden="true">&times;</button>
1437             Debe indicar el tipo de usuario.
1438         </div>
1439     ';
1440 }
1441 else {
1442     foreach($_POST['check_list'] as $check){
1443         // Comprobacion de seguridad para comprobar si ya
1444         // existe el nombre de usuario
1445         $sqlChekNick = "SELECT * FROM usuarios_". $check. "
1446             WHERE user='". $user. "'";
1447         $queryChekNick = mysql_query($sqlChekNick, $con);
1448         $retValChekNick = mysql_num_rows($queryChekNick);
1449         if($retValChekNick!=0){
1450             $errorNick = True;
1451             echo '
1452                 <div class="alert alert-danger alert-
1453                     dismissable" align="center">
1454                     <button type="button" class="close"
1455                         data-dismiss="alert" aria-hidden="
1456                             true">&times;</button>
1457                     Nombre de usuario ya existente en la
1458                     base de datos de <strong>usuarios '
1459                     .ucwords($check).'</strong>.
1460                 </div>
1461             ';
1462         }
1463     }
1464 }
1465 ////////////////////////////////////////////////////////////////////
1466 // FIN COMPROBACIONES DE SEGURIDAD //
1467 ////////////////////////////////////////////////////////////////////
1468 if($errorNick||$errorCheckList){
1469     modificaAltaUser($nombre,$apellidos,$user,$pass,$cargo,
1470         $errorNick,$errorCheckList);
1471 }
1472 else {
```

```

1460 // Anyadimos cada tipo de usuario en su tabla
1461 // correspondiente
1462 foreach($_POST['check_list'] as $check) {
1463     $reg_date = date("Y-m-d H:i:s");
1464     $sqlAltaUser = "INSERT INTO usuarios_". $check." (
1465         nombre, apellidos, user, pass, cargo, reg_date)
1466         VALUES ('$nombre', '$apellidos', '$user', '$
1467             $pass', '$cargo', '$reg_date')";
1468     $retvalAltaUser = mysql_query($sqlAltaUser, $con);
1469     if($retvalAltaUser){
1470         //Si existe avatar lo almacenamos en la base de
1471         // datos
1472         if (isset($_FILES['imagenUser']['name']) &&
1473             $_FILES['imagenUser']['name'] != "") {
1474             if (!$_FILES['imagenUser']['error']) {
1475                 $sqlGetID = "SELECT * FROM usuarios_".
1476                     $check." WHERE user='". $user.'" AND
1477                     pass='". $pass.'"';
1478                 $queryGetID = mysql_query($sqlGetID,
1479                     $con);
1480                 $rowGetID = mysql_fetch_array(
1481                     $queryGetID);
1482                 uploadPicUser($_FILES['imagenUser']['
1483                     tmp_name'], $_FILES['imagenUser']['
1484                     name'], $_FILES['imagenUser']['type
1485                     '], $rowGetID['id'], $user, '
1486                     usuarios_'. $check, $con);
1487             }
1488         }
1489         echo '
1490         <div class="alert alert-success alert-
1491             dismissable" align="center">
1492             <button type="button" class="close"
1493                 data-dismiss="alert" aria-hidden="
1494                 true">&times;</button>
1495             Usuario anyadido correctamente en la
1496             lista de usuarios <strong>'. $check.'
1497             </strong>.
1498         </div>
1499         ';
1500     }
1501     else {
1502         echo '
1503         <div class="alert alert-danger alert-
1504             dismissable" align="center">
1505             <button type="button" class="close" data-
1506                 dismiss="alert" aria-hidden="true">&
1507                 times;</button>
1508             No se ha podido anyadir el usuario a la
1509             lista de usuarios <strong>'. $check.' </
1510             strong>. Tome nota del error y contacte
1511             con el administrador del sistema.
1512             Algun valor introducido en los datos no
1513             es correcto: '. mysql_error();

```

```
1488         echo '</div>'
1489     ';
1490     }
1491     }
1492     }
1493 }
1494 else {
1495     echo '
1496     <div class="panel panel-default">
1497         <div class="panel-heading">
1498             Anyadir nuevo usuario
1499         </div>
1500         <div class="panel-body">
1501             <div class="row">
1502                 <div class="col-lg-10" style="margin-left
1503                 :40px;">
1504                     <form enctype="multipart/form-data"
1505                     role="form" action="accesoAdmin.php
1506                     ?action=altauser" method="post">
1507                         <p><h3><u>Datos del usuario</u></h3
1508                         ></p>
1509                         <br>
1510                         <div class="form-group">
1511                             <label>Seleccione el tipo de
1512                             usuario: </label>
1513                             <div class="checkbox">
1514                                 <label>
1515                                     <input type="checkbox"
1516                                     name="check_list[]"
1517                                     value="web">
1518                                     Usuario Web
1519                                 </label>
1520                             </div>
1521                             <div class="checkbox">
1522                                 <label>
1523                                     <input type="checkbox"
1524                                     name="check_list[]"
1525                                     value="app">
1526                                     Usuario App
1527                                 </label>
1528                             </div>
1529                         </div>
1530                         <div class="form-group">
1531                             <label>Nombre:</label>
1532                             <input class="form-control"
1533                             type="text" placeholder="
1534                             Introduce el nombre" name="
1535                             nombre" required>
1536                         </div>
1537                         <div class="form-group">
1538                             <label>Apellidos:</label>
1539                             <input class="form-control"
1540                             type="text" placeholder="
1541                             Introduce el apellido" name="
1542                             apellidos" required>
1543                         </div>
1544                     </form>
1545                 </div>
1546             </div>
1547         </div>
1548     </div>
1549     ';
```

```

1526         Introduce los apellidos"
1527         name="apellidos" required>
1528     </div>
1529     <div class="form-group">
1530         <label>Cargo:</label>
1531         <input class="form-control"
1532             type="text" placeholder="
1533             Introduce el cargo dentro
1534             de la organizaci&oacute;n"
1535             name="cargo" required>
1536     </div>
1537     <label>Nombre de usuario:</label>
1538     <div class="form-group input-group
1539         ">
1540         <span class="input-group-addon
1541             ">@</span>
1542         <input type="text" class="form-
1543             control" placeholder="
1544             Introduce el nombre de
1545             usuario" name="user"
1546             required>
1547     </div>
1548     <div class="form-group">
1549         <label>Contrasena:</label>
1550         <input class="form-control"
1551             type="text" placeholder="
1552             Introduce la contrasena"
1553             name="pass" required>
1554     </div>
1555     <div class="form-group">
1556         <label>Avatar de usuario:</
1557         label>
1558         <input type="file" name="
1559             imagenUser">
1560     </div>
1561     <br>
1562     <hr>
1563     <div align="right">
1564         <button type="reset" class="btn
1565             btn-default">Borrar datos
1566         </button>
1567         <input type="submit" class="btn
1568             btn-default" style="
1569             background-color:green;
1570             color:white;" value="
1571             Anyadir usuario">
1572     </div>
1573     <br>
1574     </form>
1575 </div>
1576 <!-- /.col-lg-10 (nested) -->
1577 <br><br>
1578 </div>
1579 <!-- /.row (nested) -->

```



```

1557         </div>
1558         <!-- /.panel-body -->
1559     </div>
1560     <br><br>
1561     <!-- /.panel -->
1562 </div>';
1563     }
1564 }
1565
1566 // Listar los usuarios Web
1567 else if($action == 'listauserweb'){
1568     listaUsers('Web',$con);
1569 }
1570
1571 // Modificar los usuarios Web
1572 else if($action == 'modificauserweb'){
1573     echo '<h1 class="page-header">Modificar usuario Web</h1>
1574     <br><br>
1575     <div class="col-lg-12">
1576     ';
1577     if(isset($_POST['OK'])){
1578         $nombre = $_POST['nombre']; $apellidos = $_POST['apellidos']
1579         ]; $cargo = $_POST['cargo']; $user = $_POST['user'];
1580         $pass = $_POST['pass'];
1581         $selectedUser = $_POST['selectedUser'];
1582         $sqlModificarUser = "UPDATE 'usuarios_web' SET cargo='".
1583         $cargo."', nombre='". $nombre."', apellidos='".
1584         $apellidos."',
1585         user='". $user."', pass='". $pass." ' WHERE id='".
1586         $selectedUser." '";
1587         $retvalModificarUser = mysql_query($sqlModificarUser,$con);
1588         if($retvalModificarUser){
1589             // Si se cambia el avatar lo subimos al sistema de
1590             ficheros
1591             if(isset($_FILES['newImagenUser']['name'])&&$_FILES['
1592             newImagenUser']['name']!=""){
1593                 if(!$_FILES['newImagenUser']['error']){
1594                     $sqlOldPic = "SELECT * FROM usuarios_web WHERE
1595                     id='". $selectedUser." '";
1596                     $queryOldPic = mysql_query($sqlOldPic,$con);
1597                     $rowOldPic = mysql_fetch_array($queryOldPic);
1598                     $pathOldPic = $rowOldPic['imagen'];
1599                     if($pathOldPic!=NULL){
1600                         unlink($pathOldPic);
1601                     }
1602                     uploadPicUser($_FILES['newImagenUser']['
1603                     tmp_name'], $_FILES['newImagenUser']['name']
1604                     ], $_FILES['newImagenUser']['type'],
1605                     $selectedUser, $user, 'usuarios_web', $con)
1606                     ;
1607                 }
1608             }
1609         }
1610     }
1611     echo '

```



```
1634         value="'. $nombre. '"
1635         required >
1636     </div>
1637     <div class="form-group">
1638         <label>Apellidos:</label>
1639         <input class="form-control"
1640             type="text" name="apellidos"
1641             value="'. $apellidos. '"
1642             required >
1643     </div>
1644     <div class="form-group">
1645         <label>Cargo:</label>
1646         <input class="form-control"
1647             type="text" name="cargo"
1648             value="'. $cargo. '" required
1649         >
1650     </div>
1651     <label>Nombre de usuario:</label>
1652     <div class="form-group input-group
1653         ">
1654         <span class="input-group-addon
1655             ">@</span>
1656         <input type="text" class="form-
1657             control" value="'. $user. '"
1658             name="user" required >
1659     </div>
1660     <div class="form-group">
1661         <label>Contraseña:</label>
1662         <input class="form-control"
1663             type="text" value="'. $pass.
1664             '" name="pass" required >
1665     </div>
1666     <div>
1667         <label>Avatar almacenado:</
1668             label>';
1669         if ($rowDatosUser[ 'imagen' ]!=
1670             NULL) {
1671             echo '<p><a target=_blank
1672                 href="'. $rowDatosUser[ '
1673                 imagen' ]. '">' .
1674                 $rowDatosUser[ 'user' ]. '
1675                 </a></p>';
1676         }
1677         else {
1678             echo '<p>-</p>';
1679         }
1680         echo '
1681     </div>
1682     <div class="form-group">
1683         <label>Cambiar Avatar:</label>
1684         <input type="file" name="
1685             newImagenUser">
1686     </div>
1687     <br>
```

```

1667         <hr>
1668         <input type="hidden" name="OK"
1669             value="OK">
1670         <div align="right">
1671             <input type="submit" class="btn
1672                 btn-default" value="
1673                 Modificar usuario">
1674             </div>
1675         <br>
1676     </form>
1677 </div>
1678 <!-- /.col-lg-10 (nested) -->
1679 <br><br>
1680 </div>
1681 <!-- /.row (nested) -->
1682 </div>
1683 <!-- /.panel-body -->
1684 </div>
1685 <!-- /.panel -->
1686 </div>';
1687 }
1688 else {
1689     echo ' <br><br>
1690     <div class="col-lg-10">
1691     <div class="panel panel-default">
1692     <div class="panel-heading">
1693         Listado de usuarios Web
1694     </div>
1695     <!-- /.panel-heading -->
1696     <div class="panel-body">
1697     <br>
1698     <div class="form-group">
1699     <label>Selecciona un usuario:</label>
1700     <form action="accesoAdmin.php?action=
1701         modifcauserweb" method="post">
1702     <select class="form-control" name="
1703         selectedUser" required>
1704     <option value="" selected="selected"></
1705         option>';
1706     $sqlListaUsers = "SELECT * FROM '
1707         usuarios_web' ORDER BY 'nombre' ";
1708     $queryListaUsers = mysql_query(
1709         $sqlListaUsers,$con);
1709     while ($rowListaUsers =
1710         mysql_fetch_array($queryListaUsers)
1711     ){
1712         $id = $rowListaUsers['id'];
1713         echo '<option value="'. $id. ' ">'.
1714             $rowListaUsers['user']. ' - '.
1715             $rowListaUsers['nombre']. ' '.
1716             $rowListaUsers['apellidos']. '</
1717             option></a>';
1718     }
1719     echo '

```



```

1751         $rowListaUsers[ 'nombre' ]. ' ' .
1752         $rowListaUsers[ 'apellidos' ]. '</
1753         option></a>';
1754     }
1755     echo '
1756     </select>
1757     <div align="right">
1758         <br>
1759         <button type="submit" class="btn
1760             btn-default" style="background-
1761             color:red; color:white;">
1762             Eliminar usuario</button>
1763     </div>
1764 </form>
1765 </div>
1766 </div>
1767 <!-- /.table-responsive -->
1768 </div>
1769 <!-- /.panel-body -->
1770 </div>
1771 <!-- /.panel -->
1772 </div>
1773 ';
1774 }
1775 elseif (isset($_POST[ 'selectedUser ' ])) {
1776     $userID = $_POST[ 'selectedUser ' ];
1777     // Si existe el mismo usuario en otra tabla de usuarios no
1778     // eliminamos el avatar
1779     $sqlGetUserData = "SELECT * FROM usuarios_web WHERE id='".
1780         $userID. "'";
1781     $queryGetUserData = mysql_query($sqlGetUserData,$con);
1782     $rowGetUserData = mysql_fetch_array($queryGetUserData);
1783     $sqlCheckAvatar = "SELECT * FROM usuarios_app WHERE user='".
1784         $rowGetUserData[ 'user' ]. "'";
1785     $queryCheckAvatar = mysql_query($sqlCheckAvatar,$con);
1786     $rowCheckAvatar = mysql_fetch_array($queryCheckAvatar);
1787     if (!$queryCheckAvatar || $rowCheckAvatar[ 'imagen' ]==NULL) {
1788         unlink($rowGetUserData[ 'imagen' ]);
1789     }
1790     ////////////////
1791     $sqlDropUser = "DELETE FROM 'usuarios_web' WHERE id='".
1792         $userID. "'";
1793     $retvalDropUser = mysql_query($sqlDropUser,$con);
1794
1795     if ($retvalDropUser) {
1796         echo '
1797         <div class="alert alert-success alert-dismissable"
1798             align="center">
1799             <button type="button" class="close" data-dismiss="
1800                 alert" aria-hidden="true">&times;</button>
1801             Usuario eliminado correctamente.
1802         </div>
1803         ';
1804     }

```

```

1793     else {
1794         echo '
1795         <div class="alert alert-danger alert-dismissible" align
1796             ="center">
1797         <button type="button" class="close" data-dismiss="alert
1798             " aria-hidden="true">&times;</button>
1799         Error en la base de datos. Tome nota del error y
1800             contacte con el administrador del sistema. Algun
1801             valor introducido en los datos no es correcto: '.
1802             mysql_error();
1803         echo '</div>
1804         ';
1805     }
1806 }
1807 }
1808 }
1809 // Listar los usuarios App
1810 else if($action == 'listauserapp'){
1811     listaUsers('App',$con);
1812 }
1813 // Modificar los usuarios App
1814 else if($action == 'modificauserapp'){
1815     echo '<h1 class="page-header">Modificar usuario App</h1>
1816     <br><br>
1817     <div class="col-lg-12">
1818     ';
1819     if(isset($_POST['OK'])){
1820         $nombre = $_POST['nombre']; $apellidos = $_POST['apellidos'
1821             ]; $cargo = $_POST['cargo']; $user = $_POST['user'];
1822         $pass = $_POST['pass'];
1823         $selectedUser = $_POST['selectedUser'];
1824         $sqlModificarUser = "UPDATE 'usuarios_app' SET cargo='".
1825             $cargo."' , nombre='". $nombre."' , apellidos='".
1826             $apellidos."' ,
1827             user='". $user."' , pass='". $pass."' WHERE id='".
1828             $selectedUser."'";
1829         $retvalModificarUser = mysql_query($sqlModificarUser,$con);
1830         if($retvalModificarUser){
1831             // Si se cambia el avatar lo subimos al sistema de
1832             ficheros
1833             if(isset($_FILES['newImagenUser']['name'])&&$_FILES['
1834                 newImagenUser']['name']!=""){
1835                 if(!$_FILES['newImagenUser']['error']){
1836                     $sqlOldPic = "SELECT * FROM usuarios_app WHERE
1837                         id='". $selectedUser."'";
1838                     $queryOldPic = mysql_query($sqlOldPic,$con);
1839                     $rowOldPic = mysql_fetch_array($queryOldPic);
1840                     $pathOldPic = $rowOldPic['imagen'];
1841                     if($pathOldPic!=NULL){
1842                         unlink($pathOldPic);
1843                     }
1844                     uploadPicUser($_FILES['newImagenUser']['
1845                         tmp_name'], $_FILES['newImagenUser']['name']

```

```

], $_FILES['newImagenUser']['type'],
selectedUser, $user, 'usuarios_app', $con)
;
1833     }
1834   }
1835   echo '
1836     <div class="alert alert-success alert-dismissible"
1837       align="center">
1838       <button type="button" class="close" data-dismiss="
1839         alert" aria-hidden="true">&times;</button>
1840       Usuario modificado correctamente.
1841     </div>
1842   ';
1843 }
1844 else {
1845   echo '
1846     <div class="alert alert-danger alert-dismissible" align
1847       ="center">
1848     <button type="button" class="close" data-dismiss="alert
1849       " aria-hidden="true">&times;</button>
1850     Error en la base de datos. Tome nota del error y
1851     contacte con el administrador del sistema. Algun
1852     valor introducido en los datos no es correcto: '.
1853     mysql_error();
1854   echo '</div>
1855   ';
1856 }
1857 }
1858 else if (isset($_POST['selectedUser'])) {
1859   $sqlDatosUser = "SELECT * FROM 'usuarios_app' WHERE id='".
1860     $_POST['selectedUser']."' ORDER BY 'id'";
1861   $queryDatosUser = mysql_query($sqlDatosUser,$con);
1862   $rowDatosUser= mysql_fetch_array($queryDatosUser);
1863   $nombre=$rowDatosUser['nombre']; $apellidos=$rowDatosUser['
1864     apellidos']; $user=$rowDatosUser['user']; $pass=
1865     $rowDatosUser['pass'];
1866   $cargo=$rowDatosUser['cargo'];
1867   echo '
1868     <div class="panel panel-default">
1869     <div class="panel-heading">
1870       Modificar usuario App
1871     </div>
1872     <div class="panel-body">
1873     <div class="row">
1874       <div class="col-lg-10" style="margin-left
1875         :40px;">
1876         <form enctype="multipart/form-data"
1877           role="form" action="accesoAdmin.php
1878             ?action=modificauserapp" method="
1879             post">
1880         <p><h3><u>Datos del usuario</u></h3
1881           ></p>
1882         <br>

```



```
1868 <input type="hidden" name="
1869     selectedUser" value="' . $_POST[ '
1870     selectedUser ' ]. '">
1871 <div class="form-group">
1872     <label>Nombre:</label>
1873     <input class="form-control"
1874         type="text" name="nombre"
1875         value="' . $nombre. '"
1876         required>
1877 </div>
1878 <div class="form-group">
1879     <label>Apellidos:</label>
1880     <input class="form-control"
1881         type="text" name="apellidos"
1882         value="' . $apellidos. '"
1883         required>
1884 </div>
1885 <div class="form-group">
1886     <label>Cargo:</label>
1887     <input class="form-control"
1888         type="text" name="cargo"
1889         value="' . $cargo. '" required
1890     >
1891 </div>
1892 <label>Nombre de usuario:</label>
1893 <div class="form-group input-group"
1894     ">
1895     <span class="input-group-addon"
1896         ">@</span>
1897     <input type="text" class="form-
1898         control" value="' . $user. '"
1899         name="user" required>
1900 </div>
1901 <div class="form-group">
1902     <label>Contrasena:</label>
1903     <input class="form-control"
1904         type="text" value="' . $pass.
1905         '" name="pass" required>
1906 </div>
1907 <div>
1908     <label>Avatar almacenado:</
1909     label>';
1910     if ($rowDatosUser[ 'imagen' ]!=
1911         NULL) {
1912         echo '<p><a target=_blank
1913             href="' . $rowDatosUser[ '
1914             imagen' ]. '">'.
1915             $rowDatosUser[ 'user' ]. '
1916             </a></p>';
1917     }
1918     else {
1919         echo '<p>-</p>';
1920     }
1921     echo '
```



```

1985     $queryListaUsers = mysql_query(
1986         $sqlListaUsers , $con);
1987     while ( $rowListaUsers =
1988         mysql_fetch_array( $queryListaUsers )
1989     ){
1990         $id = $rowListaUsers[ 'id' ];
1991         echo ' <option value="' . $id . '" > ' .
1992             $rowListaUsers[ 'user' ] . ' - ' .
1993             $rowListaUsers[ 'nombre' ] . ' ' .
1994             $rowListaUsers[ 'apellidos' ] . ' </
1995             option > </ a > ' ;
1996     }
1997     echo '
1998 </ select >
1999 < div align = " right " >
2000     < br >
2001     < button type = " submit " class = " btn
2002         btn - default " style = " background -
2003         color : red ; color : white ; " >
2004         Eliminar usuario </ button >
2005
2006     </ div >
2007 </ form >
2008 </ div >
2009 </ div >
2010 <!-- /.table-responsive -->
2011 </ div >
2012 <!-- /.panel-body -->
2013 </ div >
2014 <!-- /.panel -->
2015 </ div >
2016 ;
2017 }
2018 elseif ( isset ( $_POST [ 'selectedUser' ] ) ) {
2019     $userID = $_POST [ 'selectedUser' ];
2020     // Si existe el mismo usuario en otra tabla de usuarios no
2021     eliminamos el avatar
2022     $sqlGetUserData = "SELECT * FROM usuarios_app WHERE id = " .
2023         $userID . " " ;
2024     $queryGetUserData = mysql_query ( $sqlGetUserData , $con ) ;
2025     $rowGetUserData = mysql_fetch_array ( $queryGetUserData ) ;
2026     $sqlCheckAvatar = "SELECT * FROM usuarios_web WHERE user = "
2027         . $rowGetUserData [ 'user' ] . " " ;
2028     $queryCheckAvatar = mysql_query ( $sqlCheckAvatar , $con ) ;
2029     $rowCheckAvatar = mysql_fetch_array ( $queryCheckAvatar ) ;
2030     if ( ! $queryCheckAvatar || $rowCheckAvatar [ 'imagen' ] == NULL ) {
2031         unlink ( $rowGetUserData [ 'imagen' ] ) ;
2032     }
2033     ////////////////////////////////////
2034     $sqlDropUser = "DELETE FROM 'usuarios_app' WHERE id = " .
2035         $userID . " " ;
2036     $retvalDropUser = mysql_query ( $sqlDropUser , $con ) ;
2037     if ( $retvalDropUser ) {
2038         echo '

```

```

2024     <div class="alert alert-success alert-dismissible"
2025         align="center">
2026         <button type="button" class="close" data-dismiss="
2027             alert" aria-hidden="true">&times;</button>
2028         Usuario eliminado correctamente.
2029     </div>
2030     ';
2031 }
2032 else {
2033     echo '
2034     <div class="alert alert-danger alert-dismissible" align
2035         ="center">
2036     <button type="button" class="close" data-dismiss="alert
2037         " aria-hidden="true">&times;</button>
2038     Error en la base de datos. Tome nota del error y
2039     contacte con el administrador del sistema. Algun
2040     valor introducido en los datos no es correcto: '.
2041     mysql_error();
2042     echo '</div>
2043     ';
2044 }
2045 }
2046 }
2047 }
2048 // Listar las copias de seguridad existentes
2049 else if ($action == "listacopia") {
2050     echo '<h1 class="page-header">Listado de copias de seguridad</
2051     h1>
2052     <br><br>
2053     <div class="col-lg-12">
2054     <br>
2055     <div class="col-lg-10">
2056     <div class="panel panel-default">
2057     <div class="panel-heading">
2058         Copias de seguridad almacenados en el
2059         sistema
2060     </div>
2061     <!-- /.panel-heading -->
2062     <div class="panel-body">
2063     <br>
2064     <div class="form-group" style="margin-left
2065         :40px;">';
2066     $directorio = opendir("database/
2067         sqlBackups/"); // Ruta directorio
2068     while ($archivo = readdir($directorio))
2069     {
2070         // Obtenemos un archivo y luego
2071         otro sucesivamente
2072         if (!is_dir($archivo)) {
2073             if ($archivo[0] != '.' && $archivo !=
2074                 'readme.txt') {
2075                 echo '
2076                 <li class="dropdown">

```



```

2110         echo '
2111         <div class="form-group">
2112             <form action="accesoAdmin.php?action=
                altacopia" method="post">
2113                 <br>
2114                 <div class="form-group">
2115                     <label>Número de tablas en
                        la base de datos:</label>
2116                     <input class="form-control" type="
                        text" value="'.sizeof($tables).
                        '" disabled>
2117                 </div>
2118                 <div class="form-group">
2119                     <label>Tablas almacenadas en la
                        base de datos:</label>
2120                     <select multiple class="form-
                        control" disabled>';
2121                     foreach ($tables as $key =>
                        $value) {
2122                         echo '<option>'.$value.'</
                            option>';
2123                     }
2124                     echo '
2125                 </select>
2126             </div>
2127             <div class="form-group">
2128                 <label>Número de entradas
                        totales en la base de datos:</
                        label>
2129                 <input class="form-control" type="
                        text" value="'. $totalNumRows. '"
                        disabled>
2130             </div>
2131             <br>
2132             <div align="right">
2133                 <input type="hidden" name="backupOK
                        " value="backupOK">
2134                 <button type="submit" class="btn
                        btn-default" style="background-
                        color:green; color:white;">
                        Crear copia de seguridad</
                        button>
2135             </div>
2136             </form>
2137         </div>
2138     </div>
2139     <!-- /.table-responsive -->
2140 </div>
2141 <!-- /.panel-body -->
2142 </div>
2143 ';
2144 }
2145 else {
2146     // Creamos la copia de seguridad

```

```

2147 $reg_date = date("Y-m-d");
2148 $reg_time = date("H-i-s");
2149 $doBackup = doBackup($reg_date,$reg_time);
2150 if($doBackup){
2151     echo '
2152     <div class="alert alert-success alert-dismissable"
2153         align="center">
2154         <button type="button" class="close" data-dismiss="
2155             alert" aria-hidden="true">&times;</button>
2156         <p>Copia de seguridad <strong>[ '.$reg_date.'_'.'
2157             $reg_time.' '.zip ]</strong> creada correctamente
2158         .</p>
2159         <p><a href="database/sqlBackups/'.'.$reg_date.'_'.'
2160             $reg_time.' '.zip'.'" download>Pinche aqu&iacute;
2161             para descargarla.</a></p>
2162     </div>
2163     ';
2164 }
2165 else{
2166     echo '
2167     <div class="alert alert-danger alert-dismissable" align
2168         ="center">
2169     <button type="button" class="close" data-dismiss="alert
2170         " aria-hidden="true">&times;</button>
2171     Error creando la copia de seguridad. Tome nota del
2172     error y contacte con el administrador del sistema.'
2173     ;
2174     echo '</div>
2175     ';
2176 }
2177 echo '
2178 <!-- /.panel -->
2179 </div>
2180 ';
2181 }
2182
2183 // Eliminar copia de seguridad
2184 else if($action == 'eliminacopia'){
2185     echo '<h1 class="page-header">Eliminar copia de seguridad</h1>
2186     <br><br>
2187     <div class="col-lg-12">
2188     ';
2189     if(!isset($_POST['selectedCopia'])||$_POST['selectedCopia']=="")
2190     ){
2191         echo '
2192         <br><br>
2193         <div class="col-lg-10">
2194         <div class="panel panel-default">
2195             <div class="panel-heading">
2196                 BackUPs del sistema
2197             </div>
2198             <!-- /.panel-heading -->
2199             <div class="panel-body">

```



```

2190         <br>
2191         <div class="form-group">
2192             <label>Selecciona una copia de
2193                 seguridad:</label>
2194             <form action="accesoAdmin.php?action=
2195                 eliminacopia" method="post">
2196                 <select class="form-control" name="
2197                     selectedCopia">
2198                     <option value="" selected="selected"></
2199                         option>';
2200                 $directorio = opendir("database/
2201                     sqlBackups/"); //ruta directorio
2202                 while ($archivo = readdir($directorio))
2203                 {
2204                     //obtenemos un archivo y luego otro
2205                     sucesivamente
2206                     if (!is_dir($archivo)){
2207                         if ($archivo[0] != '.' && $archivo !=
2208                             'readme.txt'){
2209                             echo '<option value="'.
2210                                 $archivo.'"'>'. $archivo
2211                                 .'</option></a>';
2212                         }
2213                     }
2214                 }
2215                 echo '
2216             </select>
2217             <div align="right">
2218                 <br>
2219                 <button type="submit" class="btn
2220                     btn-default" style="background-
2221                         color:red; color:white;">
2222                     Eliminar copia de seguridad</
2223                         button>
2224             </div>
2225         </form>
2226     </div>
2227     <!-- /.table-responsive -->
2228 </div>
2229 <!-- /.panel-body -->
2230 </div>
2231 ';
2232 }
2233 else{
2234     $copia = $_POST[ 'selectedCopia ' ];
2235     try{
2236         unlink("database/sqlBackups/" . $copia);
2237         echo '
2238         <div class="alert alert-success alert-dismissable"
2239             align="center">
2240             <button type="button" class="close" data-dismiss="
2241                 alert" aria-hidden="true">&times;</button>
2242             Copia de seguridad eliminada correctamente.

```

```
2228         </div>
2229         ';
2230     }
2231     catch(Exception $e){
2232         echo '
2233         <div class="alert alert-danger alert-dismissible" align
2234         ="center">
2235         <button type="button" class="close" data-dismiss="alert
2236         " aria-hidden="true">&times;</button>
2237         Error en la base de datos. Tome nota del error y
2238         contacte con el administrador del sistema. Algun
2239         valor introducido en los datos no es correcto: '.$e
2240         ;
2241         echo '</div>
2242         ';
2243     }
2244 }
2245
2246 ///////////////////////////////////////////////////////////////////
2247 // Fin de las acciones de usuario //
2248 ///////////////////////////////////////////////////////////////////
2249
2250 }
2251
2252     echo '
2253         </div>
2254         <!-- /.col-lg-12 -->
2255     </div>
2256     <!-- /.row -->
2257 </div>
2258 <!-- /.container-fluid -->
2259 </div>
2260 <!-- /#page-wrapper -->
2261
2262 </div>
2263 <!-- /#wrapper -->
2264
2265 <!-- jQuery -->
2266 <script src="bower_components/jquery/dist/jquery.min.js"></script>
2267
2268 <!-- Bootstrap Core JavaScript -->
2269 <script src="bower_components/bootstrap/dist/js/bootstrap.min.js
2270     "></script>
2271
2272 <!-- Metis Menu Plugin JavaScript -->
2273 <script src="bower_components/metisMenu/dist/metisMenu.min.js"></
2274     script>
```

```
2275 <script src="dist/js/sb-admin-2.js"></script>
2276
2277 <!-- DataTables JavaScript -->
2278 <script src="bower_components/datatables/media/js/jquery.dataTables
2279 .min.js"></script>
2280 <script src="bower_components/datatables-plugins/integration/
2281 bootstrap/3/dataTables.bootstrap.min.js"></script>
2282
2283 <!-- Page-Level Demo Scripts - Tables - Use for reference -->
2284 <script>
2285 $(document).ready(function() {
2286     $("#dataTables-example").DataTable({
2287         responsive: true
2288     });
2289 });
2290 </script>
2291 </body>
2292 </html>';
2293
2294 ?>
```

APÉNDICE B

Códigos de la aplicación móvil

B.1 Página de inicio de la App

Este código pertenece a la página de inicio de la aplicación móvil multiplataforma desarrollada en C# mediante el IDE Xamarin Studio. En ella podremos introducir un nombre de usuario y contraseña para ser validado en la base de datos del servidor.

```
1  ¡»¿
2  //////////////////////////////////////////////////
3  // Homepage.cs //
4  //////////////////////////////////////////////////
5
6  using System;
7  using Xamarin.Forms;
8  using System.Json;
9  using System.IO;
10 using System.Net;
11 using System.Threading.Tasks;
12 using System.Net.NetworkInformation;
13
14 namespace SISERVER
15 {
16     public class HomePage : ContentPage
17     {
18
19         // Variable para la introduccion del nombre de usuario
20         Entry username = new Entry
21         {
22             Placeholder = "Nombre de usuario",
23         };
24
25         // Variable para la introduccion de la contraseña de usuario
26         Entry password = new Entry
27         {
28             Placeholder = "Contraseña",
```

```
29     IsPassword = true
30 };
31
32 public HomePage()
33 {
34
35     // Titulo de la pagina
36     this.Title = "SISERVER";
37
38     // Titulo del home page
39     var title = new Label
40     {
41         Text = "Inicio de sesion",
42         FontSize = Device.GetNamedSize(NamedSize.Large, typeof(Label)),
43         HorizontalOptions = LayoutOptions.CenterAndExpand,
44     };
45
46     // Boton de info
47     var aboutButton = new Button
48     {
49         Text = "App Info"
50     };
51
52     // Evento del boton de info
53     aboutButton.Clicked += (object sender, EventArgs e) =>
54     {
55         Navigation.PushAsync(new AboutPage());
56     };
57
58     // Boton de login
59     var login = new Button
60     {
61         Text = "Entrar"
62     };
63
64     // Evento de pulsacion del boton de login
65     login.Clicked += async (object sender, EventArgs e) =>
66     {
67         try
68         {
69             if (!string.IsNullOrWhiteSpace(username.Text) && !string.
70                 IsNullOrEmpty(password.Text))
71             {
72                 // Creamos un query para la validacion del usuario
73                 string url = "http://192.168.1.200:200/auth?usr=" +
74                     username.Text + "&pwd=" + password.Text;
75                 JsonValue result = await AuthAsync(url, "POST");
76                 if (result == true)
77                 {
78                     // Accedemos a la seccion de listado de clientes
79                     await Navigation.PushAsync(new ClientPage());
80                 }
81             }
82             else
83             {
84                 // Mensaje de error
85             }
86         }
87         catch { }
88     }
89 }
```

```
81         // Mensaje de error: datos de usuario no validos
82         await DisplayAlert("Error", "Datos de usuario no válidos
83         ", "OK");
84     }
85     else
86     {
87         // Mensaje de error: datos no introducidos
88         await DisplayAlert("Error", "Debe introducir un nombre de
89         usuario y una contraseña", "OK");
90     }
91     catch (Exception ex)
92     {
93         // Mensaje de error: runtime error
94         await DisplayAlert("Error", "El servidor no se encuentra
95         disponible: "+ex.Message, "OK");
96     }
97 };
98 // Logo del homepage
99 var img = new Image {
100     Aspect = Aspect.AspectFit,
101     WidthRequest = 80,
102     HeightRequest = 80,
103     TranslationY = -20,
104     Source = "logoApp.jpg"
105 };
106
107 // Seccion intermedia de la pantalla
108 var stackLayout1 = new StackLayout
109 {
110     Padding = 30,
111     Spacing = 10,
112     VerticalOptions = LayoutOptions.StartAndExpand,
113     Children = { img, title, username, password, login }
114 };
115
116 // Seccion inferior de la pantalla
117 var stackLayout2 = new StackLayout
118 {
119     Spacing = 10,
120     VerticalOptions = LayoutOptions.EndAndExpand,
121     Children = { aboutButton }
122 };
123
124 // Distribucion de elementos en la pagina
125 Content = new ScrollView
126 {
127     Content = new StackLayout
128     {
129         Padding = 30,
130         Spacing = 10,
131         Children = { stackLayout1, stackLayout2 }
```

```
132     }
133     };
134 }
135
136 // Funcion ejecutada al cargar la pagina
137 protected override async void OnAppearing()
138 {
139     // Reseteamos las variables de login
140     username.Text="";
141     password.Text="";
142 }
143
144 // Funcion asincrona para realizar la validacion de usuarios
145 private async Task<JsonValue> AuthAsync(string url, string method)
146 {
147     try
148     {
149         HttpWebRequest request = (HttpWebRequest)HttpWebRequest.Create(
150             new Uri(url));
151         request.ContentType = "application/json";
152         request.Method = method;
153         using (WebResponse response = await request.GetResponseAsync())
154         {
155             using (Stream stream = response.GetResponseStream())
156             {
157                 JsonValue jsonDoc = await Task.Run(() => JsonObject.Load(
158                     stream));
159                 Console.Out.WriteLine("Response: {0}", jsonDoc.ToString());
160                 return jsonDoc;
161             }
162         }
163     }
164     catch (Exception ex) { return ex.Message; }
165 }
166 }
```

B.2 Página de información de la App

Este código pertenece a la página de información de la aplicación móvil multiplataforma desarrollada en C# mediante el IDE Xamarin Studio. En ella podremos visualizar una pequeña descripción de la misma así como la información de contacto de sus desarrolladores.

```
1  ¡»¿
2  //////////////////////////////////
3  // AboutPage.cs //
4  //////////////////////////////////
5  
```



```
6 using System;
7 using System.Collections.Generic;
8 using System.Net.NetworkInformation;
9 using Xamarin.Forms;
10
11 namespace SISERVER
12 {
13     public class AboutPage : ContentPage
14     {
15         public AboutPage()
16         {
17             // Titulo de la pagina
18             this.Title = "App Info";
19
20             // Titulo de la pagina
21             var title = new Label
22             {
23                 Text = "SISERVER APP",
24                 FontSize = Device.GetNamedSize(NamedSize.Large, typeof(Label)),
25                 VerticalOptions = LayoutOptions.Center,
26                 HorizontalOptions = LayoutOptions.CenterAndExpand,
27             };
28
29             // Seccion de descripcion
30             var description = new Label
31             {
32                 Text = "Aplicacion movil SISERVER perteneciente a la empresa
33                     Embalajes Industriales Ramirez desarrollada durante el
34                     Trabajo Fin de Master del Master Universitario en
35                     Ingenieria Informatica de la Universidad Polit cnica de
36                     Valencia 'Diseno e implementacion de un Sistema de
37                     Informacion para una PYME' por Alvaro Roig Coves, y
38                     tutelado por Xavier Molero Prieto.",
39                 TextColor = Color.Gray,
40                 HorizontalTextAlignment = TextAlignment.Center,
41                 FontSize = Device.GetNamedSize(NamedSize.Small, typeof(Label))
42             };
43
44             // Autor
45             var autor = new Label
46             {
47                 Text = "Alvaro Roig Coves",
48                 FontSize = Device.GetNamedSize(NamedSize.Medium, typeof(Label))
49                 ,
50                 VerticalOptions = LayoutOptions.Center,
51                 HorizontalOptions = LayoutOptions.CenterAndExpand
52             };
53
54             // Email
55             var email = new Label
56             {
57                 Text = "alroico@inf.upv.es",
58                 FontSize = Device.GetNamedSize(NamedSize.Medium, typeof(Label))
59             }
60         }
61     }
62 }
```

```
52         VerticalOptions = LayoutOptions.Center ,
53         HorizontalOptions = LayoutOptions.CenterAndExpand
54     };
55
56     // Fecha
57     var fecha = new Label
58     {
59         Text = "08/05/2017" ,
60         FontSize = Device.GetNamedSize(NamedSize.Medium, typeof(Label))
61     } ,
62     VerticalOptions = LayoutOptions.Center ,
63     HorizontalOptions = LayoutOptions.CenterAndExpand
64 };
65
66     // Centro
67     var upv = new Label
68     {
69         Text = "MUIINF UPV" ,
70         FontSize = Device.GetNamedSize(NamedSize.Medium, typeof(Label))
71     } ,
72     VerticalOptions = LayoutOptions.Center ,
73     HorizontalOptions = LayoutOptions.CenterAndExpand
74 };
75
76     // Imagen Empresa
77     var img1 = new Image
78     {
79         Aspect = Aspect.AspectFit ,
80         WidthRequest = 50 ,
81         HeightRequest = 50 ,
82         Source = "logoEmpresa.jpg" ,
83         VerticalOptions = LayoutOptions.EndAndExpand
84     };
85
86     // Imagen UPV
87     var img2 = new Image
88     {
89         Aspect = Aspect.AspectFit ,
90         WidthRequest = 50 ,
91         HeightRequest = 50 ,
92         Source = "logoUPV.jpg" ,
93         VerticalOptions = LayoutOptions.EndAndExpand
94     };
95
96     // Seccion superior de la pantalla
97     var stackLayout1 = new StackLayout
98     {
99         Padding = 10 ,
100        Spacing = 10 ,
101        VerticalOptions = LayoutOptions.StartAndExpand ,
102        Children = { title , description , autor , email , fecha , upv }
103    };
104
105     // Seccion inferior de la pantalla
```

```
104     var stackLayout2 = new StackLayout
105     {
106         Padding = 10,
107         Spacing = 20,
108         VerticalOptions = LayoutOptions.CenterAndExpand,
109         Children = { img2, img1 }
110     };
111
112     // Distribucion de elementos en la pagina
113     Content = new ScrollView
114     {
115         Content = new StackLayout
116         {
117             Padding = 10,
118             Spacing = 10,
119             Children = { stackLayout1, stackLayout2 }
120         }
121     };
122
123 }
124 }
125 }
```

B.3 Página de listado de clientes de la App

Este código pertenece a la página de listado de clientes de la aplicación móvil multiplataforma desarrollada en C# mediante el IDE Xamarin Studio. En ella podremos visualizar el listado de clientes registrados en la base de datos del servidor.

```
1  ¡»¿
2  //////////////////////////////////
3  // ClientPage.cs //
4  //////////////////////////////////
5
6  using System;
7  using System.Collections.Generic;
8  using System.IO;
9  using System.Json;
10 using System.Net;
11 using System.Threading.Tasks;
12 using Newtonsoft.Json;
13 using Newtonsoft.Json.Linq;
14 using Xamarin.Forms;
15
16 namespace SISERVER
17 {
18     public class ClientPage : ContentPage
19     {
20
```

```
21 // Variable para el listado de clientes
22 List<Cliente> clientes = new List<Cliente>();
23
24 // Funcion ejecutada al cargar la pagina
25 protected override async void OnAppearing()
26 {
27     // Limpiamos el array de clientes para prevenir duplicados ante
28     // recargas de pagina
29     clientes.Clear();
30
31     try
32     {
33         base.OnAppearing();
34         // Obtenemos el listado de clientes
35         string url = "http://192.168.1.200:200/listClients";
36         JsonValue result = await GetClientsAsync(url, "POST");
37         // PARSE JSON //
38         foreach (var item in result)
39         {
40             string str = item.ToString().Remove(0, 1);
41             str = str.Remove(str.Length - 1);
42             string [] substrings1 = str.Split(',');
43             string [] substringsID = substrings1[0].Split(':');
44             string [] substringsName = substrings1[1].Split(':');
45             int formattedId = Int32.Parse(substringsID[1].Trim());
46             string formattedName = substringsName[1].Trim().Remove(0, 1);
47             formattedName = formattedName.Remove(formattedName.Length - 1);
48             clientes.Add(new Cliente(formattedId, formattedName));
49         }
50
51         // Titulo de la pagina
52         this.Title = "Clientes";
53
54         // Titulo cabecera
55         var title = new Label
56         {
57             Text = "Seleccione un cliente",
58             FontSize = Device.GetNamedSize(NamedSize.Large, typeof(Label)
59             ),
60             VerticalOptions = LayoutOptions.Center,
61             HorizontalOptions = LayoutOptions.CenterAndExpand,
62         };
63
64         // Variable para el listado de nombres de clientes
65         List<string> nombreClientes = new List<string>();
66
67         // Preparamos el listado de clientes por su nombre
68         foreach (Cliente c in clientes)
69         {
70             nombreClientes.Add(c.nombre);
71         }
72
73         // Listview de los clientes
74         ListView clientesView = new ListView
```

```
73     {
74         ItemsSource = nombreClientes
75     };
76
77     // Evento para cada entrada
78     clientesView.ItemSelected += async(sender, e) => {
79         if (e.SelectedItem == null) return;
80         else
81         {
82             // Obtenemos el identificador del cliente seleccionado
83             string clienteID = "0";
84             string clienteName = "";
85             foreach (Cliente c in clientes)
86             {
87                 if (e.SelectedItem.ToString() == c.nombre)
88                 {
89                     clienteID = c.id.ToString();
90                     clienteName = c.nombre.ToString();
91                 }
92             }
93             // Accedemos a la seccion de listado de entradas del
94             // cliente
95             await Navigation.PushAsync(new ClientEntryPage(clienteID,
96                 clienteName));
97         }
98     };
99
100     // Seccion superior de la pantalla
101     var stackLayout1 = new StackLayout
102     {
103         Padding = 10,
104         Spacing = 10,
105         VerticalOptions = LayoutOptions.StartAndExpand,
106         Children = { title }
107     };
108
109     // Seccion inferior de la pantalla
110     var stackLayout2 = new StackLayout
111     {
112         Padding = 10,
113         Spacing = 20,
114         VerticalOptions = LayoutOptions.CenterAndExpand,
115         Children = { clientesView }
116     };
117
118     // Distribucion de elementos en la pagina
119     Content = new ScrollView
120     {
121         Content = new StackLayout
122         {
123             Padding = 10,
124             Spacing = 10,
125             Children = { stackLayout1, stackLayout2 }
126         }
127     }
```

```
125     };
126   }
127   catch (Exception ex) {
128     Console.WriteLine(ex.Message);
129     // Mensaje de error: datos de usuario no validos
130     await DisplayAlert("Error", ex.Message, "OK");
131   }
132 }
133
134 // Constructor de la clase
135 public ClientPage() {}
136
137 // Funcion asincrona para obtener el listado de clientes
138 private async Task<JsonValue> GetClientsAsync(string url, string
139   method)
140 {
141   try
142   {
143     HttpWebRequest request = (HttpWebRequest)HttpWebRequest.Create(
144       new Uri(url));
145     request.ContentType = "application/json";
146     request.Method = method;
147     using (WebResponse response = await request.GetResponseAsync())
148     {
149       using (Stream stream = response.GetResponseStream())
150       {
151         JsonValue jsonDoc = await Task.Run(() => JsonObject.Load(
152           stream));
153         Console.Out.WriteLine("Response: {0}", jsonDoc.ToString());
154         return jsonDoc;
155       }
156     }
157   }
158   catch (Exception ex) { return ex.Message; }
159 }
160
161 // Clase del objeto Cliente
162 public class Cliente
163 {
164   public int id { get; set; }
165   public string nombre { get; set; }
166
167   public Cliente(int id, string nombre)
168   {
169     this.id = id;
170     this.nombre = nombre;
171   }
172 }
```

B.4 Página de listado de productos de un cliente de la App

Este código pertenece a la página de listado de productos asociados a un cliente de la aplicación móvil multiplataforma desarrollada en C# mediante el IDE Xamarin Studio. En ella podremos visualizar el listado de todas las entradas de productos asociadas a un cliente dado que hayan sido registradas en la base de datos del servidor.

```
1  i»¿
2  //////////////////////////////////////
3  // ClientEntryPage.cs //
4  //////////////////////////////////////
5
6  using System;
7  using System.Collections.Generic;
8  using System.IO;
9  using System.Json;
10 using System.Net;
11 using System.Threading.Tasks;
12 using Newtonsoft.Json;
13 using Xamarin.Forms;
14
15 namespace SISERVER
16 {
17     public class ClientEntryPage : ContentPage
18     {
19
20         // Variable para el listado de entradas
21         List<Entry> entradas = new List<Entry>();
22
23         // Variable para el listado de nombres de clientes
24         List<string> nombreEntrys = new List<string>();
25         List<string> nombreEntrysAux = new List<string>();
26
27         // Variable para la barra de filtrado
28         SearchBar searchBar;
29         int itemCount = 0;
30
31         // Variable de clase para el identificador del cliente
32         public string clientID { get; set; }
33         public string clientName { get; set; }
34
35         // Constructor de la clase
36         public ClientEntryPage(string clientID, string name)
37         {
38             this.clientID = clientID;
39             this.clientName = name;
40         }
41
42         // Funcion ejecutada al cargar la pagina
```

```
43 protected override async void OnAppearing()
44 {
45     try
46     {
47         // Limpiamos el array de entradas para prevenir duplicados ante
48         // recargas de pagina
49         entradas.Clear();
50         nombreEntrys.Clear();
51         nombreEntrysAux.Clear();
52
53         base.OnAppearing();
54         // Obtenemos el listado de clientes
55         string url = "http://192.168.1.200:200/listEntry?clientID="+
56             this.clientID;
57         JsonValue result = await GetClientsAsync(url, "POST");
58         // PARSE JSON //
59         foreach (var item in result)
60         {
61             //Console.WriteLine(item);
62             string str = item.ToString().Remove(0, 1);
63             str = str.Remove(str.Length - 1);
64             //Console.WriteLine(str);
65             string [] substrings = str.Split(',');
66             // ID
67             string [] substringsID = substrings[1].Split(':');
68             string formattedID = substringsID[1].Trim().Remove(0, 1);
69             formattedID = formattedID.Remove(formattedID.Length - 1);
70             // Nombre
71             string [] substringsNombre = substrings[6].Split(':');
72             string formattedNombre = substringsNombre[1].Trim().Remove(0,
73                 1);
74             formattedNombre = formattedNombre.Remove(formattedNombre.Length
75                 - 1);
76             // Tipo
77             string [] substringsTipo = substrings[7].Split(':');
78             string formattedTipo = substringsTipo[1].Trim().Remove(0, 1);
79             formattedTipo = formattedTipo.Remove(formattedTipo.Length - 1);
80             // Medidas
81             string [] substringsMedidas = substrings[5].Split(':');
82             string formattedMedidas = substringsMedidas[1].Trim().Remove
83                 (0, 1);
84             formattedMedidas = formattedMedidas.Remove(formattedMedidas.
85                 Length - 1);
86             // Descripcion
87             string [] substringsDescripcion = substrings[0].Split(':');
88             string formattedDesc = substringsDescripcion[1].Trim().Remove
89                 (0, 1);
90             formattedDesc = formattedDesc.Remove(formattedDesc.Length - 1);
91             // Imagen 1
92             string [] substringsImagen1 = substrings[2].Split(':');
93             string formattedImg1 = "-";
94             if (substringsImagen1[1].Trim() != "null")
95             {
96                 formattedImg1 = substringsImagen1[1].Trim().Remove(0, 1);
97             }
98         }
99     }
100 }
```



```
90         formattedImg1 = formattedImg1.Remove(formattedImg1.Length - 1)
91         ;
92     }
93     // Imagen 2
94     string [] substringsImagen2 = substrings [3].Split(':');
95     string formattedImg2 = "-";
96     if (substringsImagen2 [1].Trim () != "null")
97     {
98         formattedImg2 = substringsImagen2 [1].Trim ().Remove(0, 1);
99         formattedImg2 = formattedImg2.Remove(formattedImg2.Length - 1)
100        ;
101    }
102    // Imagen 3
103    string [] substringsImagen3 = substrings [4].Split(':');
104    string formattedImg3 = "-";
105    if (substringsImagen3 [1].Trim () != "null")
106    {
107        formattedImg3 = substringsImagen3 [1].Trim ().Remove(0, 1);
108        formattedImg3 = formattedImg3.Remove(formattedImg3.Length - 1)
109        ;
110    }
111    // NEW ENTRY
112    entradas.Add(new Entry (formattedID, formattedNombre,
113        formattedTipo, formattedMedidas, formattedDesc, formattedImg1
114        , formattedImg2, formattedImg3));
115 }
116
117 // Titulo de la pagina
118 this.Title = this.clientName;
119
120 // Titulo cabecera
121 var title = new Label
122 {
123     Text = "Seleccione una entrada",
124     FontSize = Device.GetNamedSize(NamedSize.Large, typeof(Label)
125     ),
126     VerticalOptions = LayoutOptions.Center,
127     HorizontalOptions = LayoutOptions.CenterAndExpand,
128 };
129
130 // Preparamos el listado de clientes por su nombre
131 foreach (Entry e in entradas)
132 {
133     nombreEntrys.Add(e.id+" | "+e.tipo+" | "+e.nombre);
134     itemCount++;
135 }
136
137 // Listview de los clientes
138 ListView entradasView = new ListView
139 {
140     ItemsSource = nombreEntrys
141 };
142
143 // Evento para cada entrada
```

```
138     entradasView.ItemSelected += async (sender, e) =>
139     {
140         if (e.SelectedItem == null) return;
141         else
142         {
143
144             string selectedItem = e.SelectedItem.ToString();
145             string[] selectSubstrings = selectedItem.Split('|');
146             string selectedID = selectSubstrings[0].Trim();
147             Entry selectedEntry = null;
148             foreach (Entry item in entradas)
149             {
150                 if (item.id == selectedID) { selectedEntry = item; }
151             }
152             if (selectedEntry == null)
153             {
154                 // Mensaje de error: datos de entrada no validos
155                 await DisplayAlert("Error", "No se encuentra la entrada
156                     con ID=" + selectedID, "OK");
157             }
158             else
159             {
160                 //Console.WriteLine(selectedEntry.toString());
161                 // Accedemos a la seccion de muestra de entrada
162                 await Navigation.PushAsync(new EntryPage(selectedEntry.
163                     toString()));
164             }
165         }
166     };
167
168     // Barra de filtrado de entradas
169     searchBar = new SearchBar {
170         Placeholder = "Buscar una entrada"
171     };
172     searchBar.TextChanged += (sender, args) => {
173         if (searchBar.Text == "" || String.IsNullOrEmpty(searchBar.
174             Text))
175         {
176             nombreEntrysAux.Clear();
177             entradasView.ItemsSource = nombreEntrys;
178         }
179         else
180         {
181             foreach (string item in nombreEntrys)
182             {
183                 string[] substr = item.Split('|');
184                 if (substr[0].Trim().Contains(searchBar.Text))
185                 {
186                     nombreEntrysAux.Add(item);
187                 }
188             }
189             entradasView.ItemsSource = nombreEntrysAux;

```

```
189     }
190     };
191
192     // Seccion superior de la pantalla
193     var stackLayout1 = new StackLayout
194     {
195         Padding = 10,
196         Spacing = 10,
197         VerticalOptions = LayoutOptions.StartAndExpand,
198         Children = { title, searchBar }
199     };
200
201     // Seccion inferior de la pantalla
202     var stackLayout2 = new StackLayout
203     {
204         Padding = 10,
205         Spacing = 20,
206         VerticalOptions = LayoutOptions.CenterAndExpand,
207         Children = { entradasView }
208     };
209
210     // Distribucion de elementos en la pagina
211     //Content = new ScrollView
212     //{
213     Content = new StackLayout
214     {
215         Padding = 10,
216         Spacing = 10,
217         Children = { stackLayout1, stackLayout2 }
218     };
219     //};
220 }
221 catch (Exception ex)
222 {
223     Console.WriteLine(ex.Message);
224     // Mensaje de error: datos de usuario no validos
225     await DisplayAlert("Error", ex.Message, "OK");
226 }
227 }
228
229 // Funcion asincrona para obtener el listado de clientes
230 private async Task<JsonValue> GetClientsAsync(string url, string
method)
231 {
232     try
233     {
234         HttpWebRequest request = (HttpWebRequest)HttpWebRequest.Create(
new Uri(url));
235         request.ContentType = "application/json";
236         request.Method = method;
237         using (WebResponse response = await request.GetResponseAsync())
238         {
239             using (Stream stream = response.GetResponseStream())
240             {
```

```
241         JsonValue jsonDoc = await Task.Run(() => JsonObject.Load(
242             stream));
243         //Console.Out.WriteLine("Response: {0}", jsonDoc.ToString()
244             );
245         return jsonDoc;
246     }
247 }
248 }
249
250 // Clase del objeto EntryList
251 public class Entry
252 {
253     public string id { get; set; }
254     public string nombre { get; set; }
255     public string tipo { get; set; }
256     public string medidas { get; set; }
257     public string descripcion { get; set; }
258     public string imagen1 { get; set; }
259     public string imagen2 { get; set; }
260     public string imagen3 { get; set; }
261
262     public Entry(string id, string nombre, string tipo, string
263         medidas, string descripcion, string imagen1, string imagen2,
264         string imagen3)
265     {
266         this.id = id;
267         this.nombre = nombre;
268         this.tipo = tipo;
269         this.medidas = medidas;
270         this.descripcion = descripcion;
271         this.imagen1 = imagen1;
272         this.imagen2 = imagen2;
273         this.imagen3 = imagen3;
274     }
275
276     public string toString()
277     {
278         return "id:" + this.id + ",tipo:" + this.tipo + ",nombre:" +
279             this.nombre + ",medidas:" + this.medidas + ",descripcion:"
280             + this.descripcion + ",imagen1:" + this.imagen1 + ",imagen2
281             : " + this.imagen2 + ",imagen3:" + this.imagen3;
282     }
283 }
284 }
```

B.5 Página de información de un producto de la App

Este código pertenece a la página de muestra de información de un producto asociado de la aplicación móvil multiplataforma desarrollada en C# mediante el IDE Xamarin Studio. En ella podremos visualizar la información asociada a una entrada de producto presente en la base de datos.

```
1  ¡»¿
2  //////////////////////////////////
3  //  EntryPage.cs  //
4  //////////////////////////////////
5
6  using System;
7  using System.Collections.Generic;
8  using System.IO;
9  using Xamarin.Forms;
10
11 namespace SISERVER
12 {
13     public class EntryPage : ContentPage
14     {
15         // Variables de la entrada de base de datos
16         public string id { get; set; }
17         public string tipo { get; set; }
18         public string nombre { get; set; }
19         public string medidas { get; set; }
20         public string descripcion { get; set; }
21         public string imagen1 { get; set; }
22         public string imagen2 { get; set; }
23         public string imagen3 { get; set; }
24
25         // Constructor de la clase
26         public EntryPage(string data)
27         {
28             // Parseamos el string recibido
29             List<String> items = new List<String>();
30             string[] substrings1 = data.Split(',');
31             foreach (string part in substrings1)
32             {
33                 string[] substrings2 = part.Split(':');
34                 items.Add(substrings2[1].Trim());
35             }
36             // Creamos el objeto
37             this.id = items[0];
38             this.tipo = items[1];
39                 this.nombre = items[2];
40                 this.medidas = items[3];
41                 this.descripcion = items[4];
42                 this.imagen1 = items[5];
43                 this.imagen2 = items[6];

```

```
44     this.imagen3 = items[7];
45 }
46
47 // Funcion ejecutada al cargar la pagina
48 protected override async void OnAppearing()
49 {
50     try
51     {
52         base.OnAppearing();
53
54         // Titulo de la pagina
55         this.Title = this.id;
56
57         // Titulo cabecera nombre
58         var tituloName = new Label
59         {
60             Text = "Nombre",
61             FontSize = Device.GetNamedSize(NamedSize.Large, typeof(Label)
62             ),
63         };
64         var name = new Label
65         {
66             Text = this.nombre.Replace(@"\r\n", Environment.NewLine),
67             FontSize = Device.GetNamedSize(NamedSize.Medium, typeof(Label)
68             )),
69         };
70
71         // Titulo cabecera tipo
72         var tituloType = new Label
73         {
74             Text = "Tipo",
75             FontSize = Device.GetNamedSize(NamedSize.Large, typeof(Label)
76             ),
77         };
78         var type = new Label
79         {
80             Text = this.tipo.Replace(@"\r\n", Environment.NewLine),
81             FontSize = Device.GetNamedSize(NamedSize.Medium, typeof(Label)
82             )),
83         };
84
85         // Titulo cabecera medidas
86         var tituloMeasures = new Label
87         {
88             Text = "Medidas",
89             FontSize = Device.GetNamedSize(NamedSize.Large, typeof(Label)
90             ),
91         };
92         var measures = new Label
93         {
94             Text = this.medidas.Replace(@"\r\n", Environment.NewLine),
95             FontSize = Device.GetNamedSize(NamedSize.Medium, typeof(Label)
96             )),
97         };
98     }
99     catch { }
100 }
```

```
92
93 // Titulo cabecera descripcion
94 var tituloDescription = new Label
95 {
96     Text = "Descripcion",
97     FontSize = Device.GetNamedSize(NamedSize.Large, typeof(Label)
98     ),
99 };
100 var description = new Label
101 {
102     Text = this.descripcion.Replace(@"\r\n", Environment.NewLine)
103     ,
104     FontSize = Device.GetNamedSize(NamedSize.Medium, typeof(Label)
105     ),
106 };
107 // Botones para la imagen/plano
108 var tituloImage = new Label
109 {
110     Text = "Imagen/Plano",
111     FontSize = Device.GetNamedSize(NamedSize.Large, typeof(Label)
112     ),
113 };
114 var buttonsList = new List<Button>();
115 var buttonLayout = new StackLayout();
116 // Imagen 1
117 string img1 = this.imagen1.Replace(@"\r\n", "");
118 if (img1 != "-")
119 {
120     Button b = new Button();
121     b.Text = img1;
122     b.Font = Font.SystemFontOfSize(NamedSize.Medium);
123     b.BorderWidth = 1;
124     b.Clicked += ButtonOnClick;
125     buttonsList.Add(b);
126 }
127 // Imagen 2
128 string img2 = this.imagen2.Replace(@"\r\n", "");
129 if (img2 != "-")
130 {
131     Button b = new Button();
132     b.Text = img2;
133     b.Font = Font.SystemFontOfSize(NamedSize.Medium);
134     b.BorderWidth = 1;
135     b.Clicked += ButtonOnClick;
136     buttonsList.Add(b);
137 }
138 // Imagen 3
139 string img3 = this.imagen3.Replace(@"\r\n", "");
140 if (img3 != "-")
141 {
142     Button b = new Button();
143     b.Text = img3;
144     b.Font = Font.SystemFontOfSize(NamedSize.Medium);
```

```
142         b.BorderWidth = 1;
143         b.Clicked += ButtonOnClick;
144         buttonsList.Add(b);
145     }
146     foreach (Button item in buttonsList)
147     {
148         buttonLayout.Children.Add(item);
149     }
150
151     // Secciones stack de la pantalla
152     var stackLayoutNombre = new StackLayout
153     {
154         Padding = 10,
155         Children = { tituloName, name }
156     };
157     var stackLayoutTipo = new StackLayout
158     {
159         Padding = 10,
160         Children = { tituloType, type }
161     };
162     var stackLayoutMedidas = new StackLayout
163     {
164         Padding = 10,
165         Children = { tituloMeasures, measures }
166     };
167     var stackLayoutDesc = new StackLayout
168     {
169         Padding = 10,
170         Children = { tituloDescription, description }
171     };
172     var stackLayoutImg = new StackLayout
173     {
174         Padding = 10,
175         Children = { tituloImage, buttonLayout }
176     };
177
178     // Distribucion de elementos en la pagina
179     Content = new ScrollView
180     {
181         Content = new StackLayout
182         {
183             Padding = 10,
184             Children = { stackLayoutNombre, stackLayoutTipo,
185                 stackLayoutMedidas, stackLayoutDesc, stackLayoutImg }
186         }
187     };
188     catch (Exception ex)
189     {
190         Console.WriteLine(ex.Message);
191         // Mensaje de error: datos de usuario no validos
192         await DisplayAlert("Error", ex.Message, "OK");
193     }
194 }
```



```
195
196 // Evento para abrir una imagen desde un boton
197 private void ButtonOnClick(object sender, EventArgs eventArgs)
198 {
199     var button = sender as Button;
200     Device.OpenUri(new Uri("http://192.168.1.200/" + button.Text));
201 }
202
203 }
204 }
```

APÉNDICE C

Códigos del API RESTful

C.1 Servicio RESTful en el servidor

Este código es el perteneciente al programa encargado de hacer las veces de interfaz RESTful, está escrito en Python y hace uso del *framework* Flask para atender peticiones en un puerto del servidor, consultar a la base de datos del mismo, y responder al cliente que llevó a cabo la consulta.

```
1 #!/usr/bin/env python
2
3 #####
4 ##### SISERVER RESTful API #####
5 #####
6 # restfulApi.py
7 # Api RESTful para la comunicacion con la app movil
8 # @auth alroico@inf.upv.es
9 # @date 08/04/2017
10
11 # Imports section
12 from flask import Flask, request
13 from flaskext.mysql import MySQL
14 from flask import json, jsonify
15 from flask import session
16 # -----
17
18 # App init definitio
19 app = Flask(__name__)
20 # -----
21
22 # Database Connection Data
23 mysql = MySQL()
24 app.config['MYSQL_DATABASE_USER'] = 'root'
25 app.config['MYSQL_DATABASE_PASSWORD'] = '*****'
26 app.config['MYSQL_DATABASE_DB'] = 'tfm'
27 app.config['MYSQL_DATABASE_HOST'] = 'localhost'
28 mysql.init_app(app)
```

```
29 # -----
30
31 # Home function
32 @app.route("/")
33 def hello():
34     info = {}
35     info["Title"] = "SISERVER RESTFul API"
36     info["Author"] = "alroico@inf.upv.es"
37     info["Date"] = "10/05/2017"
38     info["Options"] = {
39         "/": "Home",
40         "/clear": "Clear user session",
41         "/auth": "Authenticate user | @params( usr & pwd )",
42         "/listClients": "Get client list | Needs to be authenticated",
43         "/listEntry": "Get all entries from client | @params( clientID ) |
44             Needs to be authenticated"
45     }
46     return jsonify(info)
47 # -----
48 # Function to clear the session
49 @app.route('/clear')
50 def clear():
51     session.clear()
52     return "You are logged out."
53 # -----
54
55 # Authenticate function
56 @app.route("/auth", methods=['GET', 'POST'])
57 def auth():
58     if request.method == "POST":
59         username = request.args.get('usr')
60         password = request.args.get('pwd')
61         cursor = mysql.connect().cursor()
62         cursor.execute("SELECT * from usuarios_app where user='" + username
63             + "' and pass='" + password + "'")
64         data = cursor.fetchone()
65         if data is None:
66             return json.dumps(False)
67         else:
68             session['auth'] = '1'
69             return json.dumps(True)
70     else:
71         return "Just POST connections are allowed."
72 # -----
73
74 # List Clients Function
75 @app.route('/listClients', methods=['GET', 'POST'])
76 def listClients():
77     if request.method == "POST":
78         cursor = mysql.connect().cursor()
79         cursor.execute("SELECT * from clientes")
80         data = cursor.fetchall()
81         itemList = []
```

```
81     for item in data:
82         itemDict = {
83             'id': item[0],
84             'nombre': item[1]
85         }
86         itemList.append(itemDict)
87     return json.dumps(itemList)
88 else:
89     return "Just POST connections are allowed."
90 # -----
91
92
93 # List data function
94 @app.route('/listEntry', methods=["POST"])
95 def listEntry():
96     if request.method == "POST":
97         clientID = request.args.get('clientID')
98         cursor = mysql.connect().cursor()
99         cursor.execute("SELECT nombre from clientes where id='" + clientID
100                        + "'")
101         data = cursor.fetchone()
102         cursor.execute("SELECT * from " + data[0])
103         data = cursor.fetchall()
104         itemList = []
105         for item in data:
106             itemDict = {
107                 'id': item[0],
108                 'tipo': item[1],
109                 'nombre': item[2],
110                 'medidas': item[3],
111                 'descripcion': item[4],
112                 'imagen1': item[5],
113                 'imagen2': item[6],
114                 'imagen3': item[7]
115             }
116             itemList.append(itemDict)
117         return json.dumps(itemList)
118     else:
119         return "Just POST connections are allowed."
120 # -----
121
122 # Initialize function
123 if __name__ == "__main__":
124     app.secret_key = '*****'
125     app.run(host='0.0.0.0', port=200, debug=True)
126 # -----
```

APÉNDICE D

Formularios de valoración de los usuarios

D.1 Formulario de valoración general

El formulario presente en la Figura D.1 es el utilizado con el fin de obtener la valoración de los servicios y aplicaciones desplegadas por parte de los trabajadores de la empresa durante la fase de prueba del proyecto.



Formulario de valoración por parte de los usuarios

Mediante este pequeño cuestionario pretendemos obtener su valoración acerca del Sistema de Información implantado en su empresa durante el transcurso del Trabajo de Fin de Máster del Máster Universitario de Ingeniería Informática de la Universidad Politécnica de Valencia "Diseño e implementación de un Sistema de Información para una PYME". Así mismo le comunicamos que esta valoración junto con su nombre y rol dentro de la empresa serán incluidos en dicho trabajo con el fin de llevar a cabo un estudio de usabilidad del trabajo desarrollado.

Nombre:		
Rol en la empresa:		
Aplicaciones que ha utilizado:	<input type="checkbox"/> Aplicación Web de gestión	<input type="checkbox"/> Aplicación Móvil de consulta

Conteste a las siguientes preguntas marcando con una **X** la casilla de valoración que desee teniendo en cuenta el grado de conformidad con las mismas, en el que el **0** corresponde a "nada de acuerdo", y el **5** a "muy de acuerdo".

Pregunta	Valoración					
	0	1	2	3	4	5
¿Considera que el Sistema de Información (SI) desplegado le ayuda con sus labores diarias?						
¿La forma de trabajar con el SI le parece sencilla?						
¿El tiempo que tarda en consultar información en la base de datos es relativamente corto?						
¿El SI cumple con sus necesidades de obtención de información de productos?						
¿Las interfaces de usuario de las aplicaciones del SI le parecen apropiadas?						
¿Le parece sencillo navegar entre diferentes secciones de las aplicaciones del SI?						

	N/A	NO	SI
¿A detectado algún elemento del Sistema de Información que no sea correcto o se pueda mejorar?			

En caso afirmativo, por favor, indíquelo a continuación:

Figura D.1: Formulario de valoración de usuario