

## Sistema de Administración para el Telecontrol y Monitorización de Usuarios

**Nombre Estudiante:** Alejandro Cardo Grau

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

**Nombre Consultor/a:** Eduard Martin Lineros

**Profesor/a responsable de la asignatura:** Carles Garrigues Olivella

04/01/2017

© Alejandro Cardo Grau

**Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.**

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>Sistema de Administración para el Telecontrol y Monitorización de Usuarios</i>
<b>Nombre del autor:</b>	<i>Alejandro Cardo Grau</i>
<b>Nombre del consultor/a:</b>	<i>Eduard Martin Lineros</i>
<b>Nombre del PRA:</b>	<i>Carles Garrigues Olivella</i>
<b>Fecha de entrega (mm/aaaa):</b>	04/01/17
<b>Titulación:</b>	Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
<b>Idioma del trabajo:</b>	<i>Castellano</i>
<b>Palabras clave</b>	<i>phonegap, alarma, raspberry pi</i>
<b>Resumen del Trabajo (máximo 250 palabras):</b> <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i>	
<p>El objetivo del presente proyecto es el desarrollo y documentación de todas las fases del ciclo de vida de un sistema informático íntegro y específico orientado a la ayuda del control y gestión de presencia para las pequeñas y medianas empresas (PYMES) pertenecientes a cualquier sector que quiera mejorar su seguridad física.</p> <p>Gracias a la experimentación, el análisis y el estudio de las actividades relacionadas con los mecanismos y el control de la presencia de empleados podremos obtener un modelo de diseño genérico que nos permita enfocarlo a la implantación de un sistema informático de bajo coste que englobe una solución real, fiable y de utilidad para la monitorización.</p> <p>En el desarrollo de la aplicación se ha utilizado principalmente la herramienta de desarrollo rápido de aplicaciones: Android Studio para el desarrollo móvil, programación web, el framework multiplataforma Phonegap, responsive design para la adaptación multidispositivo, componentes del hardware libre, servicios de red y S.G.B.D., entre otros.</p>	

**Abstract (in English, 250 words or less):**

The objective of this project is the development and documentation of all phases of the lifecycle of a full and specific information system oriented control support and presence management for small and medium enterprises (SMEs) belonging to any sector you want improve their physical safety.

Through experimentation, analysis and study of the activities related to the mechanisms and control of the presence of employees we can get a sample of generic design that allows us to approach it with the implementation of a computerized system for low cost encompassing solution real, reliable and useful for monitoring.

In developing the application it has been used mainly Tool Rapid Application Development: Android Studio for mobile development, web programming, multiplatform framework Phonegap, responsive design for multidevice adaptation, free hardware components, network services and DBMS, among others.

## Índice

<b>Capítulo I: Introducción.....</b>	<b>12</b>
<b>1 INTRODUCCIÓN.....</b>	<b>13</b>
1.1 Motivación.....	13
1.2 Objetivos del Trabajo.....	13
1.3 Enfoque, Método Seguido y Breve Descripción de los Otros Capítulos.....	14
<b>2 ESTUDIO PRELIMINAR.....</b>	<b>15</b>
2.1 Descripción del Caso de Estudio.....	15
2.1.1 Objetivos.....	15
2.2 Descripción del Paradigma Actual.....	15
2.2.1 Estudio de alternativas de solución.....	16
2.3 Estimación Inicial del Coste del Proyecto.....	18
2.4 Planificación del Trabajo.....	18
2.4.1 Etapas del proyecto.....	18
2.4.2 Diagrama de Gantt.....	19
2.4.2.1 Hitos del proyecto.....	20
2.4.2.2 Descripción de etapas del proceso y sumario de artefactos obtenidos.....	21
2.4.2.3 Representación del diagrama de Gantt.....	23
2.5 Planificación de Costes.....	25
2.5.1 Recursos humanos.....	25
2.5.1.1 Estimación del coste inicial.....	27
2.5.2 Recursos software.....	28
2.5.3 Recursos hardware.....	29
2.6 Desarrollo por Microcontrolador.....	30
2.6.1 Control centralizado y monitorizado por microcontrolador.....	30
2.6.2 Características generales de las Raspberry Pi.....	31
2.6.3 Características específicas del modelo seleccionado.....	32
2.6.4 Módulos hardware integrados.....	33
<b>Capítulo II: Análisis de Requisitos.....</b>	<b>34</b>
<b>1 MODELADO DE PROCESOS DE NEGOCIO.....</b>	<b>35</b>
1.1 Proceso de Negocio para la Generación de Informes.....	35
<b>2 USUARIOS Y CONTEXTO DE USO.....</b>	<b>36</b>
2.1 Descripción del Capítulo.....	36
2.2 Participantes en el Proyecto.....	37
2.3 Actores del Sistema.....	37
2.4 Objetivos del Sistema.....	38
2.5 Requisitos Funcionales del Sistema.....	39
2.6 Requisitos No Funcionales del Sistema.....	45
2.7 Requisitos de Interfaz del Sistema.....	45
<b>Capítulo III: Diseño del Sistema.....</b>	<b>46</b>
<b>1 TRATAMIENTO DE DATOS PARA BBDD.....</b>	<b>47</b>

1.1 Diagrama entidad-Relación (E.R.D.).....	47
1.2 Traslado a Esquema Relacional y Normalización.....	47
1.3 Selección Justificada del S.G.B.D.....	48
<b>2 DISEÑO ARQUITECTÓNICO.....</b>	<b>49</b>
2.1 Definición de Arquitectura del Sistema.....	49
2.1.1 Descripción del Entorno Tecnológico.....	49
2.1.2 Esquema de Comunicaciones en Red.....	49
2.2 Arquitectura del Servidor.....	51
2.2.1 Tecnología Ajax.....	51
2.2.2 Servidor web Apache.....	51
2.2.3 Tecnología PHP.....	51
2.3 Componentes Software del Sistema.....	52
2.3.1 Configuración del Apache.....	52
2.3.2 Librerías del servidor.....	52
2.3.3 Configuración de MySQL.....	53
2.3.4 Librerías cliente.....	53
2.6.4.1 JQuery Mobile.....	53
2.6.4.2 Cripto-JS.....	54
2.6.4.3 ThemeRoller.....	54
2.6.4.4 Prototype Javascript.....	54
2.3.5 Integración multiplataforma móvil con Phonegap.....	55
2.3.6 Selección del Entorno de Desarrollo.....	56
<b>3 DISEÑO CONCEPTUAL.....</b>	<b>57</b>
3.1 Descripción del Diagrama de Flujo de Información.....	57
3.2 Diagrama de Flujo de Información de Menús.....	58
3.3 Mapas de Navegabilidad de la Aplicación Móvil.....	59
3.4 Mapa General de Navegabilidad de la Aplicación Móvil.....	62
<b>Capítulo IV: Pruebas.....</b>	<b>63</b>
<b>1 PLAN DE PRUEBAS.....</b>	<b>64</b>
1.1 Selección de Dispositivos.....	64
1.2 Informe de Pruebas, Adaptación y Compatibilidad.....	65
1.2.1 Pruebas.....	65
1.2.2 Adaptación Smartphone.....	68
1.2.3 Adaptación Tablet.....	68
1.2.4 Adaptación Multidispositivo.....	69
1.3 Informe de Pruebas de la Usabilidad de la Aplicación Móvil...70	70
1.3.1 Informe de Evaluación Heurística de la Usabilidad.....	70
1.3.2 Aspectos Generales.....	71
1.3.3 Estructura y Navegación.....	72
1.3.4 Rótulos de Vistas.....	73
1.3.5 Layout.....	74
1.3.6 Lenguaje.....	75
1.3.7 Control.....	76
1.3.8 Elementos Multimedia.....	77
<b>Capítulo V: Implantación del Sistema.....</b>	<b>78</b>
<b>1 IMPLANTACIÓN.....</b>	<b>79</b>
1.1 Costes Totales.....	79
1.1.1 Coste final del esfuerzo del proyecto.....	79
1.1.2 Resumen de costes.....	80
1.1.2.1 Resumen de costes del desarrollo del proyecto.....	80

1.1.2.2 Costes de personal.....	80
1.1.2.3 Costes de amortización del hardware.....	81
1.1.2.4 Costes de software.....	81
1.1.2.5 Costes indirectos.....	81
<b>1.2 Seguridad del Sistema.....</b>	<b>82</b>
1.2.1 Medidas Generales.....	82
1.2.1.1 Ataques de fuerza bruta.....	82
1.2.1.2 Registro IP.....	82
1.2.1.3 Modificación de permisos.....	83
1.2.1.4 Ataques lógicos.....	83
1.2.1.5 Inyección.....	84
1.2.1.6 Usuarios y permisos.....	84
1.2.1.7 Tecnologías de terceros.....	84
<b>Capítulo VII: Conclusiones.....</b>	<b>86</b>
<b>1 CONCLUSIONES.....</b>	<b>87</b>
1.1 Metodología de Trabajo.....	87
1.2 Resultados.....	87
1.3 Líneas de Trabajo Futuras.....	87
1.4 Conclusiones Personales.....	88
<b>GLOSARIO.....</b>	<b>89</b>
<b>BIBLIOGRAFÍA.....</b>	<b>90</b>
<b>ANEXOS.....</b>	<b>91</b>
<b>1 GENERACIÓN DEL EMPAQUETADO.....</b>	<b>91</b>
<b>2 MANUAL DE INSTALACIÓN Y CONFIGURACIÓN.....</b>	<b>96</b>
2.1 Instalación del Servidor y Cliente.....	96
2.1.1 Instalación del Servidor de Bases de Datos.....	96
2.1.1.1 Instalación de Apache.....	96
2.1.1.2 Instalación de MySQL.....	96
2.1.2 Procedimiento de Instalación de los Servicios de Comunicación en Red.....	97
2.1.2.1 Instalación de Postfix.....	97
2.1.2.2 Configuración de Postfix.....	97
2.1.2.3 Instalación de Dovecot.....	98
2.1.3 Procedimiento de Instalación en el Cliente.....	99
<b>3 MANUAL DE USUARIO.....</b>	<b>100</b>
3.1 Inicio y Acceso.....	100
3.2 Pestaña Inicio.....	101
3.2.1 Pantalla de Inicio.....	101
3.2.2 Entrada y Salida.....	101
3.2.2.1 Entrar.....	101
3.2.2.2 Salir.....	103
3.3 Pestaña Informes.....	104
3.3.1 Pantalla de Informes.....	104
3.4 Pestaña Alarma.....	105
3.4.1 Pantalla de Alarma.....	105
3.5 Pestaña Ajustes.....	107
3.5.1 Pantalla de Ajustes.....	107
3.5.1.1 Desactivación y modificación de password de alarma.....	108
3.5.1.2 Nuevo empleado.....	108

3.5.1.3 Borrar empleado.....	109
<b>4 CÓDIGOS.....</b>	<b>110</b>
<b>4.1 Base de Datos.....</b>	<b>110</b>
4.1.1 Modelado Físico Relacional: Script SQL.....	110
4.1.2 Configuración de Apache.....	111
4.1.3 Configuración de MySQL.....	112



## Lista de figuras

Figura 1: Estadística de descargas en un mes de TimeClock ST.....	16
Figura 2: Estadística de descargas de Timeclock ST desde su aparición....	17
Figura 3: Ranking de los usuarios.....	17
Figura 4: Gestión de la Planificación .....	19
Figura 5: Gestión de la Asignación de Recursos.....	19
Figura 6: Planificación mensual del proyecto en Gantt Project.....	20
Figura 7: Diagrama de Gantt del Proyecto.....	24
Figura 8: Factores de Ajuste del Esfuerzo de Boehm (1981).....	25
Figura 9: Ecuación de Estimación del Tiempo de Desarrollo.....	27
Figura 10: Ecuación de Número Medio Recomendado de Técnicos.....	27
Figura 11: Tipos de PCB basados en microcontrolador Raspberry Pi.....	30
Figura 12: GPIO (General Purpose Input/Output) para Raspberry Pi B.....	32
Figura 13: Parte delantera de la placa Raspberry Pi B.....	33
Figura 14: Módulo Raspicam para Raspberry Pi.....	33
Figura 15: Proceso de Negocio de Generación de Informes.....	35
Figura 16: Proceso de Elicitación de Requisitos.....	36
Figura 17: Diagrama de Casos de Uso del Sistema.....	39
Figura 18: Esquema Básico de Comunicaciones en Red.....	50
Figura 19: Directorio de Default Virtual Host desde Apache.....	52
Figura 20: Listado completo actualizado de frameworks multiplataforma.....	54
Figura 21: Servicios de Phonegap para Aplicaciones Móviles Híbridas.....	55
Figura 22: GUI de WorkersVision en IDE Android Studio (I).....	56
Figura 23: GUI en IDE Android Studio (II).....	56
Figura 24: Diagrama de Flujo de Información de Menús.....	58
Figura 25: Mapa de Navegabilidad de la Aplicación (I).....	59
Figura 26: Mapa de Navegabilidad de la Aplicación (II).....	60
Figura 27: Mapa de Navegabilidad de la Aplicación (III).....	61
Figura 28: Mapa Completo de Navegabilidad del Prototipado de la App.....	62
Figura 29: Smartphone Vowney v5.....	64
Figura 30: Tablet BQ Elcano.....	64
Figura 31: Informe de Pruebas: Informe usuario.....	65
Figura 32: Informe de Pruebas: Alarma.....	65
Figura 33: Informe de Pruebas: Nuevo Empleado (I).....	66
Figura 34: Informe de Pruebas: Nuevo Empleado (II).....	66
Figura 35: Informe de Pruebas: Password desactivación.....	66
Figura 36: Informe de Pruebas: Fallo de conexión.....	67
Figura 37: Informe de Pruebas: Adaptación Multidispositivo Móvil.....	69
Figura 38: Criterios de Usabilidad: Evaluación Final.....	70
Figura 39: Criterios de Usabilidad: Identidad e Información.....	71
Figura 40: Criterios de Usabilidad: Estructura y Navegación.....	72
Figura 41: Criterios de Usabilidad: Rótulos de Vistas.....	73
Figura 42: Criterios de Usabilidad: Layout.....	74
Figura 43: Criterios de Usabilidad: Lenguaje.....	75
Figura 44: Criterios de Usabilidad: Control.....	76
Figura 45: Criterios de Usabilidad: Elementos Multimedia.....	77
Figura 46: Ecuación de Estimación del Tiempo de Desarrollo.....	79

Figura 47: Ecuación de Amortización por el Método Directo.....	81
Figura 48: Configuración de Postfix.....	97
Figura 49: Reconfiguración del paquete Postfix.....	98
Figura 50: Instalación en Cliente: Permisos.....	99
Figura 51: Instalación en Cliente: Proceso.....	99
Figura 52: Acceso e Inicio de la App (I).....	100
Figura 53: Acceso e Inicio de la App (II).....	100
Figura 54: Acceso e Inicio de la App (III).....	100
Figura 55: Pantalla de Inicio.....	101
Figura 56: Entrada de Empleados (I).....	101
Figura 57: Entrada de Empleados (II).....	102
Figura 58: Salida de Empleados.....	103
Figura 59: Informe de Empleado.....	104
Figura 60: Inicio de Alarma.....	105
Figura 61: Activación de Alarma (I).....	105
Figura 62: Activación de Alarma (II).....	106
Figura 63: Recogida y Aviso por Detección de Intruso.....	106
Figura 64: Inicio de Ajustes.....	107
Figura 65: Ajustes de Configuración.....	107
Figura 66: Desactivación y Modificación de Clave.....	108
Figura 67: Nuevo Empleado.....	108
Figura 68: Borrar Empleado.....	109

## Lista de Tablas

Tabla 1: Comparativa de ventas de la app TimeClock ST.....	17
Tabla 2: Etapas Principales del Proyecto.....	18
Tabla 3: Elementos de Representación del Diagrama de Gantt.....	23
Tabla 4: Comparativa oficial de la familia Raspberry Pi .....	31
Tabla 5: UC-0001 - Caso de Uso sobre Seleccionar Empleado para Entrar	40
Tabla 6: UC-0002 - Caso de Uso sobre Seleccionar Empleado para Salir...	41
Tabla 7: UC-0003 - Caso de Uso sobre Registrar Empleado.....	42
Tabla 8: UC-0004 - Caso de Uso sobre Eliminar Empleado.....	43
Tabla 9: UC-0005 - Caso de Uso sobre Activar Alarma.....	44
Tabla 10: Elementos de un Diagrama de Flujo de Información de Menús....	57
Tabla 11: Resumen de Costes del Proyecto.....	80
Tabla 12: Resumen de Costes de Personal.....	80
Tabla 13: Resumen de Costes de Hardware.....	81
Tabla 14: Resumen de Costes Indirectos.....	82

# Capítulo I: Introducción

# 1 INTRODUCCIÓN

## 1.1 MOTIVACIÓN

El rápido avance de la tecnología e industrialización en los países desarrollados hace que las empresas deban reaccionar más rápidamente a situaciones cambiantes. Habitualmente suelen ser causadas por un consumidor más exigente y un mercado mucho más competitivo para ofrecer la seguridad física en la empresa simplificando los procesos de monitorización incluso para el propio usuario, a través de su acceso remoto y/o a través de dispositivos móviles.

A pesar de que se puede encontrar en el mercado de consumo (ofertados por empresas de domótica y seguridad): soluciones hardware y software estandarizadas para cualquier tipo de empresa que quiera monitorizar la presencia. Estas soluciones realmente no llegan a utilizarse del todo y en su completa funcionalidad, a menos de que, se haya invertido en el desarrollo de un producto totalmente adaptado.

Este proyecto trata de desarrollar una propuesta de sistema informático específico que ayude en los mecanismos de control de presencia de entradas y salidas de los empleados pertenecientes a una PYME. El modelo final a su misma vez puede servir para desarrollar e integrar otros futuros módulos de gestión para la videovigilancia con webcam ó transmisión multimedia a través de CCTV.

## 1.2 OBJETIVOS DEL TRABAJO

Los objetivos generales del proyecto se estructuran en los siguientes puntos:

- Analizar y desarrollar un sistema informático que permita facilitar la gestión del control de presencia de entrada y salida en centros de trabajo, delegaciones, áreas o departamentos controlando la periodicidad de las alarmas en periodos de tiempo de inactividad.
- Documentar y producir los entregables necesarios de cada una de las fases principales del desarrollo del sistema, desde la perspectiva de la Ingeniería del Software y el lenguaje unificado de modelado (UML).
- Profundizar en el conocimiento de la herramienta de IDE Android Studio y el framework multiplataforma Phonegap que dará soporte al desarrollo móvil cliente del proyecto.
- Ofrecer interoperabilidad y configurar el nodo a partir de los servicios de red definidos en un equipo empotrado basado en el microcontrolador de hardware libre Raspberry Pi.

## 1.3 ENFOQUE, MÉTODO SEGUIDO Y BREVE DESCRIPCIÓN DE LOS OTROS CAPÍTULOS

La documentación del proyecto está estructurada de manera que comience por el capítulo 1 de **Introducción** que comprende los objetivos generales del proyecto y el caso de estudio aplicado. Así mismo, se incluye la planificación y estimación tanto en tiempo, coste y recursos del desarrollo además del estudio de alternativas de solución existentes en el mercado.

En el capítulo 2 se le dedicará especial atención a la elicitación, que consiste en la captura de requisitos, es donde se deben tomar las decisiones más relevantes sobre las cuales se basará el resto del proyecto, además del análisis y modelado básico de los procesos de negocio.

En el capítulo 3 se estudiará el diseño de la arquitectura del sistema, la persistencia de datos y el diseño navegacional de la aplicación. Además se analiza el comportamiento del sistema definiendo los prototipos de pantalla de acuerdo a los requisitos existentes y el mapa de navegabilidad.

En el capítulo 4 se analizan los componentes usados, el entorno de desarrollo escogido y algunos casos de ejemplo para el control de usuarios, así como, la generación de informes.

El capítulo 5 determina los resultados de algunas pruebas realizadas al sistema, para verificar los requisitos estudiados en los capítulos anteriores. El siguiente capítulo 6 ofrece el estudio de coste final estimado de implantación del sistema.

El capítulo 7 representa un epílogo del proyecto, ya que comenta el cumplimiento de los objetivos esperados del documento. Se añaden algunas propuestas de mejora o ampliaciones que se podrían aplicar al sistema.

La parte última es una **bibliografía** donde se incluyen no sólo los libros y artículos de investigación consultados, sino también algunas direcciones de páginas web que han sido de bastante ayuda para profundizar en varios aspectos y capítulos de realización del proyecto. Asimismo se incorpora como **anexos**: los manuales de usuario (funcional y de instalación y configuración del sistema).

La documentación escrita del proyecto va acompañada en un CD-ROM, en el que se incluye además la aplicación móvil desarrollada junto con el código fuente completo de toda la plataforma desarrollado y comentado. Además se han incluido: los ficheros binarios ejecutables, empaquetados y librerías requeridas para su ejecución, los ficheros base para el modelado estructural u orientado a objetos UML, imágenes del sketch, del prototipado y de la interfaz de la aplicación (en cada uno de los lenguajes empleados). Por último, el software requerido y usado en el desarrollo, configuración, instalación técnica, de implantación y despliegue (SGBD, librerías, scripts SQL, kits de desarrollo, etc.).

**ESTA PARTE QUEDA  
PROTEGIDA Y NO SE  
PUBLICA POR INTERESES  
PROFESIONALES DEL  
AUTOR**



## 2.6 DESARROLLO POR MICROCONTROLADOR

### 2.6.1 Control centralizado y monitorizado por microcontrolador

El microcontrolador elegido para centralizar las conexiones principales en red es Raspberry Pi<sup>5</sup>. Dicho micro se basa en una plataforma denominada open hardware que reúne en una pequeña placa de circuito impreso (PCB) los componentes necesarios para conectar p.e. con la red local de la empresa. Al ser Open Hardware, tanto su diseño como su distribución son libres. Es decir, puede utilizarse sin inconvenientes para desarrollar cualquier tipo de proyecto sin tener que adquirir ninguna licencia adicional. Raspberry Pi es una familia que ha crecido mucho con el tiempo y ahora tenemos multitud de dispositivos entre los que elegir. Desde las placas modelo A (la ideal para prototipos y la enseñanza, tanto por precio como por capacidad), pasando por el A+ (de reducidas dimensiones, para que incluso se cosan en la ropa: intermitentes para los ciclistas), el modelo B (si necesitamos más memoria o entradas/salidas) hasta la más reciente B v2: pensada para drones, proyectos domóticos, emuladores de videojuegos ó equipos multimedia de alto rendimiento. Todas ellas con microprocesador ARM y la mayoría con conector Ethernet 10/100 Mbps. A continuación se exponen los **criterios generales** para seleccionar este microcontrolador:

- Bajo precio.
- Bajo consumo. El microcontrolador será alimentado con baterías y tiene que tener una autonomía suficiente. Proporcionan una o varias líneas de alimentación de 5 voltios para controlar conexiones con leds, motores, servos ó sensores.
- Potencia suficiente para la disponibilidad de servicios en red y de B.D.
- Escalabilidad, dotando de un número de conexiones para periféricos y módulos hardware adicionales (WIFI, sensorística, etc.) con el objeto de ampliar futuras funcionalidades.
- Factor de forma, por su reducidas dimensiones.
- Facilidad en la programación para integrarlo al desarrollo del sistema.



Figura 11: Tipos de PCB basados en microcontrolador Raspberry Pi

5 Proyecto Raspberry Pi: <https://www.raspberrypi.org>



	RPI Model A	RPI Model A+	RPI Model B	RPI Model B+	RPI 2 Model B
SoC	BROADCOM BCM2835	BROADCOM BCM2835	BROADCOM BCM2835	BROADCOM BCM2835	BROADCOM BCM2836
CPU	ARM11 ARMV6 700 MHZ.	ARM11 ARMV6 700 MHZ.	ARM11 ARMV6 700 MHZ.	ARM11 ARMV6 700 MHZ.	ARM11 ARMV7 ARM CORTEX-A7 4 NÚCLEOS 900 MHZ.
GPU	BROADCOM VIDEOCORE IV 250 MHZ. OPENGL ES 2.0	BROADCOM VIDEOCORE IV 250 MHZ. OPENGL ES 2.0	BROADCOM VIDEOCORE IV 250 MHZ. OPENGL ES 2.0	BROADCOM VIDEOCORE IV 250 MHZ. OPENGL ES 2.0	BROADCOM VIDEOCORE IV 250 MHZ. OPENGL ES 2.0
Memoria RAM	256 MB LPDDR SDRAM 400 MHZ.	256 MB LPDDR SDRAM 400 MHZ.	512 MB LPDDR SDRAM 400 MHZ.	512 MB LPDDR SDRAM 400 MHZ.	1 GB LPDDR2 SDRAM 450 MHZ.
Puertos USB	1	1	2	4	4
GPIO	26 PINES	40 PINES	26 PINES	40 PINES	40 PINES
Vídeo	HDMI 1.4 1920X1200	HDMI 1.4 1920X1200	HDMI 1.4 1920X1200	HDMI 1.4 1920X1200	HDMI 1.4 1920X1200
Almacenamiento					
Ethernet 10/100MBPS	NO	NO	SI	SI	SI
Tamaño	85,60X56,5 MM	65X56,5 MM.	85,60X56,5 MM	85,60X56,5 MM	85,60X56,5 MM
Peso en g.	45	23	45	45	45
Precio	29,95€	29,95€	34,95€	34,95€	39,95€

Tabla 4: Comparativa oficial de la familia Raspberry Pi

### 2.6.2 Características generales de las Raspberry Pi

La Fundación Raspberry Pi promueve la enseñanza de las ciencias de computación y tecnología y da soporte del sistema operativo en sus distintas SBC con las siguientes distribuciones GNU/Linux para arquitectura ARM: Raspbian (derivada de Debian), Arch Linux ARM (derivado de Arch Linux) y Pidora (derivado de Fedora). Se pueden encontrar dos modelos principales (con ligeras variaciones), Modelo A y B con un precio máximo alrededor de los 50 €. Los demás elementos como la fuente de alimentación o la tarjeta SD deben ser comprados por separado. Las características principales son las siguientes:

- Modelo A tiene 256MB de RAM, 1 puerto USB y no tiene Ethernet.
- Modelo B tiene 512MB de RAM, 2 puertos USB y Ethernet.
- Modelo B v2 tiene CPU ARM de 900 Mhz, 1 GB de RAM, 4 puertos USB y Ethernet y conexión GPIO a 40 pines.
- La mayor dimensión de la SBC de la familia son 85.60mm x 56mm x 21mm y su peso de 45 gramos.

La CPU de las Raspberry Pi suelen ser Broadcom BCM2835/36 de tipo ARM, muy similares a los equipos empotrados Arduino. Este microprocesador ARM es idóneo para tareas que requieran de un gran rendimiento tanto en aplicaciones multimedia como en movilidad debido a su bajo consumo. Está diseñado y optimizado para tener un bajo coste y una eficiencia energética. Las principales características del System on Chip (SoC) son las siguientes:

- Basado en CPU ARM1176JZ-F de bajo consumo y Quad-Core A7.
- Basado en la GPU multimedia VideoCore IV de doble núcleo.
- Reproducción a 1080p, compresión H.264

### 2.6.3 Características específicas del modelo seleccionado

Se ha escogido el microcontrolador **Raspberry Pi B** en base a los **criterios generales** anteriormente comentados. Es un SBC formada por un microcontrolador ARM1176JZF-S. Considerando que este micro es fácil de programar y de instalar (no requiere de grandes conocimientos técnicos) y tiene la posibilidad de ser alimentado a través de la conexión USB o con una fuente de alimentación externa. Además puede proporcionar la tensión necesaria para alimentar a los suficientes distintos dispositivos de nuestro sistema que se necesite para ser ajustado a los objetivos fijados del sistema y cubrir futuras funcionalidades. Las especificaciones técnicas destacadas de este SBC son las siguientes:

#### System on Chip (SoC):

- Modelo Broadcom BCM2835
- CPU ARM1176JZF-S a 700 MHz
- GPU Broadcom VideoCore IV

#### Memoria:

- RAM 512 MB (compartidos con la GPU)

#### Conexiones:

- Modelo Broadcom BCM2835
- 2 x USB 2.0
- 1 x Salida audio mini jack 3.5 mm
- 1 x Salida audio/vídeo HDMI
- 1 x Salida vídeo compuesto RCA
- 1 x Micro USB
- 1 x RJ45 10/100 Ethernet RJ45
- Slot SD para tarjetas
- 26 GPIO I/O programables con entradas a 3.3V / 5V

#### Alimentación:

- 5V/700 mA (3.5 W) via microUSB

#### Dimensiones:

- 85.6 mm x 53.98 mm

#### Raspberry Pi Model B

##### Rev 1 P1 GPIO Header

	Pin No.		
3.3V	1	2	5V
GPIO0	3	4	5V
GPIO1	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO21	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7

Figura 12: GPIO (General Purpose Input/Output) para Raspberry Pi B



Figura 13: Parte delantera de la placa Raspberry Pi B

## 2.6.4 Módulos hardware integrados

El microcontrolador Raspberry proporciona una plataforma open hardware que reúne los requisitos básicos de conectividad a la red con los dispositivos móviles. Todo esto se hará con el módulo conector ya integrado a la PBC de Ethernet 10/100 BaseT. A diferencia de otros equipos empotrados como Arduino Uno en el que se requiere el encaje del módulo a partir del conexionado de pines. Con la ayuda de la librería de la API proporcionada, podremos configurar los servicios de red que proporcionaremos al equipo. La configuración de red se realiza mediante el software base de Raspbian, por lo que podremos adaptar e integrar con facilidad la placa a nuestra red local. El software base, el conjunto de paquetes del sistema basados en servicios de red, los datos periódicos recogidos y las notificaciones generadas en el sistema necesitan ser almacenados. El microcontrolador Raspberry Pi no dispone de memoria suficiente para el almacenamiento de ficheros, teniendo entonces la necesidad de incluir una tarjeta SD (Secure Digital) compatible con tarjetas sólo SD, en su versión B. Como podemos observar en la imagen anterior, el dispositivo de almacenamiento se encuentra integrado al propio PBC.

En la web <http://comohacer.eu/comparativa-y-analisis-raspberry-pi-vs-competencia> se ofrece información sobre el conjunto de módulos hardware destacados que podrían resolver futuras funcionalidades del sistema como las Raspi Cam (véase figura lateral):



Figura 14: Módulo Raspicam para Raspberry Pi

**ESTA PARTE QUEDA  
PROTEGIDA Y NO SE  
PUBLICA POR INTERESES  
PROFESIONALES DEL  
AUTOR**



# Capítulo II: Análisis de Requisitos

**ESTA PARTE QUEDA  
PROTEGIDA Y NO SE  
PUBLICA POR INTERESES  
PROFESIONALES DEL  
AUTOR**



## 2 USUARIOS Y CONTEXTO DE USO

### 2.1 DESCRIPCIÓN DEL CAPÍTULO

La metodología de elicitación de requisitos de [Durán y Bernárdez, 2002] propone un enfoque sistemático de la Ingeniería de Requisitos que define un proceso que posibilita la descomposición sistemática y minuciosa de los requisitos software. Esta descomposición realiza una especificación detallada que constituirá el modelo conceptual del sistema deseado. La etapa de ingeniería de requisitos es la fase compuesta por la elicitación de requisitos y la de análisis de requisitos (equivalente a la de análisis del sistema y que se describirá en el siguiente capítulo). El objetivo principal es mejorar la calidad del proceso de producción de software, ya que el modelo conceptual se construye como una representación precisa, estructurada y bien definida de los requisitos de los usuarios.

El análisis de requerimientos que nos plantea la construcción sistema de monitorización de presencia en una PYME se vino descubriendo a medida que se obtenían las necesidades reales del proyecto, de las referencias Internet y del material bibliográfico consultado. Todo esto ha permitido generar una lista de requisitos para la construcción del modelo inicial del sistema. En los siguientes apartados se verá mas detalladamente toda esta información, en función del caso de estudio planteado.

En primer lugar definiremos los participantes y actores del sistema, luego los objetivos específicos del sistema, es decir, con que finalidad se desarrolla este sistema. A continuación se realizará una enumeración y definición de los requisitos funcionales a partir del diagrama de casos de uso en el contexto del sistema. La mayor parte de los flujos existentes en los escenarios principales de los casos de uso serán descritos en el manual de usuario.

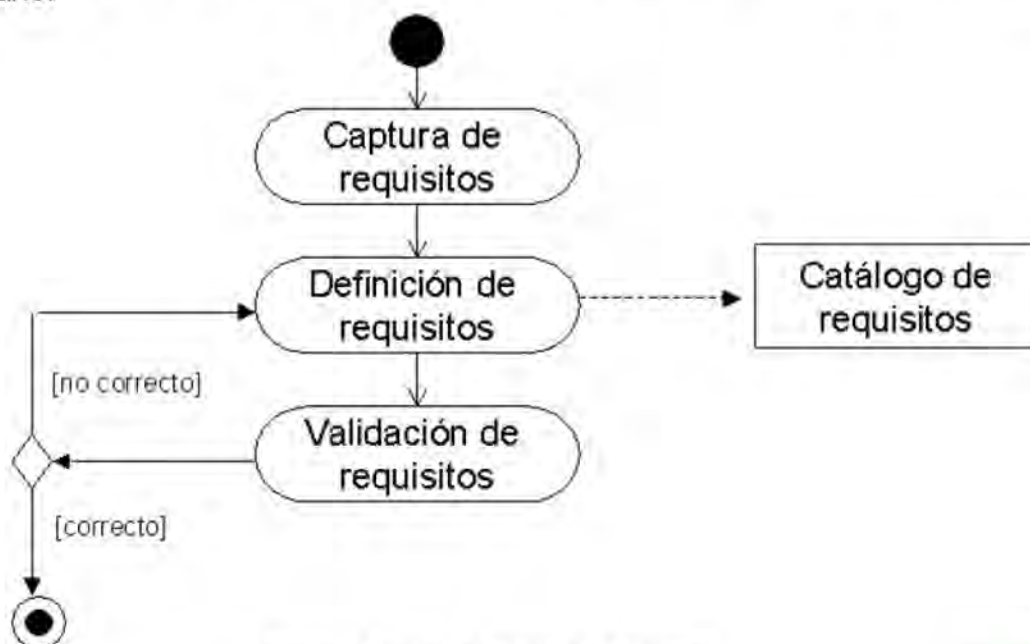


Figura 16: Proceso de Elicitación de Requisitos

## 2.2 PARTICIPANTES EN EL PROYECTO

<b>Participante</b>	Alejandro Cardo Grau
<b>Organización</b>	Univesidad Oberta de Catalunya
<b>Rol</b>	Ingeniero Informático
<b>Es desarrollador</b>	Sí
<b>Es cliente</b>	No
<b>Es usuario</b>	No
<b>Comentarios</b>	Ninguno

<b>Participante</b>	Tutor
<b>Organización</b>	Univesidad Oberta de Catalunya
<b>Rol</b>	Director del proyecto
<b>Es desarrollador</b>	No
<b>Es cliente</b>	No
<b>Es usuario</b>	No
<b>Comentarios</b>	Tutor encargado de la dirección, revisión, control, planificación y sugerencias de mejora del proyecto.

<b>Organización</b>	Universidad Oberta de Catalunya
<b>Dirección</b>	Universidad Oberta de Catalunya. Rambla del Poblenou, 156, 08018 Barcelona
<b>Teléfono</b>	(+34) 934 817 272
<b>Comentarios</b>	Es la Universidad Oberta de Catalunya (UOC) que interviene en la dirección del trabajo fin de máster.

## 2.3 ACTORES DEL SISTEMA

En el sistema, se han diferenciado los roles de *administrador* y *empleado* para algunos casos de uso. Era necesario distinguirlos para utilizarse en un número determinado de casos de uso del actor *administrador* (p.e. generación de informes de entrada-salida). A partir del establecimiento de una jerarquía de actores en cada diagrama, se designa como actor básico al *empleado* y actor derivado al *administrador*. Éste último consigue un aspecto funcional propio para algunos casos de uso, aunque complementa los casos de uso del actor básico asumiendo el papel de éste.

<b>Participante</b>	Administrador del Sistema
<b>Organización</b>	Empresa del sector
<b>Rol</b>	Empleado Administrador del Sistema
<b>Es desarrollador</b>	No
<b>Es cliente</b>	Sí
<b>Es usuario</b>	Sí
<b>Comentarios</b>	Ninguno

<b>Participante</b>	Empleado
<b>Organización</b>	Empresa del sector
<b>Rol</b>	Empleado
<b>Es desarrollador</b>	No
<b>Es cliente</b>	Sí
<b>Es usuario</b>	Sí



## 2.4 OBJETIVOS DEL SISTEMA

OBJ-001	Control de presencia
Versión	1.0 ( 10/10/2016 )
Autores	Alejandro Cardo Grau
Descripción	El sistema deberá <i>controlar la hora de entrada y salida de empleados en el centro de trabajo para poder computar el horario de trabajo realizado por cada empleado.</i>
Subobjetivos	<ul style="list-style-type: none"> <li>■ <b>[OBJ-002] Validación:</b> El sistema deberá <i>desarrollar un registro de validación y listado de los empleados dados de alta y baja para controlar sus entradas y salidas.</i></li> <li>■ <b>[OBJ-003] Control Interior:</b> El sistema deberá <i>facilitar el reconocimiento e identificación del estado de los empleados que se encuentran en la zona.</i></li> </ul>
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	El control de presencia implícitamente comprende una verificación del estado de los empleado. Al entrar o salir un empleado debe quedar reflejado en el sistema la hora del evento.

OBJ-004	Gestión de Notificaciones
Versión	1.0 ( 10/09/2015 )
Autores	Alejandro Cardo Grau
Descripción	El sistema deberá <i>detectar la presencia de intrusos en el centro de trabajo en horario de inactividad a partir de la des/conexión de las alarmas existentes.</i>
Subobjetivos	
Importancia	vital
Urgencia	hay presión
Estado	validado
Estabilidad	alta
Comentarios	Las alarmas serán temporalizadas de forma predeterminada y en su activación para un número de segundos límite en contador descendente.

OBJ-005	Gestión de Informes
Versión	1.0 ( 10/10/2016 )
Autores	Alejandro Cardo Grau
Descripción	El sistema deberá <i>gestionar informes personalizados de consulta a partir de una fecha o por empleado específico.</i>
Subobjetivos	<ul style="list-style-type: none"> <li>■ <b>[OBJ-0004] Generación de Informes Personalizados:</b> El sistema deberá <i>ofrecer las herramientas necesarias para crear nuevas informes, mantener un histórico por temporalidad y controlar el estado de los mismos.</i></li> </ul>
Importancia	vital
Urgencia	hay presión
Estado	validado
Estabilidad	alta
Comentarios	Los informes de consulta serán por empleado específico o intervalos de fecha/hora.

**ESTA PARTE QUEDA  
PROTEGIDA Y NO SE  
PUBLICA POR INTERESES  
PROFESIONALES DEL  
AUTOR**



# Capítulo III: Diseño del Sistema

**ESTA PARTE QUEDA  
PROTEGIDA Y NO SE  
PUBLICA POR INTERESES  
PROFESIONALES DEL  
AUTOR**



## 2.2 ARQUITECTURA DEL SERVIDOR

### 2.2.1 Tecnología Ajax

Puesto que el principal formato de comunicación entre el servidor y el cliente mediante consultas AJAX (Asynchronous JavaScript And XML) es XML, la elección de esta tecnología nos permitirá en las consultas POST solicitadas por la aplicación móvil ahorrarnos un coste destacado en el procesamiento final. AJAX es una técnica de desarrollo web para crear aplicaciones interactivas que se ejecutan en el cliente manteniendo una comunicación asíncrona con el servidor en segundo plano.

### 2.2.2 Servidor web Apache

HTTP (Hypertext Transfer Protocol) es el protocolo usado en cada transacción de nuestro sistema. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Nos permitirá comunicar la aplicación móvil con el microcontrolador Raspberry Pi. Por lo tanto, para el funcionamiento del sistema será necesario que el equipo empotrado atienda las peticiones del cliente móvil, ejecutando el lenguaje de contenido dinámico (PHP) y enviando el documento HTML generado a través del protocolo HTTP.

El servidor HTTP Apache es un servidor web HTTP de código abierto, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. Se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Entre sus principales ventajas se encuentran su modularidad, su disponibilidad en múltiples plataformas y el excelente soporte debido al gran número de desarrolladores y usuarios. En los repositorios de Raspbian se encuentra disponible este paquete para configurar y ofrecer los servicios de petición necesarios a la aplicación móvil.

### 2.2.3 Tecnología PHP

La mayoría de las distribuciones ya cuentan con paquetes precompilados para la instalación del servidor web Apache y el módulo PHP. Se recomienda esta opción con el fin de facilitar la administración y la actualización del sistema.

Una vez instalados, será necesario configurar Apache para que cargue el módulo PHP y procese ficheros con dicha extensión. Para ello habrá que añadir las siguientes líneas en el fichero de configuración `httpd.conf`, que se encuentra en el subdirectorio de `/etc`:

```
LoadModule php5_module modules/libphp5.so
<FilesMatch \.php$>
    SetHandler application/x-httpd-php
</FilesMatch>
```

Para arrancar el servidor, simplemente habrá que iniciar el demonio `httpd` ejecutando el siguiente comando como superusuario:

```
#> /etc/init.d/httpd start
```

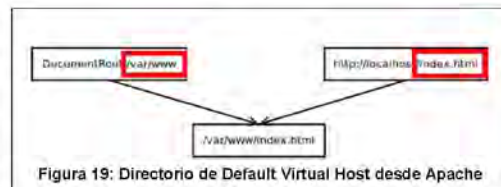
## 2.3 COMPONENTES SOFTWARE DEL SISTEMA

### 2.3.1 Configuración del Apache

La siguiente estructura de ficheros, presente en el directorio `/etc/apache2` del servidor, se deberá tener en cuenta para facilitar la administración del servidor web Apache en la presente actividad:

- **apache2.conf.** Por medio de directivas se incluyen todos los módulos necesarios para que el servidor Apache funcione: la inclusión de los ficheros de la estructura, el control de respuestas de los clientes y el servidor, etc.

- **sites-enabled:** Directorio que contiene los archivos de configuración de cada *virtual host*. Cada *virtual host* representará un sitio web que van a ser servida por el mismo servidor. Apache permite mantener al mismo tiempo varios *virtual host*, pero sólo uno de ellos activo sobre la misma dirección. Por defecto, la configuración del servidor define un *virtual host* cuyo directorio de trabajo es `/var/www`. (ver: `sites-available/default` – directiva del alias: `DocumentRoot`)



- **mods-enabled/\*.load** y **mods-available/\*.conf:** Directorio que contiene la carga y configuración de los módulos. Un ejemplo es el módulo que controla la monitorización del consumo de recursos del servidor web de Apache y ofrece información a través del sitio web por defecto: `mod_status`.

```

/etc/apache2/apache2.conf
(activación de mod_status)
...
ExtendedStatus On
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from 12.34.56.78
</Location>
...
  
```

- **ports.conf.** Se definen que puertos escucha el servidor Apache, por defecto, puerto HTTP/80.

### 2.3.2 Librerías del servidor

Para enviar los correos electrónicos, PHP usa la función del sistema `sendmail`. La mayoría de las distribuciones GNU/Linux, junto con Raspbian, ya vienen con la librería preinstalada o cuentan con paquetes precompilados para su instalación. Para iniciar el servicio y que puedan enviarse las notificaciones por correo electrónico, tan sólo habrá que ejecutar el siguiente comando como usuario `root`:

```
#> /etc/init.d/sendmail start
```

**ESTA PARTE QUEDA  
PROTEGIDA Y NO SE  
PUBLICA POR INTERESES  
PROFESIONALES DEL  
AUTOR**



# Capítulo IV: Pruebas



**ESTA PARTE QUEDA  
PROTEGIDA Y NO SE  
PUBLICA POR INTERESES  
PROFESIONALES DEL  
AUTOR**



# Capítulo V: Implantación del Sistema

**ESTA PARTE QUEDA  
PROTEGIDA Y NO SE  
PUBLICA POR INTERESES  
PROFESIONALES DEL  
AUTOR**



# Capítulo VII: Conclusiones

**ESTA PARTE QUEDA  
PROTEGIDA Y NO SE  
PUBLICA POR INTERESES  
PROFESIONALES DEL  
AUTOR**



# GLOSARIO

<b>Caso de uso</b>	Operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un/varios actores o bien desde la invocación desde otro caso de uso. Es una técnica para captura de requisitos de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje.
<b>Diagrama de casos de uso</b>	Representación de la forma en como un cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (conjunto de operaciones o casos de uso).
<b>Diagrama de actividades</b>	Representación de los flujos de control de trabajo, paso a paso, de un proceso de negocio y operacionales de los componentes en un sistema.
<b>Equipo empotrado</b>	Sistema embebido de propósito especial que usa un microcontrolador para realizar funciones específicas de forma autónoma. Permite la posibilidad de modularizar el hardware y disponer de grandes niveles de integración y reducción de sus componentes y conectores.
<b>Framework</b>	Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos de software (bibliotecas, API, plugins, herramientas, etc.), que sirven de base para el desarrollo del software.
<b>IDE</b>	Aplicación informática compuesta por un conjunto de herramientas de programación que facilitan la tarea al programador y obtienen mayor rapidez en el desarrollo. Permiten obtener artefactos como: códigos fuentes, ejecutables, enlazadores de los códigos objetos, manuales técnicos del código fuente, pruebas, etc.
<b>Layout</b>	Esquema de distribución y representación gráfica del conjunto de los elementos y componentes visuales del diseño de una interfaz gráfica.
<b>Mapa de navegabilidad</b>	Representación gráfica de la organización y flujos de la información de la estructura de las interfaces gráficas de usuario. Expresa todas las relaciones de jerarquía y secuencia y permite elaborar escenarios de comportamiento de los usuarios.
<b>Microcontrolador</b>	Circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica.
<b>Mockup</b>	Prototipo con modelo a escala o tamaño real de un diseño, utilizado para la demostración, evaluación, promoción, y para otros fines que proporcionen partes de las funcionalidades de un sistema.
<b>Proceso de negocio</b>	Conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido. Cada proceso de negocio tiene sus entradas, funciones y salidas.
<b>PYME</b>	Empresa pequeña o mediana en cuanto a volumen de ingresos, valor del patrimonio y número de trabajadores.
<b>Responsive design</b>	Conjunto de métodos que permiten adaptar y programar la apariencia de la interface gráfica al dispositivo que se esté utilizando para visualizarla.
<b>UML</b>	Lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. Es una forma de modelar objetos conceptuales como lo son procesos de negocio y funciones de sistema.

# BIBLIOGRAFÍA

## LIBROS Y ARTÍCULOS

**[Durán y Bernárdez, 2002]** Durán, A. y Bernárdez B., *Metodología para la Elicitación de Requisitos de Sistemas Software*, versión 2.3, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Abril, 2002.

**[Dolado y Fernández, 2000]** Dolado Cosín, J. J. y Fernández Sanz L., *Medición para la Gestión de la Ingeniería del Software*, RA-MA Editorial, 2000.

**[Durán y Ruiz, 2000]** Durán, A., Ruiz A., Corchuelo R. y Toro M., *Identificación de Patrones de Reutilización de Requisitos de Sistemas de Información*, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, 2000.

**[Escalona, 2004]** Escalona Cuaresma, M. J., *Modelos y Técnicas para la Especificación y el Análisis de la Navegación en Sistemas Software*, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Octubre, 2004.

**[Myer, 2012]** Myer, T., *Phonegap*, Anaya Multimedia, Marzo, 2012.

**[Ribas, 2016]** Ribas Lequerica, J., *Desarrollo De Aplicaciones Para Android*, Edición 2016, Anaya Multimedia, Mayo, 2015.

## REFERENCIAS WEB

***Enough Software. Guía Galaxia de Aplicaciones Móviles, 15º Ed. (2015)***  
[http://enough.de/fileadmin/user\\_upload/MobileDevelopersGuide\\_15th\\_Castellano.pdf](http://enough.de/fileadmin/user_upload/MobileDevelopersGuide_15th_Castellano.pdf)

***Guía de Evaluación Heurística de Sitios Web***  
<http://www.nosolousabilidad.com>

***Phonegap API Documentation***  
<http://docs.phonegap.com/en/2.9.0/index.html>

***RaspberryPi Community***  
<https://www.raspberrypi.org/community>

***Recursos para desarrolladores en Android***  
<http://developer.android.com/index.html>

**ESTA PARTE QUEDA  
PROTEGIDA Y NO SE  
PUBLICA POR INTERESES  
PROFESIONALES DEL  
AUTOR**





## 2 MANUAL DE INSTALACIÓN Y CONFIGURACIÓN

A continuación se describen los pasos a seguir para una correcta instalación del sistema desarrollado. También se describe el software adicional necesario para su funcionamiento óptimo. La instalación se divide en 2 partes: instalación de la base de datos en el servidor e instalación de los componentes necesarios para poder acceder a la aplicación móvil.

### 2.1 INSTALACIÓN DEL SERVIDOR Y CLIENTE

#### 2.1.1 Instalación del Servidor de Bases de Datos

##### 2.1.1.1 Instalación de Apache

- 1) Actualizamos los repositorios para realizar la instalación:  

```
sudo apt-get update
```
- 2) Instalamos apache con php5 y librerías de php5  

```
sudo apt-get install apache2 php5 libapache2-mod-php5
```
- 3) Reiniciar servicio de Apache:  

```
sudo /etc/init.d/apache2 restart
```

##### 2.1.1.2 Instalación de MySQL

- 1) El primer paso que se realizará será activar la interfaz *loopback* ya que si no se realiza nos dará un error al instalar MySQL:  

```
sudo ifup lo
```
- 2) Instalamos los paquetes necesarios del SGBD MySQL:  

```
sudo apt-get install mysql-server mysql-client php5-mysql phpmyadmin
```
- 3) Al instalarse nos aparecerá una pantalla de configuración preguntando la contraseña que queremos usar. Después de la instalación añadiremos en el archivo de configuración: `php.ini`, antes de la línea donde se encuentra este texto "*Dynamics Extensions*" la siguiente línea: `extension=mysql.so`:  

```
sudo nano /etc/php5/apache2/php.ini
```
- 4) Tras modificar el archivo escribimos los siguientes comandos para crear un enlace simbólico:  

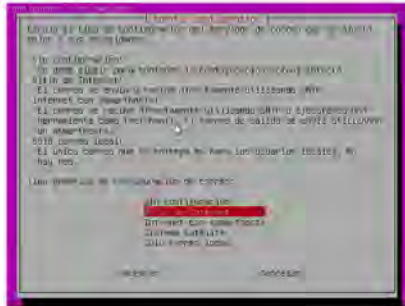
```
sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf.d/phpmyadmin.conf
```
- 5) Reiniciar servicio de Apache:  

```
sudo /etc/init.d/apache2 restart
```

## 2.1.2 Procedimiento de Instalación de los Servicios de Comunicación en Red

### 2.1.2.1 Instalación de Postfix

Postfix es un servidor de correo que se utiliza como servidor de correo remitente (MTA). Postfix se encarga de recibir el correo y lo coloca en la cola de mensajes para enviar utilizando el protocolo SMTP. Su directorio principal, donde trabajaremos y que aparecerá tras su instalación en Raspbian con `sudo apt-get install postfix` es: `/etc/postfix`



**PASO 1:**  
En el servidor se selecciona para "Sitio de Internet" tras:  
`sudo apt-get install postfix`

Se define como nombre del sistema de correo: `uoc.es` y pulsar el botón *Aceptar*. (Posteriormente el dominio especificado se utilizará los servicios POP3/IMAP del paquete `dovecot`)

### 2.1.2.2 Configuración de Postfix

Postfix ofrece por salida en pantalla todos los parámetros de configuración o directivas que se vayan actualizando en el fichero de configuración general: `/etc/postfix/main.cf` mediante el comando: `postconf`.

Los parámetros de la configuración se pueden cambiar editando el fichero `main.cf` ó accediendo a la reconfiguración del paquete postfix con : `sudo dpkg-reconfigure postfix`. El fichero `main.cf` generado incluye además las directivas necesarias para la conexión con el Directorio LDAP y su autenticación con el servidor POP3/IMAP; Dovecot. Cualquier modificación de los parámetros de configuración del fichero podrá necesitar del reinicio del servicio postfix para hacer persistentes los cambios realizados p.e. con: `sudo service postfix restart` ó sin necesidad de parar el servicio del mismo pero leyendo la configuración con: `sudo postfix reload`

```
lned@mvserverid:~$ cat /etc/mailname
```

**PASO 2:** Comprobar que el nombre de correo de `uoc.es` está establecido en el fichero `/etc/mailname` con:  
`cat /etc/mailname`

```
lned@mvserverid:~$ cat /etc/mailname
postfix
lned@mvserverid:~$ sudo netstat -tlnp | grep :25
tcp        0      0 0.0.0.0:25                0.0.0.0:25                *
lned@mvserverid:~$
```

**PASO 3:** Comprobar que el servicio postfix está levantado y el puerto SMTP TCP/25 activo:

```
sudo /etc/init.d/postfix status
sudo netstat -an | grep :25
telnet mvserverid 25
```

```
lned@mvserverid:~$ postconf -n
alias_database = hash:/etc/aliases
aliases_file = /etc/aliases
append_dot_mydomain = no
biff = no
config_directory = /etc/postfix
inet_interfaces = all
inet_protocols = all
mailbox_size_limit = 0
mydestination = mvserverid, mvserverid.localdomain, localhost
myhostname = mvserverid
mynetworks = 127.0.0.0/8 [::ffff:0:0:0:0/96] ::ffff:0:0:0:0/96
myorigin = /etc/mailname
recipient_delimiter = -
relayhost =
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
smtpd_banner = 2016 Dovecot SMTP Service (Ubuntu)
smtpd_tls_cert_file = /etc/postfix/cert/cert.pem
smtpd_tls_key_file = /etc/postfix/cert/cert-key.pem
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtpd_use_tls = yes
```

Figura 48: Configuración de Postfix

**/etc/postfix/main.cf**

```

...
# Nombre del servidor (también con: smtp.mczones.es)
myhostname = myservidor

# Enmascaramiento del correo (nombre de correo:
mczones.es)
myorigin = /etc/mailname

# Destinatarios globales de los correos en función
del dominio
mydestination = ucc.es, myservidor,
localhost, localdomain, localhost

# Determina a que redes o hosts se permite el correo
saliente
# (no es desde otro servidor SMTP, sino desde el
propio Postfix)
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104
[::1]/128 192.168.1.0/24

# Límite del buzón del correo ilimitado
mailbox_size_limit = 0
recipient_delimiter = +

# Conexión desde cualquier interface de la red y protocolos Ipv4/v6
inet_interfaces = all
inet_protocols = all
...

```

Red local de máquinas desde donde se envían los correos

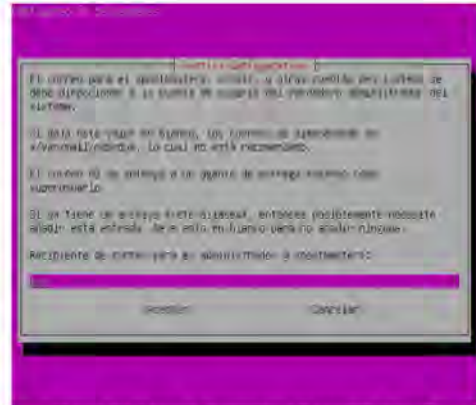


Figura 49: Reconfiguración del paquete Postfix

**2.1.2.3 Instalación de Dovecot**

Dovecot es un servidor POP3/IMAP y sus versiones cifradas (POP3s/IMAPs) el cual necesita de un MTA como Postfix para trabajar de forma adecuada. Dovecot se encarga del reparto del correo a los usuarios a través de estos protocolos. Las conexiones realizadas no cifradas aparecerán en las últimas líneas del log: `/var/log/mail.log`. Los pasos necesarios para su instalación son:

```

myservidor:~$ sudo apt-get install dovecot-postfix dovecot-ldap dovecot-
imap
...

```

**PASO 1:** El servidor se prepara para instalar el conjunto de paquetes que representa dovecot-postfix:

```

sudo apt-get install dovecot-postfix dovecot-imapd dovecot-pop3d dovecot-
ldap

```

(instala por dependencias: `dovecot-common` entre otros paquetes)

```

myservidor:~$ netstat -an | grep 110
tcp        0      0 0.0.0.0:110          0.0.0.0:*           LISTEN
tcp        0      0 0.0.0.0:143         0.0.0.0:*           LISTEN
myservidor:~$ netstat -an | grep 993
tcp        0      0 0.0.0.0:993         0.0.0.0:*           LISTEN
tcp        0      0 0.0.0.0:993         0.0.0.0:*           LISTEN

```

**PASO 2:** Comprobar que el servicio dovecot está levantado en los puertos 110/tcp y 995/tcp, correspondientes a POP3 / POP3s y 143/tcp y 993/tcp, correspondientes a IMAP/IMAPs:

```

sudo netstat -an | grep :110
sudo netstat -an | grep :143

```

¿proceso en ejecución?: `ps -aux | grep dovecot`

**PASO 3:** Si el servicio Dovecot está corriendo, intentar logear y hacer telnet a los puertos de POP3 e IMAP con:

```

telnet localhost pop3
telnet localhost imap2

```

**ESTA PARTE QUEDA  
PROTEGIDA Y NO SE  
PUBLICA POR INTERESES  
PROFESIONALES DEL  
AUTOR**

