



GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO FINAL DE GRADO

---

**Desarrollo de servicio de cita previa  
TurnoGes**

---

*Autor:*  
Carlos GARCÍA GARCÍA

*Supervisor:*  
José Guzmán FELIU VIVAS  
*Tutor académico:*  
Luis Amable GARCÍA FERNÁNDEZ

Fecha de lectura: 9 de Septiembre de 2016  
Curso académico 2015/2016

## **Resumen**

En este documento se recogen los procesos realizados para desarrollar un proyecto para la asignatura EI 1054: Prácticas externas y proyecto final de grado. El proyecto se ha realizado en la entidad Caixa Almassora y consiste en el desarrollo de un sistema de cita previa integrable con el sistema de turnos TurnoGes que posee la entidad. El sistema permitiría a los clientes pedir una cita para acudir a la entidad bancaria desde Internet. En el documento se recoge una descripción detallada del proyecto que se ha realizado, la planificación que se ha seguido para realizarlo, los detalles que se han tenido en cuenta a la hora de su desarrollo, las decisiones que se han tomado durante el proyecto y también muestras del resultado final obtenido.

## **Palabras clave**

Cita previa, calendario, gestión, PHP

## **Keywords**

Appointment, schedule, management, PHP

# Índice general

<b>1. Introducción</b>	<b>11</b>
1.1. Contexto y motivación del proyecto . . . . .	11
1.2. Objetivos del proyecto . . . . .	12
1.3. Alcance . . . . .	13
1.3.1. Alcance organizativo . . . . .	13
1.3.2. Alcance informático . . . . .	13
1.3.3. Alcance funcional . . . . .	13
<b>2. Descripción del proyecto</b>	<b>15</b>
2.1. Características del proyecto . . . . .	15
2.2. Tecnologías utilizadas . . . . .	17
2.2.1. Lado del servidor y persistencia de datos . . . . .	18
2.2.2. Lado del cliente . . . . .	18
2.2.3. Otras herramientas . . . . .	20
<b>3. Planificación del proyecto</b>	<b>21</b>
3.1. Metodología . . . . .	21
3.2. Planificación . . . . .	23
3.3. Estimación de recursos y costes del proyecto . . . . .	27

3.3.1.	Recursos humanos . . . . .	27
3.3.2.	Recursos software . . . . .	27
3.3.3.	Recursos hardware . . . . .	28
3.4.	Seguimiento del proyecto . . . . .	28
3.4.1.	Plan de gestión de riesgos . . . . .	28
<b>4.</b>	<b>Análisis y diseño del sistema</b>	<b>35</b>
4.1.	Análisis y definición de requisitos . . . . .	35
4.1.1.	Diagrama de casos de uso (DCU) . . . . .	35
4.1.2.	Requisitos de datos . . . . .	41
4.2.	Diseño de la arquitectura del sistema . . . . .	43
4.2.1.	Modelo conceptual de datos . . . . .	43
4.2.2.	Diagrama lógico de la base de datos . . . . .	44
4.2.3.	Diseño de la interfaz . . . . .	45
<b>5.</b>	<b>Implementación y pruebas</b>	<b>57</b>
5.1.	Detalles de implementación . . . . .	57
5.1.1.	Instalación del servidor de turnos . . . . .	57
5.1.2.	Desarrollo de la aplicación . . . . .	58
5.2.	Verificación y validación . . . . .	66
5.3.	Funcionamiento de la aplicación . . . . .	66
<b>6.</b>	<b>Conclusiones</b>	<b>69</b>
<b>A.</b>	<b>Manual de usuario</b>	<b>73</b>
<b>B.</b>	<b>Documentación de los ficheros y clases</b>	<b>75</b>
B.1.	Ficheros de inicio, registro y login . . . . .	76

B.2. Ficheros de conexión y acceso a la base de datos . . . . .	78
B.3. Ficheros para envío de correos . . . . .	79
B.4. Ficheros de recursos . . . . .	80
B.5. Ficheros de la parte de cliente . . . . .	82
B.6. Ficheros de la parte de gestión . . . . .	85



# Índice de cuadros

3.1. Identificación de riesgos . . . . .	29
3.2. Análisis de R01: Requisitos poco claros . . . . .	30
3.3. Análisis de R02: Falta de experiencia en las tareas de planificación . . . . .	31
3.4. Análisis de R03: Falta de experiencia con las herramientas utilizadas . . . . .	31
3.5. Análisis de R04: Conflictos entre los implicados en el proyecto . . . . .	32
3.6. Análisis de R05: Falta de seguimiento de las tareas . . . . .	32
3.7. Análisis de R06: Cambio de tecnologías . . . . .	33
3.8. Acciones de prevención y corrección . . . . .	34
4.1. Identificación de actores del sistema . . . . .	36
4.2. Descripción del caso de uso CU01 . . . . .	37
4.3. Descripción del caso de uso CU02 . . . . .	37
4.4. Descripción del caso de uso CU03 . . . . .	37
4.5. Descripción del caso de uso CU04 . . . . .	38
4.6. Descripción del caso de uso CU05 . . . . .	38
4.7. Descripción del caso de uso CU06 . . . . .	38
4.8. Descripción del caso de uso CU07 . . . . .	39
4.9. Descripción del caso de uso CU08 . . . . .	39
4.10. Descripción del caso de uso CU09 . . . . .	39

4.11. Descripción del caso de uso CU10 . . . . .	40
4.12. Descripción del caso de uso CU11 . . . . .	40
4.13. Descripción del caso de uso CU12 . . . . .	40
4.14. Descripción del requisito de datos RD01 . . . . .	41
4.15. Descripción del requisito de datos RD02 . . . . .	41
4.16. Descripción del requisito de datos RD03 . . . . .	41
4.17. Descripción del requisito de datos RD04 . . . . .	42
4.18. Descripción del requisito de datos RD05 . . . . .	42
4.19. Descripción del requisito de datos RD06 . . . . .	42
4.20. Descripción del requisito de datos RD07 . . . . .	42
4.21. Descripción del requisito de datos RD08 . . . . .	42
4.22. Descripción de las tablas de la base de datos del proyecto . . . . .	43

# Índice de figuras

2.1. Tecnologías utilizadas para la parte del servidor . . . . .	18
2.2. Tecnologías utilizadas para la parte del cliente . . . . .	19
3.1. Niveles de costo y dotación típicos durante el proyecto. . . . .	22
3.2. Diagrama de Gantt inicial . . . . .	25
3.3. Diagrama de Gantt final . . . . .	26
4.1. Diagrama de casos de uso . . . . .	36
4.2. Modelo conceptual de la base de datos. . . . .	44
4.3. Modelo lógico de la base de datos. . . . .	44
4.4. Diseño de la pantalla de registro y login. . . . .	46
4.5. Diseño de la pantalla de inicio de clientes. . . . .	47
4.6. Diseño de la pantalla de inicio de clientes con una búsqueda realizada. . . . .	48
4.7. Diseño de la pantalla de confirmación de reservas. . . . .	49
4.8. Diseño de la pantalla de gestión de citas del cliente. . . . .	50
4.9. Pantalla de información de la cita seleccionada. . . . .	51
4.10. Pantalla de edición de perfil. . . . .	52
4.11. Diseño de la pantalla de inicio de administradores. . . . .	53
4.12. Pantalla de gestión de colas en administración. . . . .	54
4.13. Pantalla de gestión de calendarios en administración. . . . .	55

5.1. Capa de abstracción PDO para PHP. . . . .	58
5.2. Código de conexión con la base de datos. . . . .	59
5.3. Parte del código del DAO de calendarios. . . . .	60
5.4. Diagrama del patrón MVC. . . . .	60
5.5. Parte del código del controlador de gestión de centros. . . . .	61
5.6. Sistema reCAPTCHA de Google. . . . .	63
5.7. Plugin date-picker de JQuery. . . . .	63
5.8. Plugin time-picker de JQuery. . . . .	63
5.9. Plugin Monthly Calendar. . . . .	64
5.10. Plugin Monthly Calendar al hacer clic en un día. . . . .	65
5.11. Plugin Monthly Calendar al hacer clic en un a cita. . . . .	65
5.12. Página de gestión. . . . .	67
5.13. Página de selección de cita. . . . .	68
B.1. Estructura de ficheros del directorio raíz. . . . .	75

# Capítulo 1

## Introducción

### 1.1. Contexto y motivación del proyecto

La entidad Caixa Almassora se trata de una entidad cooperativa de crédito en la que se pueden realizar toda clase de operaciones comunes a las entidades financieras que integran el sistema financiero Español.

Caixa Almassora cuenta con un sistema de gestión de turnos muy innovador al servicio de sus clientes llamado TurnoGes©. Este sistema permite organizar las colas de los clientes cuando esperan a ser atendidos, y consigue hacerlo de una manera muy eficiente y a bajo coste. El sistema está instalado en cada una de las entidades Caixa Almassora de la población por lo que todos sus clientes pueden disfrutar del servicio acudiendo a su entidad más cercana.

El proyecto que desea realizar la entidad es un proyecto que se desarrolla en los ámbitos de gestión y atención al cliente. Se trata de un sistema de cita previa por internet que permita a los usuarios concretar citas para el servicio que deseen y hacerlo cómodamente desde sus casas a través de Internet. Además, se quiere conseguir que este sistema de cita previa se pueda integrar perfectamente con el actual gestor de turnos para que los clientes que han pedido cita por internet sean gestionados de igual manera que el resto de clientes cuando lleguen a la entidad el día de la cita, por tanto el sistema de citas deberá estar lo más adaptado posible para poder integrarlo con TurnoGes©.

Por otro lado también se ha de desarrollar la parte de gestión interna del sistema de cita previa de tal manera que permita a los administradores de la entidad configurar todos los aspectos relacionados con las citas previas para que los clientes puedan disfrutar del servicio correctamente.

## 1.2. Objetivos del proyecto

El objetivo fundamental del proyecto es ampliar el sistema de citas que la entidad posee en la actualidad para permitir pedir cita desde internet además de desde la propia entidad, ofreciendo así un servicio mejor y con más posibilidades que consiga una mayor satisfacción por parte de sus clientes y un mejor prestigio frente a otras entidades de la competencia. Para lograrlo es necesaria la implantación del sistema y conseguir que la experiencia de los usuarios frente al servicio sea cómoda y sencilla.

Un sistema de cita previa como este ofrecería al cliente la comodidad de poder concertar citas con mayor libertad gracias a la posibilidad de elegir la fecha y la hora que más le convenga (dentro de las disponibles) y además seleccionar el servicio por el cual quiere ser atendido dentro de los que la entidad ofrece.

Por otro lado, el hecho de que el trabajador de la entidad sepa con antelación el motivo por el cual va a acudir el cliente a hablar con él, permite una mayor eficiencia a la hora de resolver cualquier conflicto o duda que el cliente pueda tener provocando también una mayor satisfacción del cliente.

Por tanto podemos definir los objetivos específicos del proyecto como:

- Implantar de sistema de cita previa cómodo y funcional para el cliente.
- Implantar la parte interna del sistema de cita previa de forma que se pueda integrar con el sistema TurnoGes©.

Y podemos definir los objetivos generales del proyecto como:

- Aumentar la cantidad de servicios ofrecidos por la entidad.
- Mejorar la satisfacción del cliente.
- Mejorar la petición de citas.
- Mejorar la eficiencia de la entidad respecto a los servicios ofrecidos.

También cabe destacar los objetivos personales que se plantean para este proyecto que son los siguientes:

- Gestionar un proyecto software en todas sus fases, un objetivo importante para adquirir experiencia en este aspecto.
- Aprender nuevas tecnologías. Siempre es importante conocer un gran abanico de tecnologías y saber adaptarse a cualquier tecnología y entorno que se proponga.
- Adaptarse al entorno de trabajo. También es importante como objetivo ya que la capacidad de adaptación es un requisito fundamental para todas las empresas.

## 1.3. Alcance

A continuación se muestra el alcance del proyecto desde el punto de vista tanto organizativo como informático y funcional.

### 1.3.1. Alcance organizativo

Desde el punto de vista organizativo podemos definir el alcance de la siguiente forma:

- Dentro de la organización:
  - Los administradores de SSCC (servicios centrales) interactúan con el sistema para mantener la gestión de los centros y usuarios que interactúan con el sistema.
  - Los administradores de cada centro interactúan con el sistema para mantener la información de las citas previas disponibles actualizadas definiendo las fechas disponibles etc.
- Fuera de la organización: Los clientes interactúan con el sistema para aprovechar los servicios que ofrece la entidad.

### 1.3.2. Alcance informático

La propuesta de creación del sistema de cita previa cubre los siguientes puntos:

- Construcción del sistema de gestión de citas integrable con TurnoGes© (back-office).
  - Desarrollo de la base de datos.
  - Desarrollo de interfaces intuitivas.
  - Desarrollo de algoritmos y procedimientos para la gestión de citas disponibles.
  - Realizar el desarrollo de la aplicación de manera que se pueda integrar perfectamente con el sistema de turnos.
- Construcción de la parte del cliente del servicio (front-office).
  - Desarrollo de interfaces intuitivas.
  - Desarrollo de algoritmos y procedimientos para la concertación de citas.

### 1.3.3. Alcance funcional

Desde el punto de vista funcional la propuesta de creación del sistema de cita previa cubre los siguientes puntos:

- Permitirá a los administradores de SSCC la gestión de centros y usuarios así como la gestión de cualquier otro tipo de información relacionada con el sistema de cita previa.
- Permitirá a los administradores de los centros la gestión de fechas disponibles para la concertación de citas.
- Permitirá a los usuarios la concertación de citas desde Internet para los servicios que deseen.
- Permitirá a los usuarios la cómoda gestión de su turno cuando lleguen al centro mediante el sistema de gestión de turnos TurnoGes©.

## Capítulo 2

# Descripción del proyecto

### 2.1. Características del proyecto

Como ya se ha mencionado brevemente en el capítulo anterior, el proyecto consiste en la implantación de un sistema de cita previa por internet para la entidad bancaria Caixa Almassora. A continuación se explicará con mayor profundidad el proyecto que se va a desarrollar y las tecnologías que se utilizarán para ello.

Para entender mejor el contexto del proyecto es importante entender una serie de conceptos sobre el actual sistema de turnos.

TurnoGes©[6] es uno de los sistemas de gestión de turnos más avanzados del mercado. Se trata de un sistema multi idioma diseñado para gestionar y regular las colas que a menudo se producen en las oficinas de atención al público. El sistema ofrece herramientas eficientes para gestionar y optimizar los recursos destinados a la atención al cliente, consiguiendo que la espera sea más placentera.

Funciona de la siguiente manera: El cliente llega al centro y obtiene un ticket donde se indica el servicio que ha solicitado y su turno. El cliente espera a que sea llamado. Cuando su turno aparece en pantalla y es avisado mediante un sistema de voz, el cliente acude al lugar de atención indicado.

El sistema consta de una serie de módulos que permiten su funcionamiento:

- **Dispensador:** el dispensador de turnos es un dispositivo situado en las oficinas en el que los clientes seleccionan el servicio por el que desean ser atendidos de entre 10 opciones. Internamente, en el sistema, estos servicios son configurados como colas. Cada servicio tiene una cola asignada en la que también se detalla el tiempo medio de espera, el horario, la mesa etc. El dispensador también se encarga de imprimir el ticket para el usuario.
- **Sistema multimedia:** se trata de pantallas que se instalan en el patio de operaciones de las oficinas y donde se muestra información sobre el estado de las colas como los tres últimos

turnos atendidos, su mesa y otra información adicional.

- Display de puesto: este dispositivo se encuentra en las mesas de los empleados que se encargan de recibir a los clientes y se encarga de mostrar el identificador de la mesa y el número del turno.
- Terminal virtual: se trata de una aplicación instalada en los ordenadores de la oficina, ofreciendo al puesto de atención varias posibilidades de control del turno, así como la información necesaria para poder dar una mejor respuesta a sus clientes.

Uno de los objetivos del proyecto es conseguir el sistema de cita previa sea perfectamente integrable con este sistema de turnos de tal forma que los clientes que pidan cita por internet sean gestionados en la cola al llegar al centro de igual manera que los clientes que piden la cita desde el dispensador en el propio centro.

Para ello se deberá realizar el diseño del sistema de tal forma que todos los datos de las citas previas puedan ser exportados al sistema de turnos de manera automática diariamente para que puedan ser añadidos a los registros del sistema de turnos y puedan ser gestionados de manera normal. En capítulos posteriores se especificará más sobre esta estrategia.

Así pues, el proyecto consta de 2 partes fundamentales y diferenciales:

## 1. Desarrollo del Back-office.

El back-office, o la parte de gestión interna, es la parte del sistema que mantiene el acceso restringido a los administradores. Esta parte del sistema permite que los administradores mantengan actualizado y configurado el sistema de cita previa para que los usuarios puedan tener las citas previas siempre disponibles.

En el back-office encontraremos dos roles distintos, superusuarios (administradores de SSCC, servicios centrales) y administradores (los administradores de los distintos centros).

Una vez iniciada la sesión con una cuenta de este tipo, las funcionalidades que el sistema ofrecerá a todo tipo de administradores son:

- Mantenimiento de Horarios: Los administradores podrán configurar los horarios de la entidad y en los que se dispondrá de servicio.
- Mantenimiento de Colas: como se ha comentado, se le llama colas a los servicios que ofrece la entidad a sus clientes. para cada servicio que se quiera ofrecer los administradores podrán configurar una cola.
- Mantenimiento de Periodos: se podrán gestionar periodos temporales asignándolos a un horario. Los periodos permitirán a los administradores establecer el espacio temporal para el que se quiere ofrecer el servicio a los clientes.
- Mantenimiento de Horas: se podrán configurar los días y las horas en las que estarán disponibles los servicios dentro de un periodo.
- Mantenimiento de Festivos: se podrán introducir la fechas festivas del año. La configuración de festivos permitirá a los administradores evitar la introducción de citas disponibles en días festivos.

- **Mantenimiento de Calendarios:** por último, los administradores podrán gestionar las citas previas para que estén disponibles a los clientes y podrán examinarlas de manera visual a través de calendarios.

Si la sesión es iniciada con una cuenta de superusuario (administradores de SSCC), estos dispondrán de una serie de funcionalidades adicionales:

- **Mantenimiento de Centros:** los administradores de SSCC podrán actualizar los centros que ofrecerán el servicio de cita previa.
- **Mantenimiento de Usuarios:** podrán gestionar los usuarios registrados en el sistema, tanto administradores como clientes registrados.

Tanto el mantenimiento de periodos, como el de horas y festivos son un sistema de limitación y automatización para que los administradores, al introducir citas para la disponibilidad de los clientes, lo puedan hacer de manera cómoda, rápida y sin cometer errores. En capítulos posteriores se describirá el funcionamiento de la aplicación para entenderlo mejor.

## 2. Desarrollo del front-office.

Por otro lado, en el front-office o la parte directamente relacionada con el cliente, los usuarios podrán concertar las citas que deseen así como también acceder y modificar su perfil y acceder a la gestión de citas. En esta sección podrán consultar tanto las citas previas solicitadas pendientes como las citas previas solicitadas y “caducadas” o pasadas. Aquí también podrán cancelar aquellas citas que deseen con lo que volverán a estar disponibles para otros usuarios.

En el front-office únicamente contamos con el rol de cliente y en general, las funcionalidades que podrá realizar son las siguientes:

- **Registro y login:** el cliente podrá registrarse y acceder al sistema siempre que los datos introducidos sean válidos.
- **Búsqueda de citas previas:** el cliente podrá buscar citas para el día y la hora que desee y comprobar si existen o no citas con las características introducidas.
- **Concertar citas previas:** el cliente podrá concertar una cita si se adapta a sus preferencias y esta pasará a mostrarse en la sección de gestión de citas del usuario.
- **Mantenimiento de citas del usuario:** el usuario podrá consultar todas las citas que haya reservado tanto las pendientes como las pasadas y podrá cancelar o cambiar aquellas citas pendientes que no desee realizar.
- **Editar perfil:** por último, el cliente podrá modificar los datos de su perfil siempre que lo desee.

## 2.2. Tecnologías utilizadas

Para desarrollar el proyecto se han utilizado diversas tecnologías tanto en el lado del cliente como del servidor. A continuación se detallan dichas tecnologías y se justifica su elección:

### 2.2.1. Lado del servidor y persistencia de datos

Para la parte del servidor se ha utilizado PHP corriendo en un servidor Apache. En este caso la elección del lenguaje fue un requisito de la entidad ya que es el lenguaje que se ejecuta en su alojamiento web.

PHP: Hypertext Preprocessor[10], es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Por tanto es un lenguaje que se adapta perfectamente a las características de este proyecto.

Cabe destacar que no se ha utilizado ningún framework PHP para realizar el proyecto siguiendo las indicaciones de la entidad ya que se ha considerado que las especificaciones del proyecto no requerían de la utilización de ningún tipo de framework para PHP.

Para mantener la consistencia de los datos y poder realizar operaciones con ellos es necesaria la implantación de una base de datos. Para ello se ha decidido utilizar MySQL[11], un sistema gestor de base de datos relacional, de código abierto, fiable, robusto y concurrente.

Durante el desarrollo del proyecto se ha utilizado XAMPP un servidor independiente de plataforma de código libre que incluye los paquetes necesarios para realizar un proyecto con dichas características. Estos paquetes son: los intérpretes de lenguajes PHP y Perl, los sistemas gestores de base de datos MySQL y su derivado MariaDB y el servidor web Apache. Además ofrece una herramienta visual para la gestión de bases de datos MySQL llamada PHPMYAdmin que simplifica mucho la gestión.



Figura 2.1: Tecnologías utilizadas para la parte del servidor

### 2.2.2. Lado del cliente

Las tecnologías de la parte del cliente son las que permiten al usuario visualizar las distintas interfaces y se ejecutan directamente sobre el navegador web del usuario que ejecuta la aplicación.

Para este proyecto se han utilizado las siguientes tecnologías base:

- El lenguaje de marcado estándar para la elaboración de páginas web, HTML (HyperText

Markup Language), en concreto la versión 5.

- También se ha utilizado CSS (cascading style sheets), en su versión 3, un lenguaje utilizado para definir y crear la presentación de un documento estructurado en HTML.
- Por último se ha utilizado Javascript un lenguaje de programación interpretado que permite la interacción con la interfaz de usuario consiguiendo así un resultado dinámico en la página web.

Además se han utilizado diversos frameworks y librerías para conseguir un aspecto visual más actual, estilizado y dinámico. Estos son:

- Bootstrap [2] (v3.3.6): un framework HTML, CSS y JS en auge y muy popular en la actualidad, que permite el desarrollo de interfaces de usuario con un aspecto muy acabado y profesional. Sus principales características son:
  - Su sistema de rejillas de 12 columnas que permite la estructuración de la página web de manera muy sencilla.
  - Su diseño responsive que permite realizar un diseño que se adapte a los distintos tipos de dispositivos más utilizados hoy en día como tablets, teléfonos o monitores.
  - Su gran cantidad de complementos como barras de navegación, contenedores, menús, botones etc.
  - Además cuenta con una gran cantidad de documentación.

En general se trata de un framework muy útil que facilita en gran medida el desarrollo de interfaces de usuario.

- Librería JQuery[7]: Una librería de javascript rápida, pequeña y con muchas características que permite realizar de manera sencilla la manipulación de documentos HTML mediante eventos, animaciones etc. Además contiene una API de AJAX muy fácil de utilizar para la recarga parcial de contenido. Funciona con la mayoría de navegadores y es muy versátil y extensible.
- Librería JQuery UI[8]: Una librería de javascript para interfaces de usuario que cuenta con un conjunto de componentes como efectos, widgets, temas etc. y está construida sobre la librería JQuery. Una opción perfecta para aplicaciones web interactivas.



Figura 2.2: Tecnologías utilizadas para la parte del cliente

### 2.2.3. Otras herramientas

Debido a la política existente en la entidad de utilizar software libre, para desarrollar la aplicación se ha utilizado un editor de texto llamado Brackets. Se trata de un editor de texto moderno, con la característica de que entiende el diseño web por lo que se puede utilizar perfectamente como un entorno de desarrollo integrado para desarrollo web y permite estructurar perfectamente documentos de tipo PHP, html, css o javascript como se haría con cualquier otro entorno de programación. Además al ser un software de código abierto, existen una gran cantidad de complementos muy útiles que la gente ha desarrollado gratuitamente.

Para la redacción de esta memoria se han utilizado herramientas como ShareLatex [5] un editor Latex online que permite la creación de proyectos Latex con mucha facilidad sin tener que hacer instalaciones previas y permite almacenarlos en sus servidores. También se han utilizado herramientas como Creately [4] o MS Project para el diseño de diagramas.

## Capítulo 3

# Planificación del proyecto

### 3.1. Metodología

Para la planificación del proyecto se ha seguido la metodología tradicional en cascada, desglosando y planificando temporalmente las tareas antes de comenzar con el desarrollo del proyecto. Para ello he decidido seguir el estándar PMBOK de gestión de proyectos que se ha estudiado durante la carrera. PMBOK[12] es un estándar en la gestión de proyectos desarrollado por el PMI (Project Management Institute) que nos permite aplicar las técnicas y herramientas necesarias para cumplir los requisitos del proyecto.

Un proyecto se define según PMBOK como “un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único”. Un proyecto implica un inicio y un final por naturaleza, se puede llegar a este final o bien porque los objetivos han sido cumplidos y se ha completado con satisfacción el proyecto, o bien porque se detecta que los objetivos no se cumplirán o no pueden ser cumplidos. PMBOK, define una serie de herramientas, conocimientos, procedimientos y técnicas que nos facilitan llegar al final del proyecto de manera exitosa.

El estándar define 5 procesos básicos que tienen lugar durante el ciclo de vida del proyecto y pueden ser aplicados a cualquier proyecto sin importar su dimensión. Estos son:

- Fase de Inicio
- Fase de Planificación
- Fase de Ejecución
- Fase de Seguimiento y Control (se aplica durante todo el ciclo)
- Cierre

En la siguiente imagen se muestra el nivel de coste y dotación del personal a medida que se avanza en las distintas fases del proyecto:

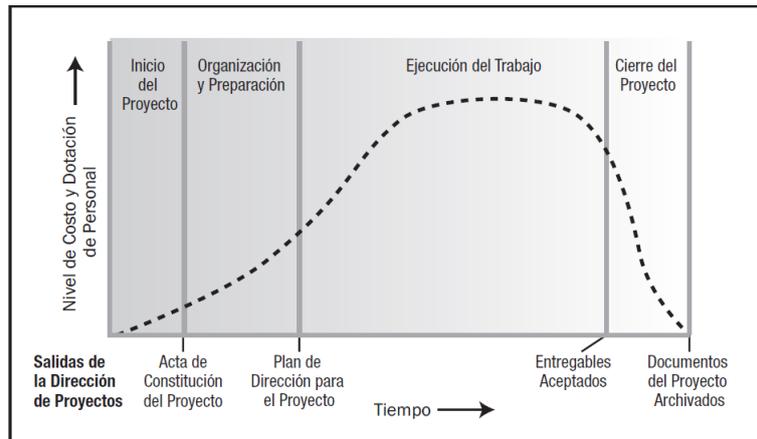


Figura 3.1: Niveles de costo y dotación típicos durante el proyecto.

Como se puede apreciar en la figura, los niveles de coste y dotación son bajos en el inicio y el cierre del proyecto y alcanzan su punto máximo durante su desarrollo.

Los objetivos de las distintas fases de un proyecto según la metodología PMBOK son los siguientes:

- En la fase de inicio se pretende clarificar el problema que se va a solucionar, el producto que se va a obtener o el servicio que se va a ofrecer.
- Durante la fase de planificación, el objetivo es definir de manera clara las tareas que se van a realizar, los roles del equipo y cargas de trabajo y las planificaciones temporales y de coste que van a limitar el proyecto. Además se definen los planes de gestión de riesgos y de calidad que permitan llevar un seguimiento del proyecto y comprobar que se sigan los estándares definidos.
- En la fase de desarrollo se pone en marcha el trabajo y se realizan todas las tareas específicas del proyecto que permiten obtener el resultado esperado.
- El seguimiento y control es una fase que se aplica durante todo el ciclo de vida del proyecto y que tiene como objetivo comprobar que se estén siguiendo los estándares de calidad definidos y gestionar los riesgos que puedan suceder durante el proyecto para garantizar el avance del proyecto de manera exitosa.
- Por último en el cierre se realiza la entrega final de producto o servicio y se realiza la documentación necesaria.

Por tanto, dentro de cada fase del ciclo de vida del proyecto, se definen una serie de procesos que pueden variar según la naturaleza del proyecto. En el próximo apartado se mostrará una definición de las distintas fases y procesos que se han seguido para desarrollar este proyecto.

Cabe destacar que he optado por seguir el estándar PMBOK para desarrollar el proyecto ya que en primer lugar, no se ha propuesto ninguna metodología específica en el lugar de trabajo. Además, el estándar PMBOK se ha utilizado con anterioridad para otros proyectos informáticos

a lo largo de la carrera universitaria, por tanto, ya se ha comprobado su eficacia y se tiene experiencia con él. Una de sus principales ventajas es que puede ser adaptada a cualquier tipo de proyecto, además ha sido probada por muchos profesionales en distintos proyectos y, como se ha comprobado en otras ocasiones, permite obtener una documentación clara y detallada, fundamental para seguir el proceso de desarrollo con orden y poder cumplir con los objetivos definidos al final del proyecto.

## 3.2. Planificación

Como describe el estándar PMBOK durante la fase de planificación, uno de los procesos que se ha de realizar es la planificación temporal. La planificación temporal permite desglosar las distintas tareas que se han de realizar para completar con éxito el proyecto. Para ello, se ha utilizado la herramienta del diagrama de Gantt.

El diagrama de Gantt es una herramienta visual que tiene el objetivo de exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado. Es una herramienta muy utilizada en la gestión de proyectos debido a su facilidad de lectura, las tareas se representan mediante barras horizontales mientras que las columnas muestran la escala temporal.

El diagrama de Gantt del proyecto se puede ver en la figura [3.2]. Como se puede observar el diagrama incluye el desglose de tareas tanto del desarrollo del proyecto como de la redacción de esta memoria junto a su estimación temporal en horas, producen un total de 373 horas. Se han destinado 300h al desarrollo del proyecto en la entidad tal y como especifica la normativa y 73h horas a la redacción de la memoria y presentación del proyecto. Se ha de tener en cuenta de que se ha considerado una jornada laboral de 5 horas, ya que es lo que se ha pactado en el proyecto formativo.

Como se puede ver, en la fase de inicio, una vez aclarados los trámites institucionales de la UJI con la entidad, se dedican los primeros días a obtener información sobre el proyecto a realizar y a definir los objetivos que se quieren alcanzar. Durante esta fase también se redacta la propuesta técnica, el primer entregable durante la estancia. En total se estiman 25 horas de trabajo para esta fase.

En la fase de planificación se describen todos los diagramas necesarios que permitan realizar el desglose de tareas y la estimación tanto temporal como de costes del proyecto. Además se definen los planes de gestión de riesgos y calidad del proyecto para mantener unos estándares a lo largo del seguimiento del proyecto. Para esta fase, se estima un total de 25 h de trabajo.

En la fase de ejecución encontramos una serie de subfases que son:

- Definición de requisitos: se definen los diagramas de casos de uso, los requisitos de datos y requisitos tecnológicos.
- Análisis: se definen las clases del sistema mediante un diagrama de clases y el flujo del sistema con diagramas de actividades.

- Diseño: se realiza el diseño lógico de la base de datos y mockups de las interfaces gráficas.
- Desarrollo: se empieza a programar el software y la base de datos.

Se trata de la fase con más carga de trabajo, como ya se había mencionado en la figura [3.1], por lo que se ha estimado un total de 240 h de trabajo.

Por último, en el despliegue se implanta el sistema y se entrega el producto final indicando las instrucciones de uso pertinentes. Esta última fase ocupa un total de 10 horas de trabajo.

En el desglose de las tareas dentro de la documentación del TFG encontramos la redacción de informes quincenales con una estimación de 2 h, la redacción de la memoria con un trabajo estimado de 50 h y por último la redacción y realización de la presentación del TFG con un coste temporal de 21h.

Al tratarse de una estimación para un proyecto con el horario muy limitado, en algunas ocasiones no se ha seguido la planificación tal y como está indicado. Realizar una estimación fiel a lo que puede suceder en la realidad es una tarea complicada que únicamente se puede mejorar con la experiencia y el trabajo en otros proyectos.

Algunas de las variaciones respecto a la planificación inicial es que se han obviado algunas tareas de la fase de planificación que no hacían mucha falta como el diagrama de PERT o los diagramas de la fase de análisis. Se añadieron tareas que no se habían planificado inicialmente como la documentación de las clases y la redacción de un manual de usuario. Y finalmente se dejó el sistema totalmente preparado para la integración con el sistema de turnos y despliegue del sistema en el servidor de la entidad para que lo pueda hacer el supervisor una vez finalizado el proyecto.

En la figura 3.2 se puede ver la planificación inicial estimada y en la figura 3.3 el diagrama de Gantt más fiel a la realidad.

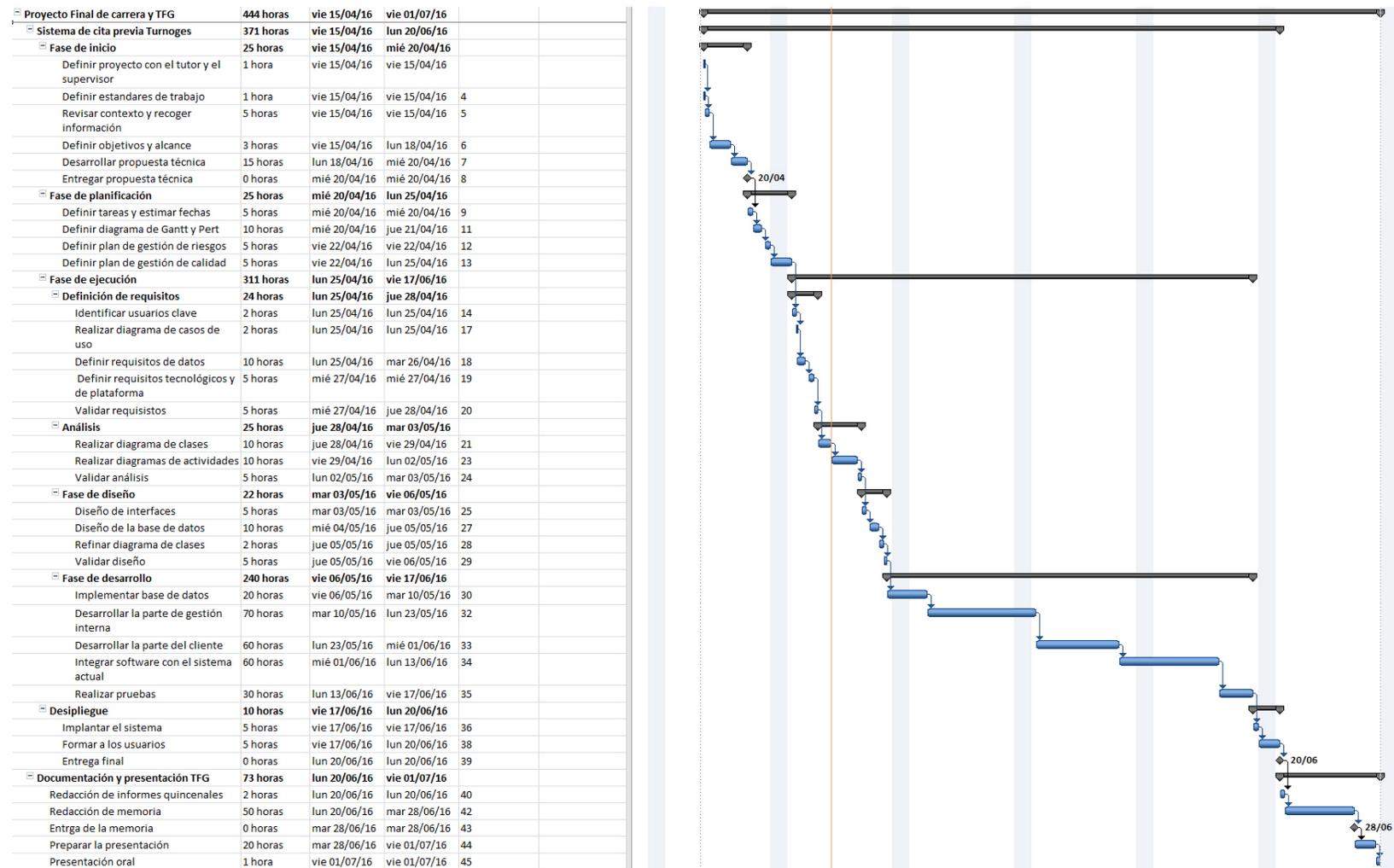


Figura 3.2: Diagrama de Gantt inicial

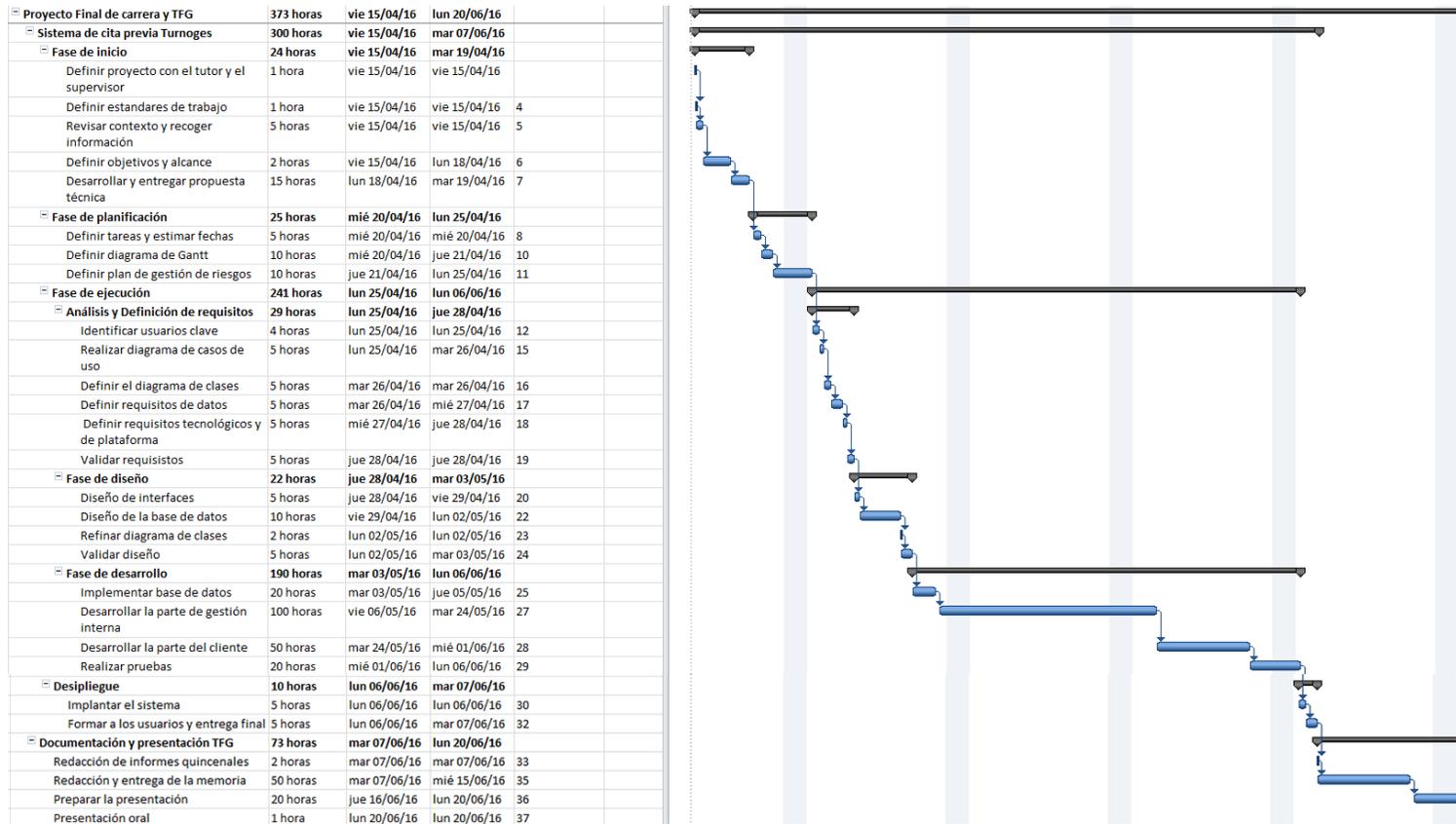


Figura 3.3: Diagrama de Gantt final

### 3.3. Estimación de recursos y costes del proyecto

Los recursos que han sido utilizados para desarrollar el proyecto los podemos dividir en recursos humanos, recursos de software y recursos de hardware. A continuación se detallan los 3 tipos de recursos:

#### 3.3.1. Recursos humanos

En este caso, el proyecto ha sido realizado por una sola persona, aunque se han realizado tareas tanto de desarrollador como de diseñador web. El coste de los recursos humanos que la empresa habría pagado por el desarrollo de este proyecto en caso de haber cobrado se muestra calculado a continuación:

Puesto: Programador Junior a 5€/h

Salario medio: 1200€/mes (jornada completa de 40h)

Salario jornada de 25h: 750€/mes

Duración proyecto: 300 horas = 3 meses

Coste:  $750 * 3 = 2250$ €

#### 3.3.2. Recursos software

En cuanto a software, solo se utilizará software libre por lo que no supone ningún coste adicional. Los recursos utilizados, que ya se habían descrito anteriormente, son los siguientes:

PHP: lenguaje de programación back-end.

MySQL: sistema de gestión de base de datos.

Apache: Servidor web

HTML5, CSS3, Javascript: herramientas desarrollo front-end

Bootstrap 3: framework front-end

JQuery, JQuery UI: frameworks front-end

Brackets: entorno de desarrollo.

### 3.3.3. Recursos hardware

Finalmente, por lo que se refiere a recursos hardware, la entidad ha proporcionado todo aquello que hacía falta. A continuación se listan dichos recursos y se estima el coste de los dispositivos:

#### **Ordenador de sobremesa: 300 €**

Sistema operativo: Windows 7 Professional

Procesador: Intel Core I5

Memoria Ram: 4.00 GB

#### **Periféricos 215 €**

Teclado

Ratón

Monitor

#### **Servidor de turnos (tableta) 250 €**

Sistema operativo: Windows 10

Procesador: Intel Core I5

Memoria Ram: 2.00 GB

## 3.4. Seguimiento del proyecto

### 3.4.1. Plan de gestión de riesgos

Uno de los puntos fundamentales para poder alcanzar el éxito en un proyecto, cumpliendo las restricciones temporales, funcionales y de calidad es realizar una buena gestión de los posibles riesgos que pueden surgir. El objetivo del plan de gestión de riesgos[3] es observar los distintos riesgos que pueden aparecer y estudiar su contexto para definir estrategias que prevengan estos riesgos o reduzcan su impacto. Los riesgos definidos se deben contemplar durante toda el ciclo del proyecto hasta que sea finalizado y se pueden modificar según se crea conveniente.

#### **Identificación de riesgos**

En la siguiente tabla se lista una serie de riesgos que se ha considerado que pueden surgir durante el desarrollo del proyecto. A cada riesgo se le ha asignado un identificador y una breve descripción así como si se trata de un riesgo del proyecto o del producto.

<b>ID</b>	<b>Descripción del riesgo</b>	<b>Tipo de riesgo</b>
R01	Requisitos poco claros	Riesgo del producto
R02	Falta de experiencia en las tareas de planificación	Riesgo del proyecto
R03	Falta de experiencia con las herramientas utilizadas	Riesgo del proyecto/producto
R04	Conflictos entre los implicados en el proyecto	Riesgo del proyecto
R05	Falta de seguimiento de las tareas	Riesgo del proyecto
R06	Cambio de tecnologías	Riesgo del proyecto/producto

Cuadro 3.1: Identificación de riesgos

### **Análisis de riesgos**

En el análisis de riesgos se contempla para cada riesgo definido, su magnitud, impacto, los posibles indicadores que nos hagan pensar que va a suceder y se realiza una descripción más detallada del riesgo.

ID	Análisis del riesgo
R01	<p data-bbox="292 651 437 685"><b>Magnitud</b></p> <p data-bbox="292 723 647 757">Variable según la aparición:</p> <ul data-bbox="336 786 635 992" style="list-style-type: none"> <li data-bbox="336 786 512 819">▪ Inicio: baja</li> <li data-bbox="336 842 612 875">▪ Elaboración: media</li> <li data-bbox="336 898 600 931">▪ Construcción: alta</li> <li data-bbox="336 954 635 987">▪ Despliegue: muy alta</li> </ul> <p data-bbox="292 1021 464 1055"><b>Descripción</b></p> <p data-bbox="292 1093 1327 1200">En la definición de requisitos se representan las ideas que tiene el cliente sobre la aplicación, sobre ellos se tendrá que desarrollar la aplicación por tanto una mala definición de requisitos puede afectar a la calidad del proyecto.</p> <p data-bbox="292 1238 416 1272"><b>Impacto</b></p> <p data-bbox="292 1310 1327 1485">Añadir o modificar requisitos durante el desarrollo provocará tener que realizar cambios en la documentación del producto. Si los cambios se producen en fases tempranas pueden no suponer un gran impacto sobre el proyecto, pero en fases finales pueden suponer tener que cambiar el código fuente y un gran impacto en tiempo disponible.</p> <p data-bbox="292 1523 464 1556"><b>Indicadores</b></p> <p data-bbox="292 1594 1327 1662">El cliente no sabe indicar con claridad cuáles son los servicios que quiere que ofrezca la aplicación.</p>

Cuadro 3.2: Análisis de R01: Requisitos poco claros

<b>ID</b>	<b>Análisis del riesgo</b>
R02	<p><b>Magnitud</b></p> <p>Media</p> <p><b>Descripción</b></p> <p>Se tiene poca experiencia en el desarrollo de software siguiendo una estructura de tareas y fechas preestablecido.</p> <p><b>Impacto</b></p> <p>En la planificación se estiman todas las tareas que se tienen que realizar para completar el proyecto con éxito. La falta de experiencia puede provocar una mala planificación que produce retrasos en el proyecto.</p> <p><b>Indicadores</b></p> <p>Diferencias entre el desarrollo real del proyecto y la planificación estimada.</p>

Cuadro 3.3: Análisis de R02: Falta de experiencia en las tareas de planificación

<b>ID</b>	<b>Análisis del riesgo</b>
R03	<p><b>Magnitud</b></p> <p>Variable según la aparición:</p> <ul style="list-style-type: none"> <li>▪ Inicio: baja</li> <li>▪ Elaboración: media</li> <li>▪ Construcción: alta</li> <li>▪ Despliegue: muy alta</li> </ul> <p><b>Descripción</b></p> <p>Se tiene poca experiencia en con las herramientas que se deben utilizar para el desarrollo del proyecto dificultando la consecución de objetivos.</p> <p><b>Impacto</b></p> <p>Puede suponer retrasos en el proyecto.</p> <p><b>Indicadores</b></p> <p>No pertinente</p>

Cuadro 3.4: Análisis de R03: Falta de experiencia con las herramientas utilizadas

<b>ID</b>	<b>Análisis del riesgo</b>
R04	<p><b>Magnitud</b></p> <p>Media</p> <p><b>Descripción</b></p> <p>Aparición de problemas y discrepancias entre los miembros del proyecto. Falta de acuerdo en las decisiones tomadas.</p> <p><b>Impacto</b></p> <p>Si los desacuerdos no son rápidamente resueltos se pueden provocar retrasos en la planificación. Teniendo en cuenta que no se puede producir un retraso en la entrega final, se tendría que reajustar la planificación con una posible pérdida de calidad del producto.</p> <p><b>Indicadores</b></p> <p>Mucho tiempo dedicado a decisiones concretas, énfasis en las posturas enfrentadas, número de enfrentamientos con respecto a una misma decisión.</p>

Cuadro 3.5: Análisis de R04: Conflictos entre los implicados en el proyecto

<b>ID</b>	<b>Análisis del riesgo</b>
R05	<p><b>Magnitud</b></p> <p>Media</p> <p><b>Descripción</b></p> <p>No se realiza un seguimiento de las tareas planificadas provocando que algunas de ellas sean dejadas para última instancia, con la consecuente baja en su calidad.</p> <p><b>Impacto</b></p> <p>Sobrecarga de trabajo en los días previos a la entrega de un presentable, pobre calidad de los entregables, se obvian detalles importantes.</p> <p><b>Indicadores</b></p> <p>Se observa gran cantidad de trabajo a medida que se llega a las fechas claves de entrega.</p>

Cuadro 3.6: Análisis de R05: Falta de seguimiento de las tareas

ID	Análisis del riesgo
R06	<p data-bbox="288 698 437 732"><b>Magnitud</b></p> <p data-bbox="288 772 647 806">Variable según la aparición:</p> <ul data-bbox="336 833 635 1048" style="list-style-type: none"> <li data-bbox="336 833 512 866">▪ Inicio: baja</li> <li data-bbox="336 891 612 925">▪ Elaboración: media</li> <li data-bbox="336 949 601 983">▪ Construcción: alta</li> <li data-bbox="336 1008 635 1041">▪ Despliegue: muy alta</li> </ul> <p data-bbox="288 1072 464 1106"><b>Descripción</b></p> <p data-bbox="288 1144 1327 1211">Se decide realizar un cambio en cualquiera de las tecnologías usadas durante el desarrollo del proyecto.</p> <p data-bbox="288 1252 416 1285"><b>Impacto</b></p> <p data-bbox="288 1323 1327 1435">En fases iniciales no supone un gran impacto, pero en fases avanzadas supone tener que cambiar el código fuente de la aplicación, provocando retrasos en el proyecto.</p> <p data-bbox="288 1467 464 1500"><b>Indicadores</b></p> <p data-bbox="288 1538 1327 1606">Se observa dificultad para alcanzar objetivos con las herramientas que se están utilizando o no se están viendo los resultados esperados.</p>

Cuadro 3.7: Análisis de R06: Cambio de tecnologías

## Acciones de Prevención y de Corrección

Por último, en la siguiente tabla se describe para cada riesgo definido las acciones que se pueden realizar para prevenir o corregir los riesgos si se han producido.

ID	Plan de prevención	Plan de corrección
R01	Realizar todas las consultas necesarias con el propietario del producto.	En las primeras fases se realizarán los cambios pertinentes para introducir los nuevos requisitos. En fases finales se deberá valorar la importancia de las modificaciones frente al tiempo disponible para realizar los cambios. Si se realizan se deberá modificar toda la documentación pertinente y las partes del código necesarias.
R02	Se deberá establecer un metodología de planificación y estudiar sus detalles si no se conoce para poder seguirla con éxito.	Si se utiliza una metodología con iteraciones se pueden observar los errores y tratar de corregirlos en etapas anteriores.
R03	Una parte del tiempo del proyecto se destinará al aprendizaje de las nuevas herramientas.	Si se produce un retraso en el aprendizaje por parte de un miembro del equipo, los demás miembros tratarán de ayudar a superarlo. Si no resultara, consultar a fuentes externas como profesores, bibliografía, foros en Internet. En último lugar se haría una redistribución de tareas.
R04	Cada vez que se tome una decisión, todo tiene que quedar totalmente claro y con la aceptación total de todos los miembros del proyecto.	Se establecerán reglas para determinar quien tomará determinadas decisiones.
R05	Llevar al día una revisión del estado del proyecto para anotar los posibles atrasos y poder así tomar medidas en el instante.	Realizar una revisión de todas las tareas retrasadas u olvidadas e procurar que se mantenga al día desde ese momento.
R06	Realizar un buen análisis de las tecnologías más apropiadas para la realización del proyecto antes de iniciar su desarrollo.	Realizar cambios pertinentes de la manera mas eficiente para adaptar la documentación o el código fuente para adaptarlo a las nuevas tecnologías.

Cuadro 3.8: Acciones de prevención y corrección

## Capítulo 4

# Análisis y diseño del sistema

### 4.1. Análisis y definición de requisitos

La definición de los requisitos es una parte fundamental en un proyecto software. Una mala definición de requisitos puede llevar un proyecto directamente al fracaso. Un requisito desde el punto de vista del usuario lo podemos definir como una “condición o capacidad que necesita un usuario para resolver un problema o conseguir un objetivo”. Desde el punto de vista del sistema se puede definir como la “condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.”[1]

En general podemos decir que en la definición de requisitos se describe el comportamiento del sistema que se va a desarrollar y su importancia radica en que una definición de requisitos pobre provoca que el sistema que se obtenga como resultado sea probablemente también pobre. Por tanto para definir los requisitos del sistema es importante tener claros los objetivos que se quieren conseguir y qué necesidades se pretenden cubrir.

#### 4.1.1. Diagrama de casos de uso (DCU)

El diagrama de casos de uso describe la interacción entre el sistema y los usuarios. En el DCU encontramos casos de uso que se definen como las acciones o procesos que el sistema debe poder realizar. También encontramos actores que representan los distintos tipos de roles del sistema y se relacionan con aquellos casos de uso que podrán realizar. Un usuario del sistema en la realidad puede adaptar distintos tipos de roles.

Por tanto, para realizar el DCU, en primer lugar debemos definir los distintos tipos de actores del sistema. El cuadro [4.1] describe los distintos actores que existen en el sistema de cita previa. En la figura [4.1] se muestra el diagrama de casos de uso para este proyecto. o

Actor	Descripción
Superadmin	Encargado de gestionar la aplicación al completo, tiene acceso a todo el contenido y acciones. Trabajadores de SSCC.
Admin	Gestiona parte del sistema, puede realizar todo tipo de acciones excepto el mantenimiento de usuarios y centros.
Cliente	Beneficiario del servicio.

Cuadro 4.1: Identificación de actores del sistema

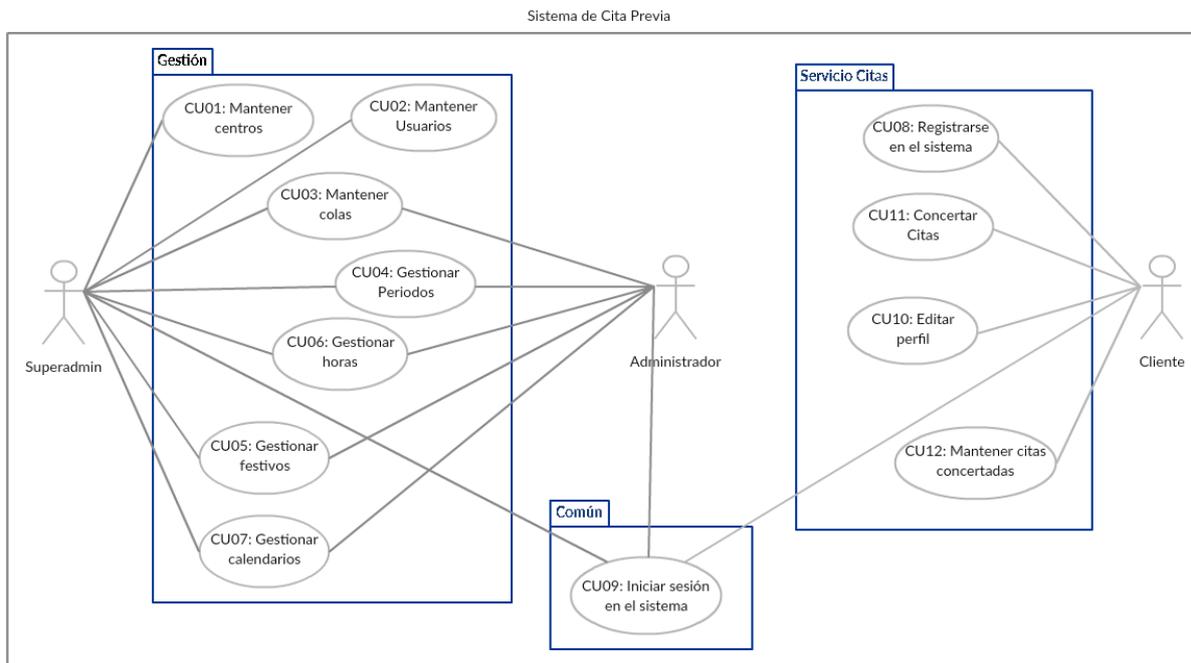


Figura 4.1: Diagrama de casos de uso

A continuación se describen los distintos casos de uso que aparecen en el diagrama:

<b>Identificador</b>	CU01
<b>Caso de uso</b>	Mantener centros
<b>Actores</b>	Superadmin
<b>Casos relacionados</b>	Ninguno
<b>Descripción</b>	El sistema debe permitir al usuario añadir, editar y eliminar centros que ofrecen el servicio de citas.
<b>Precondición</b>	El usuario debe estar iniciado como super administrador.
<b>Prioridad</b>	Alta

Cuadro 4.2: Descripción del caso de uso CU01

<b>Identificador</b>	CU02
<b>Caso de uso</b>	Mantener usuarios
<b>Actores</b>	Superadmin
<b>Casos relacionados</b>	Ninguno
<b>Descripción</b>	El sistema debe permitir al usuario añadir, editar o eliminar usuarios que se encuentran registrados en el sistema.
<b>Precondición</b>	El usuario debe estar iniciado como super administrador.
<b>Prioridad</b>	Media

Cuadro 4.3: Descripción del caso de uso CU02

<b>Identificador</b>	CU03
<b>Caso de uso</b>	Mantener colas
<b>Actores</b>	Superadmin, admin
<b>Casos relacionados</b>	Ninguno
<b>Descripción</b>	El sistema debe permitir al usuario añadir, editar o eliminar colas para los distintos centros disponibles.
<b>Precondición</b>	El usuario debe estar iniciado como super administrador o administrador
<b>Prioridad</b>	Alta

Cuadro 4.4: Descripción del caso de uso CU03

<b>Identificador</b>	CU04
<b>Caso de uso</b>	Gestionar periodos
<b>Actores</b>	Superadmin, admin
<b>Casos relacionados</b>	Ninguno
<b>Descripción</b>	El sistema debe permitir al usuario añadir, editar o eliminar perdidos en un horario determinado.
<b>Precondición</b>	El usuario debe estar iniciado como super administrador o administrador.
<b>Prioridad</b>	Alta

Cuadro 4.5: Descripción del caso de uso CU04

<b>Identificador</b>	CU05
<b>Caso de uso</b>	Gestionar festivos
<b>Actores</b>	Superadmin, admin
<b>Casos relacionados</b>	Ninguno
<b>Descripción</b>	El sistema debe permitir al usuario añadir, editar o eliminar los festivos del año.
<b>Precondición</b>	El usuario debe estar iniciado como super administrador o administrador
<b>Prioridad</b>	Media

Cuadro 4.6: Descripción del caso de uso CU05

<b>Identificador</b>	CU06
<b>Caso de uso</b>	Gestionar horas
<b>Actores</b>	Superadmin, admin
<b>Casos relacionados</b>	Ninguno
<b>Descripción</b>	El sistema debe permitir al usuario añadir, editar o eliminar el horario de trabajo diario para cada periodo.
<b>Precondición</b>	El usuario debe estar iniciado como super administrador o administrador y debe existir algún periodo en el sistema.
<b>Prioridad</b>	Alta

Cuadro 4.7: Descripción del caso de uso CU06

<b>Identificador</b>	CU07
<b>Caso de uso</b>	Gestionar calendarios
<b>Actores</b>	Superadmin, admin
<b>Casos relacionados</b>	Ninguno
<b>Descripción</b>	El sistema debe permitir añadir citas al calendario de un centro para un periodo y una cola determinados.
<b>Precondición</b>	El usuario debe estar iniciado como super administrador o administrador. Deben haber configurados algún horario, cola, periodo y hora.
<b>Prioridad</b>	Alta

Cuadro 4.8: Descripción del caso de uso CU07

<b>Identificador</b>	CU08
<b>Caso de uso</b>	Registrarse en el sistema
<b>Actores</b>	Cliente
<b>Casos relacionados</b>	Ninguno
<b>Descripción</b>	El sistema debe permitir a los clientes registrarse en el sistema introduciendo los datos válidos que deseen.
<b>Precondición</b>	El cliente no debe estar ya dado de alta.
<b>Prioridad</b>	Alta

Cuadro 4.9: Descripción del caso de uso CU08

<b>Identificador</b>	CU09
<b>Caso de uso</b>	Iniciar sesión en el sistema
<b>Actores</b>	Superadmin, admin, cliente
<b>Casos relacionados</b>	Ninguno
<b>Descripción</b>	El sistema debe permitir a los usuarios iniciar sesión en el sistema con sus datos de registro.
<b>Precondición</b>	El usuario debe estar dado de alta en el sistema.
<b>Prioridad</b>	Alta

Cuadro 4.10: Descripción del caso de uso CU09

<b>Identificador</b>	CU10
<b>Caso de uso</b>	Editar perfil
<b>Actores</b>	Cliente
<b>Casos relacionados</b>	Ninguno
<b>Descripción</b>	El sistema debe permitir al usuario modificar sus datos personales.
<b>Precondición</b>	El usuario tiene que estar iniciado en el sistema como cliente.
<b>Prioridad</b>	Alta

Cuadro 4.11: Descripción del caso de uso CU10

<b>Identificador</b>	CU11
<b>Caso de uso</b>	Concertar citas
<b>Actores</b>	Cliente
<b>Casos relacionados</b>	Ninguno
<b>Descripción</b>	El sistema debe permitir al cliente buscar y concertar citas previas.
<b>Precondición</b>	El usuario tiene que estar iniciado en el sistema como cliente.
<b>Prioridad</b>	Alta

Cuadro 4.12: Descripción del caso de uso CU11

<b>Identificador</b>	CU12
<b>Caso de uso</b>	Mantener citas concertadas
<b>Actores</b>	Cliente
<b>Casos relacionados</b>	Ninguno
<b>Descripción</b>	El sistema debe permitir al cliente consultar sus citas previas tanto pendientes como pasadas y cancelar o cambiar citas previas pendientes.
<b>Precondición</b>	El usuario tiene que estar iniciado en el sistema como cliente.
<b>Prioridad</b>	Alta

Cuadro 4.13: Descripción del caso de uso CU12

#### 4.1.2. Requisitos de datos

Los requisitos de datos determinan las entidades y atributos que es necesario almacenar en el sistema para poder satisfacer los casos de uso que se han descrito en el apartado anterior.

En las siguientes tablas se describen los requisitos de datos para este proyecto.

<b>Código</b>	RD01
<b>Nombre</b>	Usuario
<b>Información</b>	Para cada usuario el sistema debe guardar su nombre de usuario, contraseña cifrada, correo electrónico, nombre, apellidos, teléfono y permisos.
<b>Comentarios</b>	Permisos indica el tipo de usuario del que se trata (administrador, súper administrador o usuario).

Cuadro 4.14: Descripción del requisito de datos RD01

<b>Código</b>	RD02
<b>Nombre</b>	Centro
<b>Información</b>	Para cada centro se tiene que almacenar su código y su nombre.
<b>Comentarios</b>	Ninguno

Cuadro 4.15: Descripción del requisito de datos RD02

<b>Código</b>	RD03
<b>Nombre</b>	Horario
<b>Información</b>	Para cada horario se tiene que almacenar su hora de inicio, su hora de fin y su nombre.
<b>Comentarios</b>	Ninguno

Cuadro 4.16: Descripción del requisito de datos RD03

<b>Código</b>	RD04
<b>Nombre</b>	Cola
<b>Información</b>	Para cada cola se tiene que almacenar el centro asociado, el código de cola, el nombre, un horario, el tiempo de espera y la la prioridad.
<b>Comentarios</b>	Ninguno

Cuadro 4.17: Descripción del requisito de datos RD04

<b>Código</b>	RD05
<b>Nombre</b>	Festivo
<b>Información</b>	Para cada festivo se tiene que almacenar el nombre y la fecha de la festividad.
<b>Comentarios</b>	Ninguno

Cuadro 4.18: Descripción del requisito de datos RD05

<b>Código</b>	RD06
<b>Nombre</b>	Periodo
<b>Información</b>	Para cada periodo se tiene que almacenar el día de inicio, el día de fin, el horario y el espacio temporal entre citas.
<b>Comentarios</b>	El espacio temporal indica el espacio de tiempo en minutos que habrá entre cada cita para las citas creadas en ese periodo.

Cuadro 4.19: Descripción del requisito de datos RD06

<b>Código</b>	RD07
<b>Nombre</b>	Hora
<b>Información</b>	Para cada hora se tiene que almacenar el periodo asociado, la hora de inicio, la hora de fin y el día de la semana.
<b>Comentarios</b>	Ninguno

Cuadro 4.20: Descripción del requisito de datos RD07

<b>Código</b>	RD08
<b>Nombre</b>	Calendario o cita
<b>Información</b>	Para cada calendario o cita se tiene que almacenar el centro, la cola, el usuario, el día, la hora, el día del aviso, la hora del aviso, la disponibilidad y el turno.
<b>Comentarios</b>	Esta entidad representa cada cita previa.

Cuadro 4.21: Descripción del requisito de datos RD08

## 4.2. Diseño de la arquitectura del sistema

En la fase de diseño del sistema se recogen los requisitos definidos en la fase de análisis y se convierten en elementos más cercanos a la fase de desarrollo. En el siguiente apartado se muestra el modelo conceptual de datos donde se plasma los datos que debe almacenar el sistema y la relación entre ellos.

### 4.2.1. Modelo conceptual de datos

Para el diseño de la base de datos, la entidad planteó una serie de posibles tablas iniciales sobre las cuales se han realizado algunas modificaciones. La notación de las tablas se ha hecho siguiendo el estándar `turnos_cp_nombreTabla` que también indicó la entidad. A continuación se muestran dichas tablas y su descripción:

Nombre	Descripción
<code>turnos_cp_calendarios</code>	Tabla que representa el conjunto de citas previas disponibles o no de cada centro
<code>turnos_cp_centros</code>	Tabla que representa los centros que ofrecen el servicio de cita previa a sus clientes
<code>turnos_cp_colas</code>	Tabla que representa los servicios que ofrece cada centro a sus clientes
<code>turnos_cp_festivos</code>	Tabla que mantiene las festividades anuales
<code>turnos_cp_horario</code>	Tabla que mantiene los horarios de trabajo que son valores constantes (mañana y tarde)
<code>turnos_cp_horas</code>	Tabla que mantiene las horas de trabajo para cada día de la semana dentro de un periodo de tiempo
<code>turnos_cp_periodo</code>	Tabla que mantiene el horario de trabajo durante un periodo de tiempo
<code>turnos_cp_usuarios</code>	Tabla que representa a los distintos tipos de usuarios del sistema

Cuadro 4.22: Descripción de las tablas de la base de datos del proyecto

El planteamiento ofrecido por la entidad incluía también una tabla “`turnos_cp_contador`” con el objetivo de mantener actualizado el último turno de cada cola. Finalmente se decidió no utilizar esta tabla porque suponía un problema que se detallará en capítulos posteriores.

A la hora de diseñar la base de datos se ha tenido muy en cuenta que el sistema ha de ser integrable con TurnoGes. Por eso se ha diseñado una vista con todos aquellos datos que requiere el registro de los turnos en el sistema TurnoGes. A esta vista se le ha llamado “`turnos_cp_citas`” y será la que proporcione los datos de las citas previas a TurnoGes.

A continuación se muestra el diagrama conceptual de la base de datos.

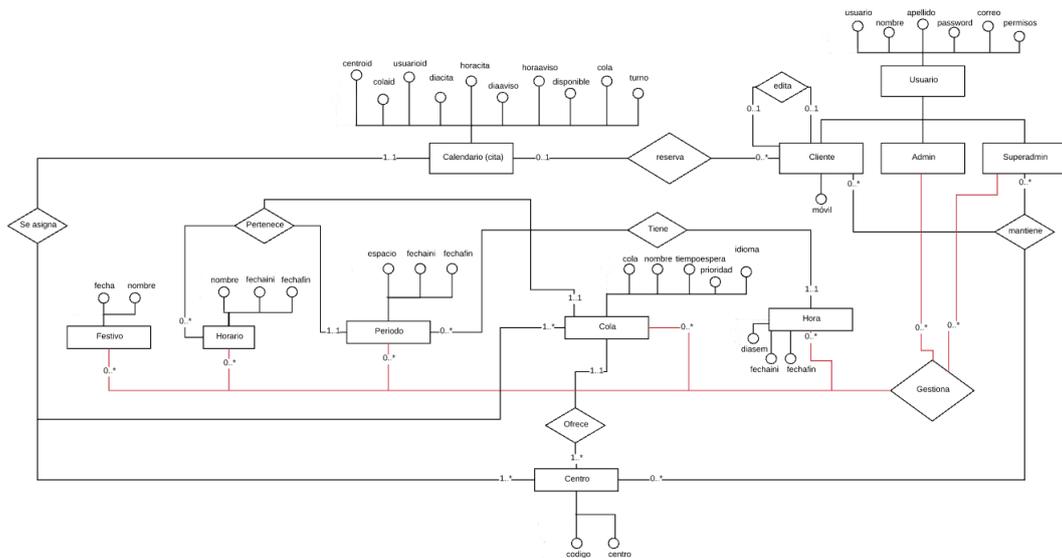


Figura 4.2: Modelo conceptual de la base de datos.

#### 4.2.2. Diagrama lógico de la base de datos

En la siguiente figura se muestra el modelo lógico de la base de datos. Se ha obtenido directamente de la herramienta PhpMyAdmin a través de su función de vista de diseñador.

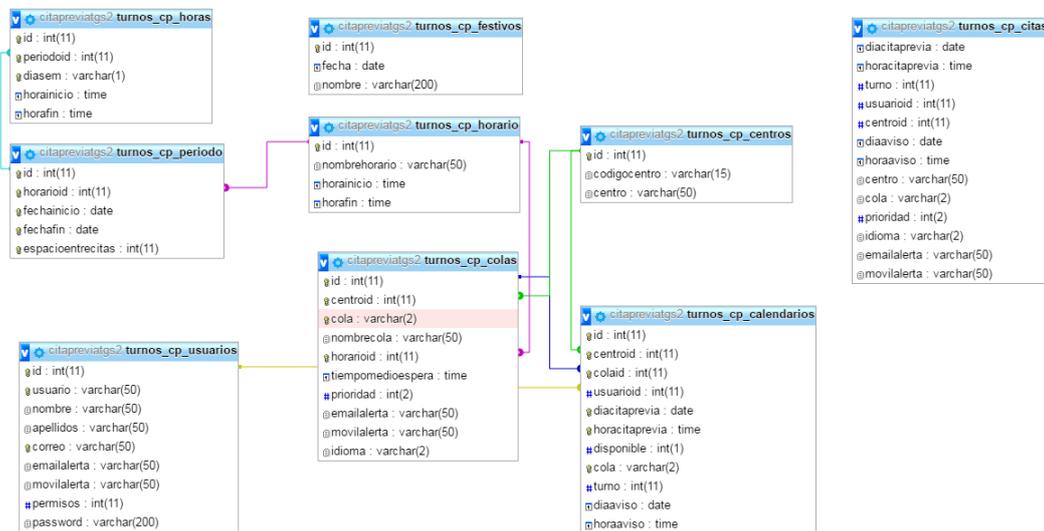


Figura 4.3: Modelo lógico de la base de datos.

En el diseño lógico, figura [4.3.], se muestran las diferentes tablas de la base de datos y como se relacionan entre sí mediante claves ajenas. Además de las tablas, también se muestra la vista **turnos\_cp\_citas**, como se ha dicho, esta vista recoge los datos fundamentales de cada cita concertada y su funcionalidad es mantener dichas citas para importarlas diariamente a la base de datos del sistema de turnos. De esta manera, el sistema TurnoGes, puede gestionar las citas previas de ese día en el centro.

Como se puede ver también en el diseño lógico se ha decidido optar por una única tabla de usuarios (`turnos_cp_usuarios`) en la que se diferenciarán los distintos tipos de roles mediante el campo permisos. En el caso de que el usuario sea un cliente se le asigna el permiso 0, si se trata de un administrador el 9 y si se trata de un superadmin, se le asigna el 10. Además los campos “emailalerta” y “movilalerta” aceptan nulos ya que únicamente serán necesitados para el rol de cliente.

Si observamos la tabla de periodos (`turnos_cp_periodos`), se podrá ver que consta de una fecha de inicio y una de fin para representar el periodo temporal, otro campo para representar el espacio temporal entre las citas de ese periodo y una referencia a la tabla horario. De esta manera cada periodo tiene asignado un horario.

En la tabla de horas (`turnos_cp_horas`), se observan los campos hora de inicio y hora de fin, el día de la semana y una referencia a la tabla periodos. De esta manera, para cada periodo se tienen bien definidas los días y las horas del servicio.

En la tabla colas (`turnos_cp_colas`), se puede ver que entre otros campos, cuenta con una referencia al centro, esto sirve para indicar a que centro pertenece dicha cola. También cuenta una referencia a la tabla horario para indicar el horario en el que se ofrecerá dicho servicio.

También cabe destacar que la tabla calendarios (`turnos_cp_calendarios`) mantiene el conjunto de citas previas disponibles o no disponibles para cada centro. Por tanto cuenta con una referencia al centro, una referencia a la cola o servicio que se ofrece y también una referencia al usuario que puede ser nula si la cita no ha sido concertada.

El resto de tablas son triviales por lo que no hay nada destacable que mencionar.

### 4.2.3. Diseño de la interfaz

En cualquier aplicación siempre es importante tener una interfaz de usuario bonita, fácil de utilizar y amigable para el usuario ya que la interfaz es la primera impresión que se llevará el usuario sobre la aplicación. Una interfaz fea o difícil de utilizar puede provocar que el usuario no quede satisfecho con el servicio por mucho que funcione de manera infalible internamente.

Además de esto, hoy en día es un factor importante que la interfaz se adapte a cualquier tipo de dispositivo, ya que una gran parte de los usuarios navegan desde dispositivos móviles o tabletas, no sólo desde ordenadores, por tanto se ha de conseguir que también ellos disfruten de una experiencia agradable.

Por todo esto se decidió utilizar el framework Bootstrap que proporciona una serie de herramientas que permiten obtener un diseño bonito y “responsive” de una manera sencilla.

Para el desarrollo de las interfaces se ha intentado tener en cuenta en todo momento alguna reglas como:

- **Mantener la coherencia:** es importante que todas las interfaces tengan consistencia y

no varíen mucho unas de otras para conseguir que el usuario se acomode con más facilidad al uso de la aplicación.

- **Utilizar accesos directos:** cuando aumenta la frecuencia de uso, el usuario desea un mayor ritmo de interacción por lo que disponer de accesos directos es algo muy importante especialmente para usuarios avanzados.
- **Retroalimentación informativa:** se debe intentar que para cada acción haya una respuesta especialmente si se trata de una acción trascendente.
- **Manipulación de errores:** en primer lugar se debería evitar que el usuario pueda cometer errores intentando detectarlos con el sistema, pero si los comete, se debe permitir al usuario corregir sus errores de una manera directa y sencilla.
- **Reducción de la carga de memoria:** se ha intentado reducir al máximo el hecho de que el usuario tenga que recordar ciertos datos mientras navega por la página ya que esto supone una limitación para el usuario y por tanto una peor experiencia. Por tanto siempre que se pueda, se intenta ofrecer al usuario la máxima información posible.

A continuación se muestran el resultado de las interfaces del sistema para comprender mejor todos los aspectos que se han tratado en este apartado:



The image shows a login screen for a service. At the top, there is a green square icon with a white leaf-like symbol. Below the icon, the text "Servicio Cita Previa" is displayed in a bold, black font. The main content area is a light gray rounded rectangle with the title "Iniciar sesión" in a bold, gray font. Below the title, there are two input fields: "Nombre de usuario" and "Contraseña". Each field has a small icon to its right: a person icon for the username field and a lock icon for the password field. Below the input fields is a green button with the text "Iniciar" in white. At the bottom of the gray box, there are three lines of text in a smaller font: "¿No tienes cuenta de acceso? [Regístrate](#)", "¿Has olvidado tu usuario? [Recuperar](#)", and "¿Has olvidado tu contraseña? [Recuperar](#)".

Figura 4.4: Diseño de la pantalla de registro y login.



## Concertar cita

### 1.- Seleccione el centro donde acudirá:

- CAIXALMASSORA-Oficina Principal
- Oficina Central 1
- Oficina Central 2
- Oficina Central 3

### 2.- Seleccione el servicio que quiere solicitar:

- CP: Cita Previa (Idioma: ES) (Horario de 08:30:00 a 14:45:00)

### 3.- Seleccione el día y la hora de la cita:

Fecha de la cita



Hora de la cita

Buscar citas

## Citas disponibles

Completa el formulario para ver si hay citas disponibles.

Figura 4.5: Diseño de la pantalla de inicio de clientes.

**1.- Seleccione el centro donde acudirá:**

- CAIXALMASSORA-Oficina Principal
- Oficina Central 1
- Oficina Central 2
- Oficina Central 3

**2.- Seleccione el servicio que quiere solicitar:**

- CP: Cita Previa (Idioma: ES) (Horario de 08:30:00 a 14:45:00)

**3.- Seleccione el día y la hora de la cita:**

Fecha de la cita   Hora de la cita

## Citas disponibles

**Cita disponible encontrada:**

El día 04/08/2016 a las 08:30

**Otras citas disponibles ese mismo día:**

El día 04/08/2016 a las : 09:00	El día 04/08/2016 a las : 09:30	El día 04/08/2016 a las : 10:00
<input type="button" value="Aceptar cita"/>	<input type="button" value="Aceptar cita"/>	<input type="button" value="Aceptar cita"/>

Figura 4.6: Diseño de la pantalla de inicio de clientes con una búsqueda realizada.

The image shows a web interface for booking appointments. A modal dialog titled "Confirmar cita" is open, displaying the following content:

- Header: "Confirmar cita" with a close button (X).
- Message: "Has seleccionado la cita para el día 04/08/2016 a las 08:30."
- Question: "¿quieres recibir una notificación antes de la cita?"
- Radio buttons: "Si" (selected) and "No".
- Form fields: "Fecha de la aviso" (Formato dd/mm/AAA. with a calendar icon) and "Hora del aviso" (Formato hh:mm with a clock icon).
- Buttons: "Cancelar" and "Confirmar".

In the background, the main interface is visible, showing three steps:

- 1.- Seleccione el centro de**
  - CAIXALMASSORA-Oficina B
  - Oficina Central 1
  - Oficina Central 2
  - Oficina Central 3
- 2.- Seleccione el servicio de**
  - CP: Cita Previa (Idioma: ES)
- 3.- Seleccione el día y la hora**
  - Fecha de la cita: 04/08/2016

Below the modal, there is a "Buscar citas" button. Under the heading "Citas disponibles", a "Cita disponible encontrada:" is shown with the text "El día 04/08/2016 a las 08:30" and an "Aceptar cita" button. Below that, it says "Otras citas disponibles ese mismo día:".

Figura 4.7: Diseño de la pantalla de confirmación de reservas.



## Mis citas pendientes

#	Día	Hora	Centro	
0	29/07/2016	08:30:00	CAIXALMASSORA-Oficina Principal	  

Ocultar historial de citas

## Historial de citas

#	Día	Hora	Centro	Cola	Turno
0	30/06/2016	09:00:00	CAIXALMASSORA-Oficina Principal	Cola Rápida	1
1	13/07/2016	12:00:00	CAIXALMASSORA-Oficina Principal	Cola Rápida	8
2	14/07/2016	11:00:00	CAIXALMASSORA-Oficina Principal	Cola Rápida	6
3	19/07/2016	11:30:00	CAIXALMASSORA-Oficina Principal	Cita Previa	7

Mi Perfil



Próximas citas



- 29/07/2016

Total citas

#  
5

Figura 4.8: Diseño de la pantalla de gestión de citas del cliente.

Petición cita previa

Mis citas pendientes

#	Día	Hora
0	29/07/2016	08:30:00

Ocultar historial de citas

Historial de citas

#	Día	Hora	Centro	Cola	Turno
0	30/06/2016	09:00:00	CAIXALMASSORA-Oficina Principal	Cola Rápida	1
1	13/07/2016	12:00:00	CAIXALMASSORA-Oficina Principal	Cola Rápida	8
2	14/07/2016	11:00:00	CAIXALMASSORA-Oficina Principal	Cola Rápida	6
3	19/07/2016	11:30:00	CAIXALMASSORA-Oficina Principal	Cita Previa	7

Más información

Cita el día 29/07/2016 a las 08:30

Centro: CAIXALMASSORA-Oficina Principal

Aviso configurado: No

Servicio seleccionado: CP - Cita Previa

Turno: 1

Cerrar

Mi Perfil

Próximas citas

- 29/07/2016

Total citas

#  
5

Figura 4.9: Pantalla de información de la cita seleccionada.



## Perfil de Carlos Garcia Garcia

### Editar datos personales

Nombre de usuario:

Carlito 

Nombre:

Carlos 

Apellidos:

Garcia Garcia 

Email:

carluo@gmail.com 

Teléfono móvil:

655321478 

Editar

### Cambiar contraseña

Nueva contraseña:



Repetir contraseña:

Figura 4.10: Pantalla de edición de perfil.

# Turnoges

## Gestión Servicio Cita Previa



### Administración

#### Gestión de Datos

- Gestión de centros
- Gestión de usuarios
- Gestión de horarios
- Gestión de colas

#### Gestión de calendarios y restricciones de horario

- Gestión de periodos
- Gestión de horas
- Gestión de festivos
- Gestión de calendarios

Figura 4.11: Diseño de la pantalla de inicio de administradores.

# Gestión de colas

Inicio

Sesión superadmin. Salir ↗

Mostrar  registros

Buscar:

ID ▲	Centro	Cola	Nombre cola	Horario	Tiempo medio espera	Prioridad	Email alerta	Móvil alerta	Idioma	Opciones
7	Oficina Central 1	CR	Cola Rápida	Mañana	00:03:00	1			ES	
11	Oficina Central 3	CR	Cita Previa	Tarde	00:01:00	1			ES	
17	CAIXALMASSORA	CP	Cita Previa	Mañana	00:05:00	1			ES	
18	Oficina Central 2	CP	Cita Previa	Mañana	00:10:00	1			EN	

Mostrando registros del 1 al 4 de un total de 4 registros

Anterior  Siguiente

Añadir Cola

Figura 4.12: Pantalla de gestión de colas en administración.



En la figura [4.5] y [4.6] se puede observa la pantalla de búsqueda de citas antes y después de realizar una búsqueda. El campo "Fecha de la cita" al se seleccionado muestra un calendario visual con aquellos días en los que existen citas remarcados y seleccionables.

En la figura [4.13] se puede observar el calendario de la parte de gestión con todas aquellas citas configuradas para que sean reservadas por los clientes.

## Capítulo 5

# Implementación y pruebas

### 5.1. Detalles de implementación

Como se ha mencionado en capítulos anteriores, no se ha utilizado ningún framework para el desarrollo de la aplicación, sino que se ha desarrollado todo el código desde cero. Personalmente, creo que es conveniente utilizar frameworks ya que simplifican la implementación y ahorran bastante trabajo, pero, por supuesto, es perfectamente posible trabajar con php sin ningún framework. Por otro lado, al ser la primera vez que uso php, no me parece mal aprenderlo sin ningún tipo de abstracción ya que a partir de ahí siempre será más sencillo utilizar un framework en el futuro.

A continuación se detalla el proceso que se siguió para desarrollar la aplicación.

#### 5.1.1. Instalación del servidor de turnos

Al iniciar el desarrollo, una de las primeras tareas que se realizó fue la instalación del servidor de turnos TurnoGes©. En primer lugar se utilizó un ordenador DELL con las mismas características que el ordenador usado para la programación, pero dicho ordenador se tuvo que retirar porque era requerido por otra entidad, por tanto se acabó realizando la instalación en una tableta con el sistema operativo Windows 10. Para realizar la configuración del servidor, se siguió el manual de instrucciones de TurnoGes©. Consistió en ejecutar la instalación de un paquete de aplicaciones necesarias para que funcione el sistema como son:

AppServ: una herramienta con apache, php y mysql muy similar a XAMPP, (la herramienta utilizada en el proyecto) de tal forma que se pueda construir y acceder a la base de datos de turnos. Aplicaciones de simulación del dispensador de turnos, sistema multimedia, display de puesto y terminal virtual para recrear el funcionamiento real del sistema. Otras aplicaciones requeridas como flash player, Framework .NET 4.0, etc.

Una vez instalado el servidor estudié su funcionamiento y especialmente como estaba construida la base de datos. Esto permitió entender mejor como había que realizar el desarrollo

del sistema de cita previa para que se pudiera integrar con el sistema de turnos. La base de datos del sistema cuenta con una tabla “**turnos\_registro**” donde se almacenan los datos de los turnos que piden los clientes al llegar a la entidad a través del dispensador. La estrategia que se decidió realizar para la integración fue construir una vista en la base de datos de cita previa que recogiera todos los datos que son necesarios en “**turnos\_registro**”, de esta manera se puede realizar una importación diaria de la base de datos al empezar el día y incorporar las citas previas al registro de turnos cuando llegue la hora de la cita y así ser gestionada por TurnoGes©. El sistema no permite al reserva de citas en el mismo día por lo que no supone ningún inconveniente a la hora de realizar la importación.

### 5.1.2. Desarrollo de la aplicación

Una vez comprendida como se realizaría la integración se empezó el desarrollo de la aplicación. En primer lugar se implementó la base de datos teniendo en cuenta los modelos obtenidos en la fase de análisis y diseño y también el requisito de la vista para realizar la importación de las citas. Una vez obtenida la implementación de la base de datos con todas sus relaciones bien establecidas, se empezó la programación.

#### Acceso a la base de datos

Para el acceso a la base de datos se decidió utilizar PDO (PHP Data Objects), se trata de una extensión que define una capa de abstracción o una interfaz ligera para poder acceder a bases de datos en PHP. Al ser una capa de abstracción permite que independientemente de la base de datos que se esté utilizando, se emplean las mismas funciones para realizar consultas y obtener datos. Esta extensión se encuentra disponible a partir de la versión 5.1 de PHP.

A diferencia de otras extensiones para acceso a base de datos como MySQL o MySQLi, PDO no tiene una sintaxis específica de ningún gestor de base de datos, por lo que se puede realizar cambios de base de datos y plataforma de manera realmente fácil. PDO soporta hasta 12 tipos de drivers de base de datos diferentes.

En la siguiente figura [5.1] se muestra el esquema de abstracción de PDO.



Figura 5.1: Capa de abstracción PDO para PHP.

Además de esto PDO posee otras muchas ventajas como un sistema de conexión facilísimo, un alto rendimiento o un sistema de seguridad ante inyecciones SQL.

La implementación de la conexión a la base de datos junto a las credenciales de la base de

datos se realizó en un fichero aparte para que en caso de cambiar la base de datos o cualquier otra situación, únicamente modificando este fichero se pueda solucionar.

A continuación se muestra la parte del código que mantiene las credenciales de la base de datos y realiza la conexión:

```
4 //Este if es para que no redefina la función
5 if (!function_exists('getDBConn')) {
6     function getDBConn(){
7         $host = 'localhost';
8         $dbname = 'citapreviatgs';
9         $username = 'root';
10        $password = 'root';
11        $params = array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8");
12        $db = null;
13        try {
14            $db = new PDO("mysql:host=$host;dbname=$dbname", $username, $password, $params);
15        } catch (PDOException $e) {
16            header('location: /serviciocita/index.php?code=-1');
17        }
18        return $db;
19    }
20 }
21 }
```

Figura 5.2: Código de conexión con la base de datos.

El código contiene un método “getDBConn” que devuelve la conexión a la base de datos.

## Estructura de la aplicación

Al no utilizar ningún tipo de framework, se ha tratado de realizar una estructura que siga el patrón de diseño MVC (modelo, vista, controlador) y así conseguir la mayor independencia entre los distintos componentes de la aplicación.

En el modelo se ha implementado una clase para cada objeto de dominio. Se ha seguido el patrón de diseño DAO (database access object) para obtener los distintos tipos de objetos. El patrón DAO nos ofrece una interfaz con las operaciones básicas de cada objeto sobre la base de datos. Cada DAO utiliza la conexión a la base de datos vista en el anterior apartado para obtener el objeto.

En la figura [5.3] se muestra el código de uno de estos objetos DAO.

Como se puede ver en el código se obtiene la conexión a la base de datos y se implementan los métodos oportunos para poder realizar operaciones sobre la tabla del objeto en la base de datos. En este caso se ve el método “getCals” que obtiene todas las entradas de la tabla turnos\_cp\_calendarios y devuelve los objetos en un vector. A parte de este método también hay otros como updateCal para actualizaciones, deleteCal para el borrado o addCal para la adición, etc.

La parte de gestión y la del cliente se ha dividido en carpetas diferentes. Para la gestión se ha creado una vista con su controlador para cada elemento que se ha de gestionar. Por ejemplo gesCalendarios.php será la vista de la gestión de calendarios y calendariosController.php será el controlador de dicha vista. En el controlador se define toda la lógica que se puede realizar para la vista correspondiente a través de funciones, estas funciones utilizan el modelo para mantener el estado de la aplicación actualizado y que pueda ser sea visualizado por el usuario. En la vista

```

1 k?php
2 /**
3  * Clase que interactua directamente con la base de datos.
4  */
5  include $_SERVER['DOCUMENT_ROOT']."/serviciocita/db/db.php";
6
7  class CalendariosModel{
8
9      private $db;
10
11     /**
12      * Constructor: Crea la conexión a la base de datos y lanza una excepción si se produce algún error.
13      */
14     public function __construct(){
15         try {
16             $this->db = getDBConn();
17         } catch (PDOException $e) {
18             print "Error al conectar con la BD: " . $e->getMessage();
19         }
20     }
21
22     public function closeDB(){
23         $this->db = null;
24     }
25
26     /**
27      * getCals
28      * obtiene un vector con los datos de todo tipo de citas.
29      */
30     public function getCals()
31     {
32         try {
33             $query = $this->db->prepare('SELECT * from turnos_cp_calendarios ORDER BY diacitaprevia, horacitaprevia');
34             $query->execute();
35             return $query->fetchAll();
36         } catch (PDOException $e) {
37             $e->getMessage();
38             $this->db = null;
39         }
40     }
41
42     /**
43      * Obtiene los días en los que hay citas con turnos sin asignar.
44      */
45     public function getDiasCalSinAsignar(){
46
47         try {
48             $query = $this->db->prepare('SELECT diacitaprevia FROM `turnos_cp_calendarios` WHERE turno Like -1 GROUP BY diacitaprevia');
49             $query->execute();
50             return $query->fetchAll();
51         } catch (PDOException $e) {
52             $e->getMessage();
53             $this->db = null;
54         }
55     }
56

```

Figura 5.3: Parte del código del DAO de calendarios.

se han implementado las interfaces con las que interactúa el usuario, cuando el usuario realiza una acción que requiere una actualización del modelo, se le hace una petición al controlador. El controlador recibe dicha petición y decide que acción debe realizar.

En la parte del cliente, como es más reducida, se ha utilizado un único controlador para atender las peticiones de la vista.

La estructura utilizada se puede representar gráficamente con el siguiente diagrama:

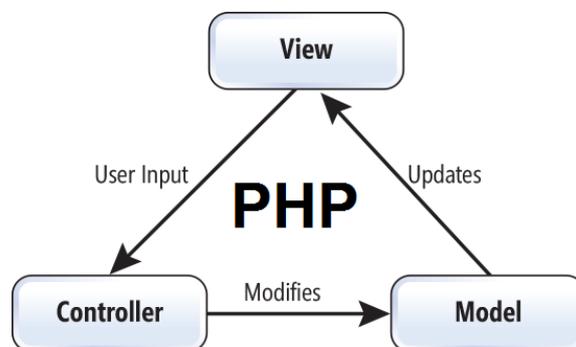


Figura 5.4: Diagrama del patrón MVC.

A continuación se muestra un ejemplo del código de un controlador en la parte de gestión.

```
9 include_once '../..//modelo/CentrosModel.php';
10
11 $accion = $_REQUEST['action'];
12 $centrosDao = new CentrosModel();
13
14 /**
15  * El switch comprueba que accion se deber realizar y ejecuta el método correspondiente
16  */
17 switch ($accion) {
18     case "add":
19         addRequest();
20         break;
21     case "del":
22         deleteRequest();
23         break;
24     case "upd":
25         updateRequest();
26         break;
27 }
28
29 /**
30  * addRequest()
31  * El método se ejecuta cuando recibe una petición de add.
32  */
33 function addRequest(){
34     global $centrosDao;
35     $codigo = $_POST["codigo"];
36     $nombre = $_POST["nombre"];
37     $centrosDao-> addCentro($codigo, $nombre);
38     $centrosDao->closeDB();
39     header('location: /serviciocita/gestion/centros/gescentros.php');
40 }
41
42 /**
43  * deleteRequest()
44  * El método se ejecuta cuando recibe una petición de delete.
45  */
46 function deleteRequest(){
47     global $centrosDao;
48     $id = $_REQUEST['id'];
49     $centrosDao-> deleteCentro($id);
50     $centrosDao->closeDB();
51     header('location: /serviciocita/gestion/centros/gescentros.php');
52 }
53
54 }
55
56 /**
57  * updateRequest()
58  * El método se ejecuta cuando recibe una petición de update.
59  */
60 function updateRequest(){
61     global $centrosDao;
62     $centrosDao-> updateCentro($_POST["id"], $_POST["codigo"], $_POST["nombre"]);
63     $centrosDao->closeDB();
64     header('location: /serviciocita/gestion/centros/gescentros.php');
65 }
66
```

Figura 5.5: Parte del código del controlador de gestión de centros.

Como se ve en la figura, en primer lugar se inicializa el o los modelos que se van a utilizar y se almacena la petición del usuario proveniente de la vista en la variable acción. Posteriormente se realiza un switch sobre dicha variable para que el controlador decida qué método debe ejecutarse. En el caso de la figura [5.5], si recibe una petición add, se ejecutará el método “addRequest” encargado de añadir un nuevo centro.

## Otros detalles de implementación

En el apartado anterior se explica la estructura general que se ha seguido para desarrollar la aplicación, en este apartado, se van a explicar algunos detalles más concretos y destacables o curiosos de la aplicación.

### Asignación de turnos

En primer lugar cabe destacar la estrategia seguida para asignar los turnos a las citas configuradas por los administradores. Como se ha comentado en el apartado 4.2.1 inicialmente se planteó el uso de una tabla en la base de datos llamada `turnos_cp_contador`, que mantuviera el último turno de cada cola para ir incrementándolo con cada reserva, de esta forma el turno de la cita se asignaba a la hora de que el cliente concertara la cita. Esta estrategia presentaba un error ya que si un usuario reservaba una cita con un horario anterior a una cita ya reservada, su turno sería inferior. Por ejemplo si alguien ha reservado cita a la 10:00 y se le ha asignado el turno 3, alguien que reservara a las 9:00 se le asignaría el turno 4.

Para solucionar este problema, lo que se decidió hacer es asignar los turnos durante la creación de citas, en lugar de asignarlos en el momento de la reserva. De esta forma, el administrador puede configurar todas las citas en el calendario de la manera que desee y una vez estén configuradas puede asignar los turnos. La asignación se realiza de manera automática y el turno se incrementa en relación a la hora de la cita. Una vez asignados los turnos, las citas pasan a estar disponibles para los clientes. Con esta estrategia no es necesario el uso de la tabla `turnos_cp_contador`.

### Sistema de CAPTCHA

Una característica que se ha utilizado para reforzar la seguridad de la aplicación es un sistema de CAPTCHA (Automated Public Turing test to tell Computers and Humans Apart) que incapacita el registro en el sistema de bots a través de un simple test que puede ser únicamente completado por humanos. Para ello se ha utilizado el servicio reCAPTCHA, un servicio de Google que ofrece la protección de bots y spam gracias a sus motores de análisis y su CAPTCHAs adaptables. Se decidió utilizar este sistema ya que el sistema de Google para la resolución de CAPTCHAs es uno de los más cómodos que existen para el usuario.

Para utilizar es necesario tener una cuenta de Google y registrar el dominio en el que se va a utilizar el servicio, una vez hecho esto, se deben importar las librerías necesarias, también se debe añadir el código necesario para visualizar el sistema en el formulario de registro y por último realizar una serie de operaciones en el back-end del registro para validar que el captcha ha sido completado con éxito.

El sistema es muy sencillo de utilizar para el usuario, únicamente debe hacer clic sobre el recuadro al lado de "No soy un robot" si todo va bien el registro se completará con éxito.



Figura 5.6: Sistema reCAPTCHA de Google.

### Plugins utilizados

Otros detalles destacables son el uso de algunos plugins como time-pickers y date-pickers, que ofrecen al usuario interfaces para la introducción de fechas y horas, dos acciones muy comunes en la aplicación que se ha desarrollado. JQuery ofrece un date-picker bastante adecuado que permite la configuración de bastantes opciones como la limitación de fechas seleccionables que ha sido utilizada para este proyecto. Para la selección de horas se ha utilizado un plugin gratuito llamado bootstrap clock-picker que ofrece una interfaz de selección de hora bastante similar a los dispositivos Android. Al selector de fecha se le ha cambiado el estilo para adaptarlo a la interfaz de la aplicación. El resultado es el siguiente:



Figura 5.7: Plugin date-picker de JQuery.

Por otro lado el plugin de selección de hora tiene el siguiente aspecto:

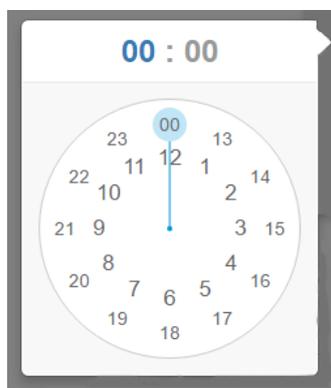


Figura 5.8: Plugin time-picker de JQuery.

Una de las partes fundamentales de la aplicación es la gestión de las citas que son añadidas por los administradores. Para conseguir una gestión cómoda, fácil y visual era necesario utilizar

algún calendario interactivo que diera la posibilidad de navegar entre sus días y ver las citas que hay programadas. Para ello se ha utilizado un plugin gratuito llamado Monthly Calendar [9]. Este plugin ofrece la interfaz de un calendario en el que se pueden añadir eventos y seleccionar días y eventos.

El plugin lee los eventos a través de un fichero XML, con una estructura determinada que sigue la forma siguiente: una etiqueta principal "monthly" que recoge todos los eventos que aparecerán en el calendario con las etiquetas "event" para cada evento o en este caso cita, dentro de event se recogen los datos de cada evento con las etiquetas "id", "nombre", "fechaInicio", "horaInicio", "color", "url".

Por tanto, para poder cargar las citas en el calendario, se ha tenido que desarrollar un método que lea las citas de la base de datos y genere este fichero XML dinámicamente el cual se le proporciona al plugin para la visualización de las citas en el calendario.

Una vez generado el fichero, el calendario tiene el aspecto de la figura [5.9]

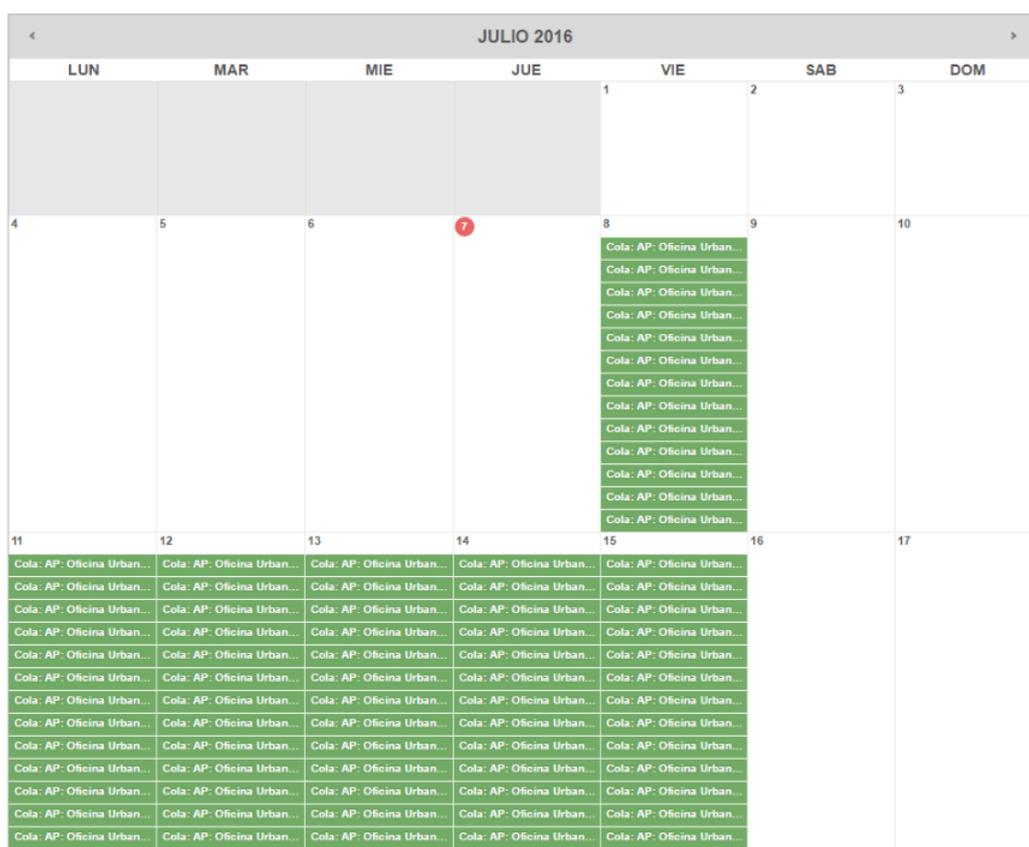


Figura 5.9: Plugin Monthly Calendar.

Si se hace clic en uno de los días se pueden ver las citas con algo más de detalle y si hacemos clic en una cita en concreto se muestra una nueva página con todos los detalles de la cita seleccionada gracias al sistema de URLs que posee el plugin.



Figura 5.10: Plugin Monthly Calendar al hacer clic en un día.

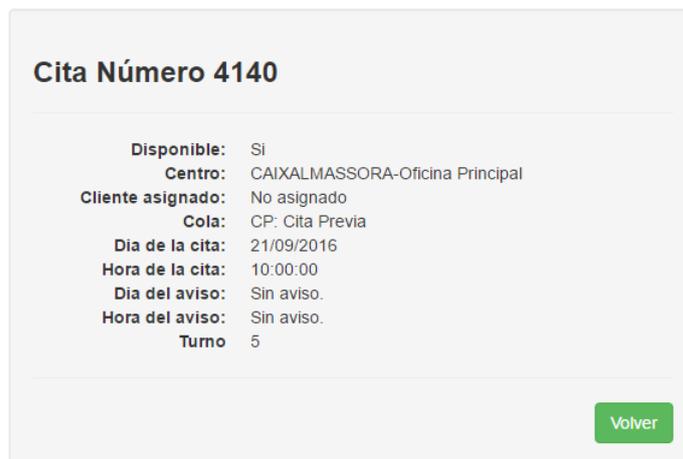


Figura 5.11: Plugin Monthly Calendar al hacer clic en un a cita.

## 5.2. Verificación y validación

Para la verificación y validación de la aplicación, el software fue sometido a test de aceptación por los algunos miembros de la entidad. Estos test permiten descubrir problemas de usabilidad o errores que los usuarios se encuentran al utilizar la aplicación para así corregirlos.

Es recomendable que no sea el propio desarrollador el que realice estas pruebas, ya que su perspectiva frente al proyecto es menor al haber trabajado en él, además, la objetividad probablemente también es menor al ser un producto de su creación. Por eso las pruebas fueron realizadas por otros empleados de la entidad que hicieron la función de usuario de la aplicación. A través de su experiencia se anotaron distintos errores o potenciales mejoras. Los errores que se encontraron se fueron corrigiendo progresivamente, por ejemplo secciones poco intuitivas, faltas de ayuda, errores ortográficos, también herramientas que no funcionaban como se esperaba etc. Se fueron realizando pruebas de este tipo hasta conseguir un resultado deseable.

Además de esto, los compañeros informáticos, sometieron la aplicación a pruebas más duras, como son la realización de acciones poco habituales, la prueba de casos extremos o otras pruebas de seguridad como inyecciones SQL. Todas estas pruebas permitieron corregir errores a priori difíciles de detectar.

## 5.3. Funcionamiento de la aplicación

En este apartado se va a describir de manera general como se utiliza la aplicación de cita previa desde el punto de vista del administrador y del usuario para terminar de comprender el proyecto y como los usuarios utilizarán la aplicación.

Al iniciar la aplicación el usuario observa la pantalla de registro y login del sistema. Si iniciamos sesión como administradores, encontramos todo aquello que se ha de gestionar para poder añadir citas al calendario para que puedan ser reservadas por los clientes.

Las dos primeras secciones de administración, gestión de centros y gestión de usuarios, son accesibles únicamente por el superadministrador y permiten gestionar los usuarios registrados en el sistema y los centros que ofrecen el servicio de cita previa. El resto de opciones son accesibles por todo tipo de administradores. En primer lugar se deben configurar los horarios básicos de la entidad, esto se hace en el apartado gestión de horarios donde se podrán añadir, eliminar o modificar horarios. Una vez configurados los centros y los horarios, el administrador podrá configurar las colas. Como ya se ha explicado, las colas son los servicios que se ofrecen a los clientes, las colas tendrán que tener un centro y un horario asignados de los que han sido creados anteriormente. Estos tres datos mencionados, los centros, los horarios y las colas forman los elementos básicos que el cliente deberá seleccionar en el momento de buscar citas, además del día y la hora de la cita.

A continuación el administrador puede proceder a gestionar los datos necesarios para configurar los calendarios. En gestión de periodos el administrador podrá configurar el rango de fechas en el que se quieren ofrecer citas y la separación temporal que habrá entre las citas. Por



### Gestión Servicio Cita Previa



Figura 5.12: Página de gestión.

ejemplo se puede configurar un periodo del 1 de agosto al 31 de agosto con un espacio temporal de 15 minutos entre las citas. Los periodos deberán tener asignados un horario de los que han sido configurados, por ejemplo: horario de mañana.

En la gestión de horas el administrador podrá acotar los días y las horas en las que habrá citas previas dentro de un periodo. Por ejemplo para el periodo del ejemplo anterior (del 1 de agosto al 31 de agosto con horario de mañana y espacio temporal de 15 minutos) el administrador puede indicar que se ofrecerán citas de lunes a viernes entre las 8:30 y las 14:30.

En la gestión de festivos el administrador podrá configurar los festivos del año para que no se generen citas en esos días concretos de forma automática.

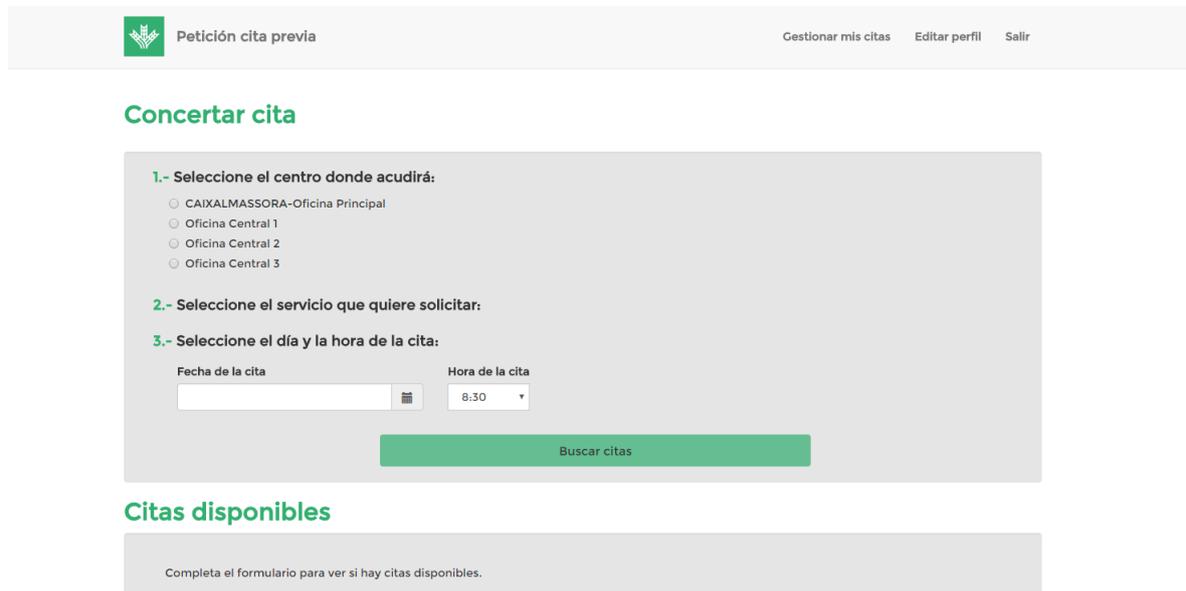
El objetivo de configurar todos estos aspectos es automatizar al máximo la configuración de los calendarios para agilizar todo lo posible las tareas del administrador.

Una vez todo configurado, si el administrador se dirige a gestión de calendarios, lo único que deberá hacer es hacer click en añadir citas, seleccionar un centro, una cola y un periodo de los que se han configurado, hacer click en aceptar y automáticamente se añadirán todas las citas que sea posible dentro de las restricciones configuradas.

Para seguir con el ejemplo anterior, cuando el administrador añada citas, se generarán citas cada 15 minutos entre las 8:30 y las 14:30 de Lunes a Viernes entre el 1 de agosto y el 31 de agosto.

Cuando el administrador termina de configurar el calendario a su gusto, podrá hacer click en asignar turnos. y esto hará que las citas pasen a estar visibles para los clientes.

Al iniciar sesión como un cliente, iremos a la página de selección de cita.



Petición cita previa

Gestionar mis citas Editar perfil Salir

### Concertar cita

1.- Seleccione el centro donde acudirá:

- CAIXALMASSORA-Oficina Principal
- Oficina Central 1
- Oficina Central 2
- Oficina Central 3

2.- Seleccione el servicio que quiere solicitar:

3.- Seleccione el día y la hora de la cita:

Fecha de la cita  Hora de la cita

Buscar citas

### Citas disponibles

Completa el formulario para ver si hay citas disponibles.

Figura 5.13: Página de selección de cita.

Como se ha mencionado, el cliente deberá seleccionar el centro, el servicio, el día y la hora que desee y podrá comprobar si existen citas disponibles, si lo desea podrá reservar la cita y configurar un aviso por correo electrónico. Si se dirige a "gestionar mis citas" podrá ver sus citas pendientes y su historial de citas. Por último si se dirige a "editar perfil" podrá modificar sus datos personales.

## Capítulo 6

# Conclusiones

A lo largo del trabajo de final de grado se ha realizado un proyecto completo de dimensiones reales que permite ofrecer un servicio de cita previa a los usuarios de una entidad bancaria. Se ha podido completar el proyecto de manera satisfactoria y se podrá poner el servicio en funcionamiento.

Ha sido muy satisfactorio poder completar un proyecto de estas características por mi cuenta y ver como se iban consiguiendo alcanzar los objetivos poco a poco hasta conseguir una aplicación completamente funcional.

Ha sido necesario enfrentarse a nuevas tecnologías con las que no había trabajado anteriormente y he comprobado que la base estudiada en la carrera proporciona una gran flexibilidad y capacidad de adaptación a la hora de enfrentarse a tecnologías nuevas ya que no ha supuesto un gran esfuerzo el aprendizaje del lenguaje PHP.

En cuanto al planteamiento del proyecto las primeras semanas fue un poco confuso comprender y sintetizar bien los detalles del proyecto que se iba a desarrollar ya que el planteamiento estaba un poco difuso y era necesario profundizar en él. Después de analizar bien el proyecto y entender lo que se quería hacer y especialmente como había que hacerlo fue todo mucho más sencillo y ágil.

Haber trabajado en una entidad como Caixa Almassora de semejantes características ha sido una buena experiencia ya que he tenido disponibles todas las tecnologías que he necesitado, me he encontrado en un buen ambiente y en general he enriquecido mis conocimientos.



# Bibliografía

- [1] Abián Ruiz, autor del Blog TRW. La importancia de los requisitos en un proyecto de software. <http://www.blogtrw.com/2011/07/la-importancia-de-los-requisitos-en-un-proyecto-de-software/>, 2011. [Consulta: Julio de 2016].
- [2] Bootstrap Team. Bootstrap. <http://getbootstrap.com/>, 2016. [Consulta: Junio de 2016].
- [3] DevsInside. Plan gestión de riesgos (dpgr) virtual class ii; versión 1.3. <https://es.scribd.com/document/151896854/Plan-de-Gestion-de-RiesgosVC2>. [Consulta: Abril de 2016].
- [4] Equipo Creately. Creately. <http://creately.com/>, 2016. [Consulta: Mayo, Julio de 2016].
- [5] Equipo ShareLatex. Sharelatex. <https://es.sharelatex.com/>, 2016. [Consulta: Julio de 2016].
- [6] Grupo TurnoGes. Turnoges. <http://www.turnoges.es/html/>, 2016. [Consulta: Junio de 2016].
- [7] JQuery Team. JQuery. <http://api.jquery.com/>, 2016. [Consulta: Junio de 2016].
- [8] JQuery UI Team. JQuery ui. <http://api.jqueryui.com/>, 2016. [Consulta: Junio de 2016].
- [9] Kthornbloom. Documentación de monthly. <http://kthornbloom.com/monthly/>, 2016. [Consulta: Junio de 2016].
- [10] Mehdi Achour, Friedhelm Betz, Antony Dovgal y otros. Documentación de php. <http://php.net/manual/es/index.php>, 2016. [Consulta: Mayo de 2016].
- [11] Oracle Corporation. Mysql. <http://www.mysql.com/>, 2016. [Consulta: Junio de 2016].
- [12] Project Management Institute. *Guía de los fundamentos para la dirección de proyectos; 5º Edición*. Project Management Institute, 2013. [Consulta: Abril de 2016].



## Anexo A

# Manual de usuario

En este anexo se va a incluir el manual de usuario que se redactó al terminar el proyecto donde se recogen todos los detalles a tener en cuenta para poder hacer un uso correcto de la aplicación. El manual está dividido en una introducción y tres apartados. En el primer apartado se describe todo aquello relacionado con el registro y el inicio de sesión en el sistema, en el segundo apartado se encuentran los detalles sobre la parte de gestión de citas a la que pueden acceder los administradores y por último en el tercer apartado se detalla todo aquello relacionado con la parte del cliente. Este documento tiene finalidad dar soporte a la entidad y resolver cualquier duda o problema que pueda surgir durante el uso de la aplicación.

Se puede acceder al manual a través del siguiente enlace:

[https://www.dropbox.com/s/gvyhbgxyl426tkv/Manual\\_usuario\\_cita\\_previa.pdf?dl=](https://www.dropbox.com/s/gvyhbgxyl426tkv/Manual_usuario_cita_previa.pdf?dl=0)

0



## Anexo B

# Documentación de los ficheros y clases

En este anexo se va a describir la ubicación, descripción y otros detalles de los distintos ficheros del proyecto. Este documento servirá como apoyo a la entidad si en un futuro se necesitan realizar cambios en el código fuente de la aplicación o añadir ciertas funcionalidades. En la siguiente imagen se muestra la estructura de ficheros del directorio raíz del proyecto sobre la cual se va a hacer referencia a los distintos ficheros.

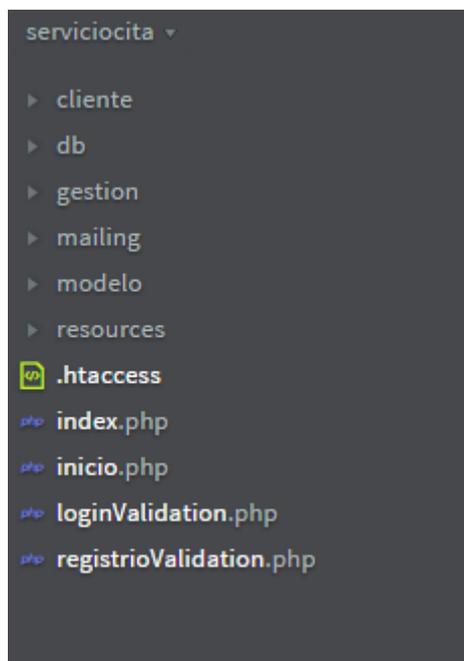


Figura B.1: Estructura de ficheros del directorio raíz.

## B.1. Ficheros de inicio, registro y login

**/index.php**

### Función

Fichero que se muestra al iniciar la aplicación. Reinicia la sesión si ya se encontraba iniciada. Contiene el formulario de login y registro y los formularios de recuperación de usuario y contraseña.

### Detalles

El formulario de login enlaza con `/loginValidation.php`.

El formulario de registro enlaza con `/registroValidation.php`

El formulario de recuperación de usuario enlaza con `/mailing/userrestore.php`

El formulario de recuperación de password enlaza con `/mailing/passrestore.php`

---

**/loginValidation.php**

### Función

Recibe los parámetros del formulario de login y comprueba su validez. Obtiene los datos del usuario de la base de datos con el método `getUsuario(usuario)`. Valida la contraseña encriptada mediante `password.verify`. Comprueba el tipo de usuario logueado y dirige a la página correspondiente.

### Detalles

El formulario de login enlaza con `/loginValidation.php`.

El formulario de registro enlaza con `/registroValidation.php`

El formulario de recuperación de usuario enlaza con `/mailing/userrestore.php`

El formulario de recuperación de password enlaza con `/mailing/passrestore.php`

---

## **/registroValidation.php**

### **Función**

Obtiene los datos del formulario de registro y añade un nuevo usuario a la base de datos si son válidos.

### **Detalles**

Realiza la validación del captcha utilizando el sistema reCAPTCHA de Google (para ello es necesario descomentar las líneas indicadas en el fichero).

Encripta la contraseña introducida por el usuario mediante password\_hash.

Comprueba si el usuario o el email ya existen con los métodos userNameExists(usuario) y emailExists(emailalerta).

Si los datos son válidos redirige a la página /cliente/inicioclientes.php.

Si existe el usuario redirige a /index.php con el código de error 2 (usuario existente).

Si existe el correo redirige a /index.php con el código de error 3 (correo existente).

Si no se ha realizado el captcha correctamente redirige a /index.php con el código de error 4 (captcha no completado).

### **Requiere**

- /resources/other/recaptchalib.php: librería de reCAPTCHA.

---

## **/inicio.php**

### **Función**

Página de inicio de los administradores. Enlaza con los distintos contenidos accesibles por los administradores.

### **Detalles**

Comprueba la sesión, si es válida, muestra el contenido de la página, si no es válida, redirige a /index.php.

## B.2. Ficheros de conexión y acceso a la base de datos

`/db/db.php`

### Función:

Contiene los datos de la base de datos. Genera una conexión a la base de datos mediante PDO y devuelve dicha conexión para que sea usada.

---

`/modelo/**Model.php`

### Función

Los ficheros ubicados en `/modelo` con un nombre terminado en `**Model.php` son objetos de acceso a la base de datos (DAOs) que realizan transacciones con la base de datos.

### Detalles

Obtienen la conexión a la base de datos en el constructor de la clase. Tienen un método `CloseDB()` para cerrar la conexión.

Contienen todas las operaciones necesarias sobre la base de datos para añadir, eliminar, modificar u obtener datos.

### Requiere

- `/db/db.php`: conexión a la base de datos.
-

## B.3. Ficheros para envío de correos

`/mailing/passrestore.php`

### Función

Recoge el usuario del formulario de recuperación de contraseña, obtiene los datos del usuario, genera una contraseña aleatoria, actualiza los datos del usuario y envía un correo al usuario con la nueva contraseña aleatoria.

### Detalles

Si el usuario introducido en el formulario no es válido vuelve a `/index.php` con el código “bu” mostrando el mensaje de error de usuario no existente.

Si se realiza el envío con éxito se vuelve a `/index.php` y se muestra un mensaje informativo al usuario.

### Requiere

- `/db/db.php`
  - `/resources/other/generadorPasswords.php`
- 

`/mailing/userrestore.php`

### Función

Recoge la dirección de correo del formulario de recuperación de usuario, obtiene los datos del usuario y envía un correo al usuario indicando su nombre de usuario.

### Detalles

Si el correo introducido en el formulario no es válido vuelve a `/index.php` con el código “bm” mostrando el mensaje de error de correo no existente.

Si se realiza el envío con éxito se vuelve a `/index.php` y se muestra un mensaje informativo al usuario.

### Requiere

- `/db/db.php`
-

## B.4. Ficheros de recursos

*/resources/css/*

### **Función**

Directorio que contiene ficheros de estilo para el diseño de las distintas páginas.

### **Detalles**

*/resources/css/estilo.css* contiene los estilos para la parte de gestión.

*/resources/css/estiloCliente.css* contiene los estilos para la parte del cliente.

El resto de ficheros son estilos de los distintos frameworks o plugins utilizados.

---

*/resources/img/*

### **Función**

Directorio que contiene imágenes utilizadas en la aplicación.

### **Detalles**

La imagen del logo es una imagen genérica que se puede cambiar.

---

*/resources/js/*

### **Función**

Directorio que contiene documentos javascript requeridos por diferentes páginas de la aplicación.

### **Detalles**

*/resources/js/inicializaciones.js* inicializa el plugin de tablas para visualizar tablas y el plugin de datepicker para seleccionar fechas en un calendario.

El resto de ficheros son funciones de los distintos frameworks o plugins utilizados.

---

`/resources/other/`

### **Función**

Directorio que contiene otros tipo de ficheros requeridos por el sistema.

### **Detalles**

`/resources/other/citas.xml` contiene toda la información de las citas previas requerida por el plugin monthly para el calendario de la parte de gestión.

`/resources/other/generadorDeIntervalos.php` contiene un método para generar las horas de las citas en un intervalo horario personalizable.

`/resources/other/generadorPasswords.php` genera una contraseña aleatoria para la recuperación de contraseña.

`/resources/other/recaptchalib.php` librería del plugin reCAPTCHA de Google.

---

`/resources/templates/plantillaCliente.php`

### **Función**

Fichero que contiene el formato de la página básica de la parte de clientes.

### **Detalles**

Se incluye en todos los ficheros de la parte de cliente que necesiten este formato.

---

## B.5. Ficheros de la parte de cliente

`/cliente/inicioclientes.php`

### Función

Página de inicio de los clientes donde se puede realizar la reserva de citas previas y acceder a las distintas secciones de la parte de clientes.

### Detalles

Comprueba la sesión, si es válida muestra el contenido y si no redirige a `/index.php`.  
Obtiene los datos de la búsqueda realizada y muestra las citas disponibles existentes.  
Llama al controlador cuando se realiza una búsqueda o una reserva.

### Requiere

- `/resources/templates/plantillaClientes.php`
- `/cliente/clientescontroller.php`

---

`/cliente/clientesController.php`

### Función

Recibe las peticiones del cliente y realiza las acciones necesarias.

### Detalles

Contiene los métodos necesarios para satisfacer las peticiones de buscar citas, reservar citas, cancelar citas, cambiar citas, editar perfil y cambiar contraseña.

Contiene otros métodos requeridos por la vistas de la parte de clientes.

Utiliza las clases modelo para interactuar con la base de datos.

### Requiere

- `/modelo/CentrosModel.php`
- `/modelo/CalendariosModel.php`

- /modelo/ColasModel.php
- /modelo/UsuariosModel.php
- /modelo/HorariosModel.php

---

## **/cliente/gestionCitas.php**

### **Función**

Página de gestión de citas donde los usuarios pueden consultar sus citas pendientes y ver su historial de citas.

### **Detalles**

Comprueba la sesión, si es válida muestra el contenido y si no redirige a /index.php.

Realiza peticiones al controlador para cambiar o cancelar una cita.

### **Requiere**

- /modelo/CalendariosModel.php
- /modelo/CentrosModel.php
- /modelo/ColasModel.php
- /resources/templates/plantillaCliente.php

---

## **/cliente/perfil.php**

### **Función**

Página de perfil del usuario. Muestra los datos del usuario y permite su modificación.

### **Detalles**

Comprueba la sesión, si es válida muestra el contenido y si no redirige a /index.php.

Realiza peticiones al controlador para editar la contraseña y para editar los datos personales.

## Requiere

- /modelo/CalendariosModel.php
  - /modelo/CentrosModel.php
  - /modelo/ColasModel.php
  - /resources/templates/plantillaCliente.php
-

## B.6. Ficheros de la parte de gestión

`/gestion/calendarios/`

### Función

Directorio que contiene el controlador y las vistas para la gestión de calendarios.

### Detalles

- `/gestion/calendarios/calendariosController.php` atiende las peticiones del usuario y realiza las acciones necesarias. Contiene el método que genera el fichero `/resources/other/citas.xml` a partir de las citas almacenadas en la base de datos.
- `/gestion/calendarios/gescalendarios.php` muestra el calendario con todas las citas disponibles en el sistema. Las citas se leen del fichero `/resources/other/citas.xml`. Realiza peticiones al controlador para añadir citas, eliminar un rango de citas, asignar turnos y eliminar citas sin turno asignado. Comprueba la sesión antes de mostrar el contenido.
- `/gestion/calendarios/infoFecha.php` pagina que se muestra al hacer click sobre una cita en concreto del calendario. Si no tiene turno asignado se puede modificar. Realiza peticiones al controlador para modificar citas. Comprueba la sesión antes de mostrar el contenido.

### Requiere

- `calendariosController`
  - `/modelo/CalendariosModel.php`
  - `/modelo/CentrosModel.php`
  - `/modelo/ColasModel.php`
  - `/modelo/PeriodosModel.php`
  - `/modelo/UsuariosModel.php`
  - `/modelo/HorasModel.php`
  - `/modelo/FestivosModel.php`
  - `/modelo/HorariosModel.php`
  - `/resources/other/generadorIntervalos.php`
- `gescalendarios`
  - `/gestion/calendarios/calendariosController.php`
- `infoFecha`
  - `/modelo/CalendariosModel.php`

- /modelo/UsuariosModel.php
  - /modelo/CentrosModel.php
  - /modelo/ColasModel.php
- 

## **/gestion/centros/**

### **Función**

Directorio que contiene el controlador y las vista para la gestión de centros.

### **Detalles**

- **/gestion/centros/centroscontroller.php**: atiende las peticiones del usuario y realiza las acciones necesarias. Atiende peticiones de añadir, editar y eliminar centros.
- **/gestion/centros/gescentros.php**: muestra una tabla con los distintos centros almacenados en el sistema. Realiza peticiones al controlador para añadir, eliminar o modificar centros. Comprueba la sesión antes de mostrar el contenido.

### **Requiere**

- centroscontroller
    - /modelo/CentrosModel.php
  - gescentros
    - /modelo/CalendariosModel.php
    - /modelo/CentrosModel.php
- 

## **/gestion/colas/**

### **Función**

Directorio que contiene el controlador y las vista para la gestión de colas.

### **Detalles**

**/gestion/colas/colasController.php** atiende las peticiones del usuario y realiza las acciones necesarias. Atiende peticiones de añadir, editar y eliminar colas.

**/gestion/colas/gesColas.php** muestra una tabla con las distintas colas almacenados en el sistema. Realiza peticiones al controlador para añadir, eliminar o modificar colas. Comprueba la sesión antes de mostrar el contenido.

### Requiere

- colasController
    - /modelo/ColasModel.php
  - gesColas
    - /modelo/CalendariosModel.php
    - /modelo/CentrosModel.php
    - /modelo/CalendariosModel.php
    - /modelo/HorariosModel.php
- 

**/gestion/festivos/**

### Función

Directorio que contiene el controlador y las vista para la gestión de festivos.

### Detalles

**/gestion/festivos/festivosController.php** atiende las peticiones del usuario y realiza las acciones necesarias. Atiende peticiones de añadir, editar, clonar año, eliminar año y eliminar todos los festivos.

**/gestion/festivos/gesFestivos.php** muestra una tabla con los distintos festivos almacenados en el sistema. Realiza peticiones al controlador para añadir y modificar festivos, clonar año, eliminar año y eliminar festivos. Comprueba la sesión antes de mostrar el contenido.

### Requiere

- festivosController
  - /modelo/FestivosModel.php
- gesFestivos
  - /modelo/FestivosModel.php

---

**`/gestion/horarios/`**

### **Función**

Directorio que contiene el controlador y las vista para la gestión de horarios.

### **Detalles**

**`/gestion/horarios/horariosController.php`** atiende las peticiones del usuario y realiza las acciones necesarias. Atiende peticiones de añadir, editar, eliminar uno y eliminar todos los horarios.

**`/gestion/horarios/gesHorarios.php`** muestra una tabla con los distintos horarios configurados en el sistema. Realiza peticiones al controlador para añadir, eliminar o modificar horarios. Comprueba la sesión antes de mostrar el contenido.

### **Requiere**

- `horariosController`
  - `/modelo/HorariosModel.php`
- `gesHorarios`
  - `/gestion/horarios/horariosController.php`

---

**`/gestion/horas/`**

### **Función**

Directorio que contiene el controlador y las vista para la gestión de horas.

### **Detalles**

**`/gestion/horas/horasController.php`** atiende las peticiones del usuario y realiza las acciones necesarias. Atiende peticiones de añadir, editar, eliminar una y eliminar todas las horas.

**`/gestion/horas/gesHoras.php`** muestra una tabla con las distintas horas configuradas en el sistema. Realiza peticiones al controlador para añadir, modificar y eliminar horas. Comprueba la sesión antes de mostrar el contenido.

## Requiere

- horasController
    - /modelo/HorasModel.php
    - /modelo/CalendariosModel.php
    - /modelo/PeriodosModel.php
    - /modelo/HorariosModel.php
  - gesHoras
    - /gestion/horas/horasController.php
- 

**/gestion/periodos/**

## Función

Directorio que contiene el controlador y las vista para la gestión de periodos.

## Detalles

**/gestion/periodos/periodosController.php** atiende las peticiones del usuario y realiza las acciones necesarias. Atiende peticiones de añadir, editar y eliminar periodos.

**/gestion/periodos/gesPeriodos.php** muestra una tabla con los distintos periodos configurados en el sistema. Realiza peticiones al controlador para añadir, eliminar o modificar periodos. Comprueba la sesión antes de mostrar el contenido.

## Requiere

- periodosController
  - /modelo/PeriodosModel.php
  - /modelo/CalendariosModel.php
- gesPeriodos
  - /modelo/PeriodosModel.php
  - /modelo/HorariosModel.php

---

`/gestion/usuarios/`

### **Función**

Directorio que contiene el controlador y las vista para la gestión de usuarios.

### **Detalles**

`/gestion/usuarios/usuariosController.php` atiende las peticiones del usuario y realiza las acciones necesarias. Atiende peticiones de añadir, editar y eliminar usuarios.

`/gestion/usuarios/gesUsuarios.php` muestra una tabla con los distintos usuarios configurados en el sistema. Realiza peticiones al controlador para añadir, eliminar o modificar usuarios. Comprueba la sesión antes de mostrar el contenido.

### **Requiere**

- UsuariosModel
  - `/modelo/HorariosModel.php`
- gesUsuarios
  - `/modelo/UsuariosModel.php`