

**ESCUELA POLITÉCNICA
SUPERIOR DE CÓRDOBA**
Universidad de Córdoba



TRABAJO FIN DE GRADO

Grado en Ingeniería

Informática

Reserva Cultura

**Aplicación Web y Android para la programación
de oferta cultural para la tercera edad**

Manual Técnico

Autor: Antonio Cabezas Jarabo

Director: Rafael Muñoz Salinas

Fecha: Septiembre de 2024



UNIVERSIDAD DE CÓRDOBA

D. Rafael Muñoz Salinas,

profesor del departamento de Informática y Análisis numérico de la Escuela Politécnica Superior de la Universidad de Córdoba.

INFORMA

Que el presente trabajo fin de grado titulado Aplicación Web y Android para la programación de oferta cultural para la tercera edad, que constituye la memoria presentada por D. Antonio Cabezas Jarabo para aspirar al grado de Ingeniería Informática, ha sido realizado bajo mi dirección en la Escuela Politécnica Superior de la Universidad de Córdoba reuniendo, a mi juicio, las condiciones necesarias exigidas para este tipo de trabajos. Y para que así conste se expide y firma el presente informe en Córdoba, 9 de Septiembre de 2024.

EL AUTOR

D. Antonio Cabezas Jarabo

EL DIRECTOR

Fdo: Rafael Muñoz Salinas

Índice general

1. Introducción	3
2. Definición del problema	5
2.1 Identificación del problema real	5
2.2 Identificación del problema técnico	6
3. Objetivos	15
4. Antecedentes	17
4.1 Gym Pass	17
4.2 Padel Manager	19
5. Restricciones	21
5.1 Factores dato	21
5.2 Factores estratégicos	22
6. Recursos	25
6.1 Recursos humanos	25
6.2 Recursos software	25
6.3 Recursos hardware	26
7. Especificación de requisitos	27
7.1 Requisitos funcionales	27
7.2 Requisitos no funcionales	30
8. Especificación del sistema	31
8.1 Modelo de datos relacional	31
8.2 Estructura de la aplicación web	55
8.3 Estructura de la aplicación móvil	58

8.4	Diseño de la interfaz de usuario aplicación web	61
8.5	Diseño de la interfaz de usuario aplicación móvil	64
8.6	Desarrollo de las funciones de la app web	67
8.7	Desarrollo de las funciones de la app móvil	87
9.	Pruebas del sistema	96
9.1	Pruebas en el entorno de desarrollo	96
9.2	Pruebas en Aplicación Web	99
10.	Conclusiones	102
10.1	Conclusiones generales	102
10.2	Conclusiones de las pruebas	106
10.3	Futuras mejoras	106
11.	Bibliografía	108

Índice de tablas

Tabla 1.-Distribución del proyecto	12
Tabla 2.-Caso de uso, Inicio de sesión en aplicación web	87
Tabla 3.-Caso de uso, Cerrado de sesión en la aplicación web	88
Tabla 4.-Caso de uso, Listar áreas	89
Tabla 5.-Caso de uso, Creación de área	90
Tabla 6.-Caso de uso, Editar área	91
Tabla 7.-Caso de uso, Borrar área	91
Tabla 8.-Caso de uso, Listar centros	92
Tabla 9.-Caso de uso, Creación de centro	93
Tabla 10.-Caso de uso, Editar centro	93
Tabla 11.-Caso de uso, Borrar centro	94
Tabla 12.-Caso de uso, Listar usuarios	95
Tabla 13.-Caso de uso, Registro de usuario	96
Tabla 14.-Caso de uso, Editar usuario	96
Tabla 15.-Caso de uso, Dar de baja/alta a un usuario	97
Tabla 16.-Caso de uso, Listar clientes	98
Tabla 17.-Caso de uso, Registro de cliente	99
Tabla 18.-Caso de uso, Editar cliente	99
Tabla 19.-Caso de uso, Dar de baja/alta a un cliente	100
Tabla 20.-Caso de uso, Editar configuración	101
Tabla 21.-Caso de uso, Listar clases	102
Tabla 22.-Caso de uso, Creación de clase	103
Tabla 23.-Caso de uso, Editar clase	103
Tabla 24.-Caso de uso, Dar de baja/alta una clase	104

Tabla 25.-Caso de uso, Listar habitaciones	105
Tabla 26.-Caso de uso, Creación de habitación	106
Tabla 27.-Caso de uso, Editar habitación	106
Tabla 28.-Caso de uso, Dar de baja/alta una habitación	107
Tabla 29.-Caso de uso, Listar horarios	108
Tabla 30.-Caso de uso, Creación de horario	109
Tabla 31.-Caso de uso, Editar horario	109
Tabla 32.-Caso de uso, Borrar horario	110
Tabla 33.-Caso de uso, iniciar sesión en app móvil	112
Tabla 34.-Caso de uso, cerrar sesión en app móvil	113
Tabla 35.-Caso de uso, recordar cliente en app móvil	114
Tabla 36.-Caso de uso, elección de centro en app móvil	115
Tabla 37.-Caso de uso, elección de clase	116
Tabla 38.-Caso de uso, reserva de horario	117
Tabla 39.-Caso de uso, ver perfil	118
Tabla 40.-Caso de uso, cancelar reserva	119
Tabla 41.-Prueba 1, pantalla de carga	121
Tabla 42.-Prueba 2, pantalla de inicio de sesión	122
Tabla 43.-Prueba 3, elegir centro	122
Tabla 44.-Prueba 4, elegir clase	122
Tabla 45.-Prueba 5, reservar horario	123
Tabla 46.-Prueba 6, ver perfil	123
Tabla 47.-Prueba 7, cancelación de reserva	123
Tabla 48.-Prueba 8, cierre de sesión	124
Tabla 49.-Prueba 9, pantalla de inicio de sesión app web	124
Tabla 50.-Prueba 10, listar centros app web	125
Tabla 51.-Prueba 11, crear clase	125
Tabla 52.-Prueba 12, registrar cliente	126

Índice de Figuras

Figura 1.-Distribución gráfica del proyecto	13
Figura 2.- Capturas 1, 2 y 3 GymPass	30
Figura 3.- Capturas 1, 2 y 3 Padel Manager	31
Figura 4.- Base de datos SQL en documentos	43
Figura 5.- Pantalla de Inicio de Sesión	73
Figura 6.- Pantalla de horarios	74
Figura 7.- Pantalla de clases	74
Figura 8.- Pantalla de habitaciones	75
Figura 9.- Pantalla de áreas	76
Figura 10.- Pantalla de centros	77
Figura 11.- Pantalla de usuarios	78
Figura 12.- Pantalla de clientes	79
Figura 13.- Pantalla de configuración	80
Figura 14.- Pantalla de perfil	81
Figura 15.- Pantalla de carga ReservaCultura	82
Figura 16.- Pantalla de login ReservaCultura	83
Figura 17.- Pantalla de bienvenida ReservaCultura	84
Figura 18.- Pantallas de clases ReservaCultura	85
Figura 19.- Pantalla de horarios	86
Figura 20.- Pantalla perfil cliente	87

Parte I

MANUAL TÉCNICO

Capítulo 1

Introducción

En la actualidad, el bienestar y la calidad de vida de las personas mayores son de vital importancia. El aumento de la esperanza de vida y el envejecimiento de la población hacen patente la necesidad de programas culturales y educativos específicos para este grupo. Los centros cívicos se convierten en espacios esenciales para promover el bienestar integral de la tercera edad, brindando una variedad de clases y actividades que fomentan el aprendizaje continuo y el desarrollo personal.

Este proyecto tiene como objetivo principal optimizar la gestión de las clases ofrecidas a las personas de la tercera edad en los centros cívicos. A través de la implementación de una aplicación móvil y una plataforma web para la administración de reservas y la creación de clases, se busca facilitar el acceso a estas, garantizando una experiencia fluida y personalizada para los usuarios.

La aplicación móvil permitirá a los usuarios:

- Ver las clases disponibles en los centros cívicos que las ofrezcan
- Reservar fácilmente sus clases desde la comodidad de sus teléfonos móviles

La plataforma web de administración permitirá:

- Gestionar las reservas por parte del personal del centro cívico
- Gestionar los horarios de las clases por parte de los profesores de las mismas o del personal con permisos del centro cívico
- Gestionar cancelaciones

En resumen, este proyecto busca fortalecer y enriquecer la oferta de clases para la tercera edad en los centros cívicos, promoviendo su participación activa en actividades culturales y educativas que contribuyan a su bienestar y calidad de vida. La implementación de herramientas tecnológicas modernas facilitará el acceso y la gestión de estas clases, garantizando una experiencia satisfactoria para todos los involucrados.

Capítulo 2

Definición del problema

En este capítulo se definirá el problema real y técnico a resolver con este proyecto.

2.1 Identificación del problema real

Muchas personas de la tercera edad no conocen las clases que los centros cívicos de su zona imparten, y mucho menos las clases de los demás.

Se busca permitir a los usuarios de la aplicación móvil conocer, reservar y cancelar las clases disponibles en todos los centros cívicos.

La aplicación web está conectada a la misma base de datos que la aplicación móvil, en la que una persona o personas tendrán los permisos necesarios para crear, editar y borrar clases, centros cívicos y otros usuarios de la base de datos.

En la aplicación web se podrán conectar tanto administradores del centro cívico como profesores, esto se podrá realizar mediante un sistema de autenticación que tendrá como mínimo las funcionalidades de inicio de sesión y registro de usuarios.

2.2 Identificación del problema técnico

En este capítulo se utilizará la PDS, una técnica de ingeniería que permite la completa identificación del problema técnico mediante la elaboración de respuestas a una lista de preguntas estandarizadas.

2.2.1 Funcionamiento

El funcionamiento de la aplicación es sencillo, el usuario deberá iniciar sesión o registrarse, si se registra por primera vez deberá ser de forma presencial en un centro cívico entregando su DNI a un administrador para que le den de alta y luego poder iniciar sesión. Una vez iniciada la sesión podrá acceder a la lista de centros cívicos, y dentro de cada uno de ellos la lista de clases disponibles para reservar. Además, disponen de un botón en la pantalla principal para acceder a la lista de las reservas de cada uno de ellos.

En el caso de los administradores o los profesores en la aplicación web tendrán un menú con las diferentes ventanas. Los administradores pueden crear clases, horarios y aulas e incluso dar de alta a otros profesores. Los profesores sin embargo solamente pueden reservar las aulas.

2.2.2 Entorno

El entorno de la aplicación móvil puede dividirse en dos partes:

- Entorno de desarrollo: Será Android Studio [1]. Android Studio es un IDE que permite desarrollar aplicaciones Android e ISO a través de Flutter [2], con el lenguaje Dart [3]. Android Studio también permite la construcción de interfaces de usuario gracias al uso de componentes llamados layouts que en realidad son archivos xml interconectados que usan etiquetas de elementos. Mediante atributos podemos customizar estos elementos, creando así la interfaz de usuario.
- Entorno de uso: Será donde y cuando necesite el usuario. La app puede ser usada en modo retrato y en tablet aunque ha sido diseñada para ser usada en dispositivos móviles y en modo vertical.

El entorno de la aplicación web puede dividirse en dos partes:

- Entorno de desarrollo: Será PHPStorm [4]. PHPStorm es un IDE que permite desarrollar aplicaciones web con PHP [5], con el framework Symfony [6].
- Entorno de uso: Será o bien el administrador del centro cívico desde el mismo centro o bien el profesor desde cualquier navegador

2.2.3 Vida esperada

La vida esperada del proyecto será definida por los centros cívicos, si su uso crece y se extiende, la aplicación recibirá retroalimentación de sus usuarios y podrá ser mejorada mediante actualizaciones. Esto hará que sea un producto más duradero y eficaz.

2.2.4 Ciclo de mantenimiento

La primera versión del producto no necesitará mantenimiento obligatoriamente, pero es posible que el producto sufra actualizaciones si los usuarios necesitan nuevas funcionalidades o subsanación de errores.

2.2.5 Competencia

No existen aplicaciones enfocadas a la reserva de clases de centros cívicos, pero si existen otras aplicaciones para reservas y seguimiento de las mismas.

Este tema se desarrollará mucho más a fondo en el capítulo de antecedentes en este documento.

2.2.6 Aspecto externo

La interfaz de usuario será usable e intuitiva siguiendo pautas de diseño de interfaces para dispositivos móviles y aplicaciones web. Se desarrolla usando medidas relativas que permiten adaptarse al tamaño de pantalla de cada dispositivo, tanto la aplicación móvil o la web.

La aplicación contará con un logo propio que identificará la marca del proyecto y hará que sea reconocible en el menú de aplicaciones del usuario.

2.2.7 Estandarización

El desarrollo se llevará a cabo utilizando distintos estándares, proporcionados por la documentación oficial de Android, estándares de escritura de código, estándares relacionados con patrones de diseño y librerías estándar recomendadas para la implementación de buenas prácticas.

El entorno de desarrollo Android Studio y PHPStorm también cuenta con un sistema de consejos que examina el código en la lógica de la aplicación y en las propias vistas, generando informes en los que señala las secciones que pueden ser reconfiguradas para seguir los estándares definidos.

2.2.8 Calidad y fiabilidad

Proporcionar calidad y fiabilidad es uno de los objetivos de Reserva Cultura, demostrando así que un producto gratuito puede ofrecer la misma o mejor satisfacción y buen funcionamiento como un producto de pago.

Se realizarán varios tipos de pruebas previas a la salida del producto para garantizar que tenga un buen funcionamiento, pero también, se instará a los usuarios a que cumplan los requisitos de funcionamiento de Reserva Cultura, principalmente serán dos:

- Conexión a internet
- Versión mínima de Android e IOS

El segundo requerimiento será proporcionado automáticamente por la generación automática de archivos de Flutter para Android e IOS, solo permitirá la descarga de la aplicación a versiones superiores de la mínima requerida. El primer requerimiento será comprobado al iniciar la aplicación.

Además, se implementará un sistema en la aplicación que permita detener el acceso de los usuarios cuando haya un fallo de seguridad, de funcionamiento o de cualquier tipo que afecte a la experiencia de estos usuarios.

2.2.9 Programa de tareas

Todo proyecto para su correcta realización necesita tener una distribución ordenada de tareas que describan en conjunto las duraciones del proyecto, el coste que supondrá y los recursos que se necesitarán.

Las fases que constituirán este proyecto son:

- Fase de estudio, formación e investigación

La primera fase será la encargada de realizar un estudio previo sobre las diferentes herramientas, tecnologías y métodos a usar durante el proyecto. También se elegirá una metodología que optimice estos recursos para llevar a cabo el proyecto en el tiempo estimado.

- Fase de análisis

Se realizará un análisis de los requisitos que debe cumplir el proyecto. Los distintos requisitos obtenidos se clasifican en funcionales (funciones del sistema software) o no funcionales (restricciones en el diseño o implementación).

- Fase de diseño

Diseño de la base de datos relacional [7], interfaces de usuario, patrones de diseño, distribución de actividades, diseño del sistema de autenticación y diseño de las pruebas.

- Fase de implementación

Una vez finalizada la fase de diseño del sistema, se procederá a la implementación del mismo, es decir la realización de un software funcional y usable que cumpla los objetivos y resuelva el problema planteado.

- Fase de pruebas

Se diseñarán pruebas específicas para comprobar la respuesta de la aplicación, esto será útil para detectar errores y solucionarlos de una forma eficiente antes de crear el producto final.

- Fase de documentación

En la fase de documentación se realizará toda la documentación necesaria.

La documentación referente al proyecto se desarrollará cuando hayan finalizado todas las fases del proyecto.

Se incluirá un manual de usuario que explique de forma no técnica qué funcionalidades se pueden llevar a cabo en la aplicación y un manual de código que explique detalladamente el código fuente de la aplicación para futuro personal técnico que revise el proyecto.

La siguiente Tabla 1 muestra la distribución del proyecto:

FASE	TIEMPO
Estudio, Formación e Investigación	30
Análisis	40
Diseño	40
Implementación	130
Pruebas	20
Documentación	40

Tabla 1.-Distribución del proyecto

La siguiente Figura 1 muestra de forma gráfica la distribución del proyecto:

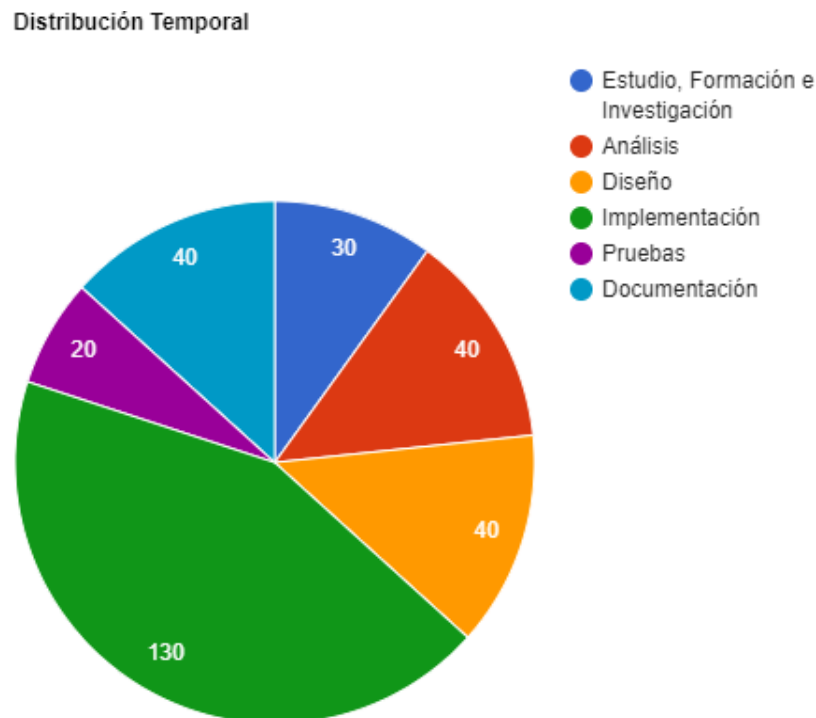


Figura 1.-Distribución gráfica del proyecto

2.2.10 Pruebas

Se realizarán pruebas en el entorno de desarrollo. Se diseñarán pruebas que se ejecutarán en el entorno de desarrollo que permitirán controlar la eficiencia desde módulos individuales hasta el sistema completo. Habrá pruebas tanto para componentes lógicos y de rendimiento como para interfaces de usuario.

2.2.11 Seguridad

La seguridad de Reserva Cultura está respaldada por los servicios de phpmyadmin, al utilizar bases de datos y el sistema de autenticación de phpmyadmin para almacenar todos los datos, siempre están sujetos a las políticas de privacidad de estas empresas y garantizan a sus clientes.

La app Reserva Cultura se desarrollará teniendo en cuenta la normativa de protección de datos vigente.

La aplicación web se desarrollará usando los JWTToken [8] (JSON Web Token), que mantienen a los usuarios seguramente autenticados.

Capítulo 3

Objetivos

El objetivo final de este proyecto es crear una aplicación funcional que facilite la programación de oferta cultural para la tercera edad. Para llegar a este objetivo final será necesario cumplir estas acciones previas:

1. Se creará un sistema de autenticación dentro de la aplicación que permita iniciar sesión, registrarse, recordar la contraseña y cerrar sesión. Esto servirá para poder identificar a cada usuario y poder proporcionarle su respectiva información.
2. La aplicación recordará al usuario cada vez que acceda a ella si no cerró la sesión para facilitar el acceso diario.
3. Diseño y creación de una base de datos SQL [9] en PHPMyAdmin [10] que será usada por ambas aplicaciones, donde se guardará información en tablas, como pueden ser usuarios, clases, clientes, horarios...
4. Diseño e implementación de la interfaz gráfica de usuario cómoda, simple y usable.
5. Desarrollo de la lógica de la aplicación en Android Studio haciendo uso del lenguaje Dart de Flutter.

6. La aplicación comprobará los datos introducidos en la aplicación y mantendrá un correcto funcionamiento gracias al control de errores.
7. La aplicación deberá ser gratuita
8. La aplicación utilizará iconos e imágenes para que el uso sea más intuitivo y no sobrecargue de información al usuario.
9. La aplicación deberá ofrecer conexión ininterrumpida con el servidor.
10. La aplicación web deberá ofrecer acceso a la información necesaria dependiendo de los permisos de cada usuario
11. La aplicación web no debe mostrar información confidencial de los clientes de la aplicación móvil.
12. La aplicación web debe permitir las operaciones que cada usuario puede realizar dependiendo de sus permisos

Capítulo 4

Antecedentes

Existen varias aplicaciones en el mercado que fueron creadas con objetivos parecidos a Reserva Cultura, en este capítulo compararemos sus características y se estudiarán las ventajas y desventajas para comprobar que nuestro proyecto ofrece innovación y calidad al mercado actual.

Si se utiliza Google Play para buscar aplicaciones de reservas de horarios tienden a aparecer aplicaciones de reserva de clases en gimnasios, o reservas de pistas de deportes.

A continuación se presentan las más relevantes encontradas:

4.1 Gym Pass

GymPass [11] es una de las aplicaciones más populares. Dispone de una lista de las clases que se pueden reservar dependiendo del plan que hayas escogido con información adicional dentro de la misma pantalla, como: fotos, disponibilidad, información general, etc.

La principal diferencia con nuestro proyecto es el tema principal de la aplicación. Además, nuestra aplicación web permite realizar las reservas de los clientes sin necesidad de que tengan la aplicación móvil.

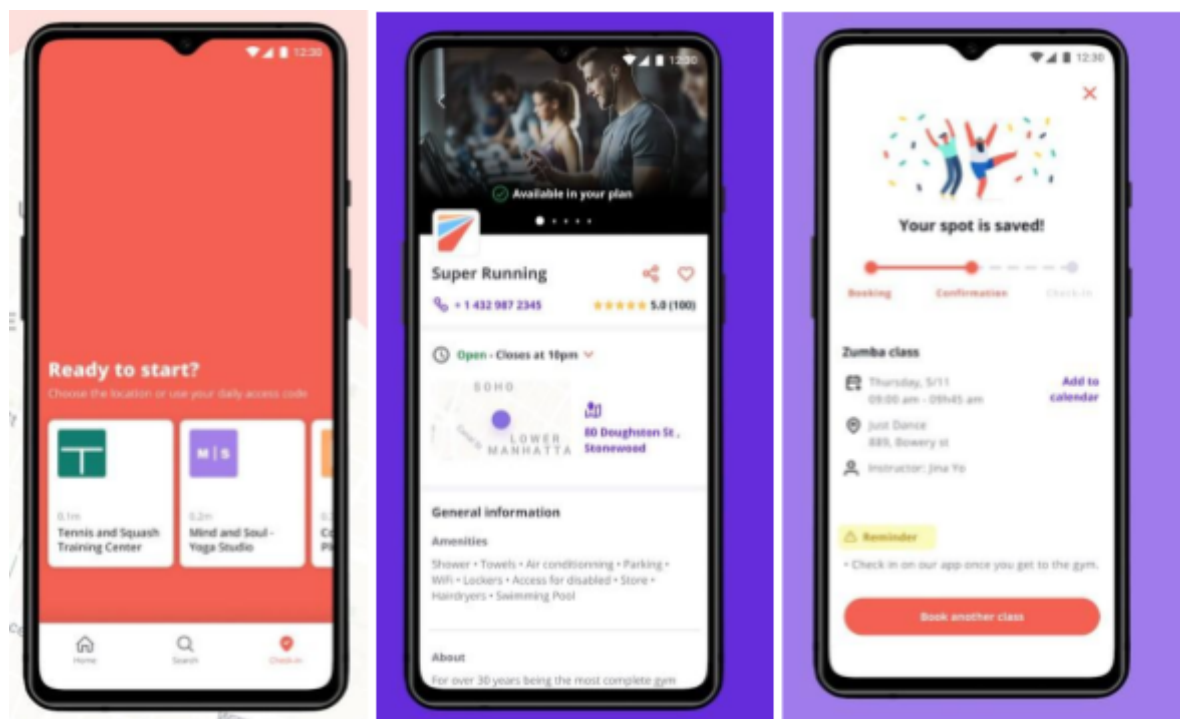


Figura 2.- Capturas 1, 2 y 3 GymPass

4.2 Padel Manager

Padel Manager [12] es una aplicación para que los integrantes de un club de padel puedan realizar las reservas fácilmente desde su teléfono.

Dispone de un tablón de anuncios del club de padel, a diferencia de Reserva Cultura, torneos a los que se pueden apuntar los miembros y una opción para crear tus propios partidos y dejar huecos abiertos para que otros miembros puedan apuntarse.

La siguiente Figura 3 muestra la galería de imágenes de Padel Manager:

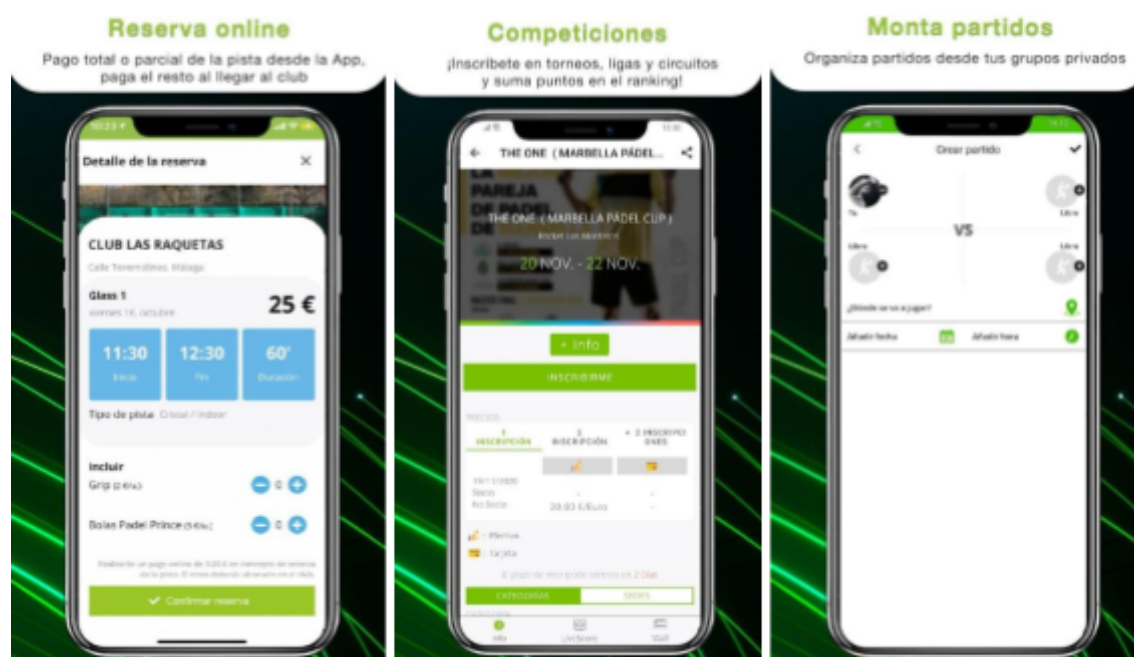


Figura 3.- Capturas 1, 2 y 3 Padel Manager

Capítulo 5

Restricciones

En este capítulo se expondrán las restricciones necesarias para el desarrollo del diseño, divididas a su vez en factores dato relacionados con la naturaleza del problema y factores estratégicos que representan las alternativas escogidas en el proceso de diseño.

5.1 Factores dato

Estos factores son inherentes al problema y no pueden ser modificados.

- La aplicación móvil debe ser gratuita ya que todos los clientes de los centros cívicos deben poder usarla.
- La aplicación móvil tiene que estar enfocada a la reserva de las clases de los centros cívicos.
- Las aplicaciones móvil y web requieren de conexión a internet y funcionamiento en tiempo real.
- Las aplicaciones móvil y web deben ser independientes del centro cívico en el que se implante.
- Las aplicaciones móvil y web deben recordar los datos de cada usuario si cambia el dispositivo.
- La aplicación web debe reconocer los diferentes tipos de usuarios: superadministrador, administrador y profesor.
- La aplicación web debe permitir realizar las mismas funciones a los administradores que la aplicación móvil permite a los clientes.

5.2 Factores estratégicos

Estos factores son variables de diseño que condicionarán el resultado final del proyecto.

5.2.1 Framework

Como framework [13] para el desarrollo de la aplicación móvil, se eligió Flutter ya que es el recomendado por los propios desarrolladores multiplataforma. Los frameworks de desarrollo móvil se dividen en nativos y multiplataforma, Android Studio es el más utilizado para Android y Xcode [14] para iOS, ambos son nativos. Existen frameworks multiplataforma que hacen uso de las API's de Android e iOS pudiendo así solo escribir una aplicación y que pueda ser ejecutada en ambos entornos, como es nuestro caso con Flutter.

Los frameworks multiplataforma aún siendo muy útiles, tienen varios problemas significativos, si ocurre alguna actualización en las API's de Android o iOS estos serán los últimos en integrarlas. Deben traducir todo lo codificado a los lenguajes que son interpretados por cada entorno, esto hace que disminuya la velocidad en comparación a los frameworks nativos.

En nuestro proyecto se ha escogido un framework multiplataforma por llegar al mayor número de usuarios posibles. Aún siendo menos veloz, la complejidad de la aplicación móvil es mínima, pues la mayoría de ventanas simplemente se conecta a la base de datos y muestra los resultados.

Como framework para el desarrollo de la aplicación web, se eligió Symfony ya que PHP es un lenguaje muy utilizado para el desarrollo de aplicaciones web y tiene integrado métodos de autenticación como JWT tokens.

5.2.2 Lenguaje

Flutter utiliza Dart como lenguaje de programación y Symfony usa PHP. PHP tiene Twig [15] como Java tiene jsp [16], lo que permite la creación de aplicaciones web con mucha facilidad, pues se renderizan en el servidor y podemos pasarle variables a los archivos HTML.

5.2.3 Servidor

El desarrollo del “Backend” [17] o parte de servidor implica reservar una parte del proyecto solo para eso. Como la aplicación web y la aplicación móvil ambas se conectan a la misma base de datos, un solo Backend suministra la lógica necesaria para el procesamiento de la información y su uso ambos Frontend.

Tener un Backend desarrollado a medida para el proyecto también implica el pago de este, que puede ser mediante el contrato de un servicio hosting o simplemente el costo de electricidad si decides instalar tu propio hardware.

Para este proyecto de momento simplemente se usa Docker [18] para hostear localmente el backend y la base de datos. El proceso de subirlo a un servidor sería pagar por el hosting y abrir Docker en el mismo. Como el proceso es el mismo en local que en la nube, no veo la necesidad de gastar el dinero en hostear en un servidor.

5.2.4 Modelo de datos

El modelo de datos escogido es SQL, esto quiere decir que se sigue el tradicional modelo relacional.

Se ha escogido este modelo debido a la seguridad e integridad de los datos, los objetos representados en nuestra base de datos no pueden cambiar de estructura y así evitar inserciones en la base de datos.

Capítulo 6

Recursos

6.1 Recursos humanos

- Dr Rafael Muñoz Salinas. Departamento de Informática y Análisis Numérico de la Universidad de Córdoba. Encargado de la supervisión como director del proyecto.
- Antonio Cabezas Jarbo. Alumno del Grado de Ingeniería Informática con mención en la especialidad de Software en la Universidad de Córdoba. Encargado del desarrollo del proyecto.

6.2 Recursos software

- Sistema Operativo: El sistema operativo será Ubuntu para el desarrollo y test de la aplicación web, un teléfono Android será también utilizado como sistema de test de la aplicación móvil.
- Entorno para la elaboración de la documentación: Word.
- Entorno de desarrollo: Android Studio, PHPStorm, PHPMyAdmin
- Lenguajes de Programación: PHP, Dart.

6.3 Recursos hardware

- Ordenador: Medion S15449, con las siguientes características:
 - Procesador: Intel Core i5-1135G7 (2,4 GHz)
 - Memoria RAM: 2x 8GB DDR4 Synchronous (3200 MHz).
 - Tarjeta gráfica: Tarjeta integrada.
 - Disco duro: 512 SSD.
- Dispositivo Android Realme 8 5G, con las siguientes características:
 - Pantalla: IPS 6,5 pulgadas (19,5:9)
 - Procesador: Dimensity 700 Octa Core
 - Ram: 4 GB
 - Almacenamiento: 64 GB

Capítulo 7

Especificación de requisitos

En este capítulo se desarrollarán los requisitos de la aplicación. Dichos requisitos se dividen en funcionales y no funcionales.

7.1 Requisitos funcionales

Los requisitos funcionales indican las funcionalidades que debe poder realizar la aplicación, en este documento se denotarán con el formato RF-X, siendo X el identificador de cada requisito.

RF-1. La aplicación móvil permitirá iniciar sesión y cerrar sesión.

RF-2. La aplicación móvil recordará al usuario cuando vuelva a iniciar la sesión.

RF-3. La aplicación móvil permitirá a los clientes hacer reservas.

RF-4. La aplicación móvil permitirá a los clientes visualizar todas sus reservas realizadas.

RF-5. La aplicación móvil permitirá a los clientes cancelar reservas que no hayan ocurrido aún.

RF-6. La aplicación web permitirá a los usuarios iniciar sesión y cerrar sesión.

RF-7. La aplicación web permitirá al superadministrador dar de alta a usuarios y asignarlos a cualquier centro cívico y dar de alta a clientes.

RF-8. La aplicación web permitirá al superadministrador cambiar cualquier aspecto de la configuración de la misma (nombre completo, nombre abreviado, logo, color principal y secundario, etc.).

RF-9. La aplicación web permitirá al superadministrador crear, editar, listar y eliminar áreas.

RF-10. La aplicación web permitirá al superadministrador crear, editar, listar y eliminar centros cívicos.

RF-11. La aplicación web permitirá a un superadministrador crear, editar, listar y eliminar habitaciones para cualquier centro cívico.

RF-12. La aplicación web permitirá a un superadministrador crear, editar, listar y eliminar clases para cualquier centro cívico y asignarlas a cualquier profesor del mismo centro.

RF-13. La aplicación web permitirá a un superadministrador crear, editar, listar y eliminar horarios de las clases de cualquier centro en las habitaciones del mismo centro cívico.

RF-14. La aplicación web permitirá a un superadministrador crear, editar, listar y eliminar reservas de cualquier cliente a cualquier horario de una clase de cualquier centro cívico.

RF-15. La aplicación web permitirá a un administrador dar de alta a usuarios para su propio centro cívico (otro administrador o un profesor) y dar de alta a clientes.

RF-16. La aplicación web permitirá a un administrador ver los datos de su centro cívico.

RF-17. La aplicación web permitirá a un administrador crear, editar, listar y eliminar habitaciones de su centro cívico.

RF-18. La aplicación web permitirá a un administrador y al superadministrador crear, editar, listar y eliminar clases del centro y asignarlas a cualquier profesor del centro cívico.

RF-19. La aplicación web permitirá a un administrador crear, editar, listar y eliminar horarios de las clases de su centro cívico en una habitación del mismo.

RF-20. La aplicación web permitirá a un administrador crear, editar, listar y eliminar reservas de cualquier cliente a cualquier horario de una clase del centro cívico.

RF-21. La aplicación web permitirá a un profesor crear nuevas clases, y editar y eliminar sus propias clases.

RF-22. La aplicación web permitirá a un profesor crear, editar y borrar horarios de una clase suya.

RF-23. La aplicación web permitirá a los usuarios ver y modificar su perfil.

7.2 Requisitos no funcionales

Los requisitos no funcionales hacen referencia a una serie de características del producto que no son servicios y que deben satisfacer los requerimientos de los usuarios.

Se denotarán con el formato RNF-X, siendo X el identificador de cada requisito.

RNF-1. Las aplicaciones serán utilizadas durante todo el año.

RNF-2. La aplicación móvil estará disponible en todos los dispositivos móviles y tablets.

RNF-3. La aplicación móvil estará disponible en los sistemas operativos Android e IOS.

RNF-4. Las aplicaciones deberán ofrecer un tiempo de respuesta con el servidor aceptable.

RNF-5. Las aplicaciones deberán ser eficientes y hacer el mínimo uso de la batería disponible.

RNF-6. Las aplicaciones deben ser fiables para asegurar que las comunicaciones sean estables.

RNF-7. Las aplicaciones dispondrán de una interfaz de usuario amigable y simple para que sea usable e intuitiva para el usuario.

Capítulo 8

Especificación del sistema

En este capítulo se desarrollará la aplicación del diseño a los requisitos especificados anteriormente, quedando definidos elementos como la interfaz de usuario, modelo de datos y cada una de las funciones de la app.

8.1 Modelo de datos relacional

El modelo de datos consta de varias tablas, cada tabla representa una entidad o una relación entre entidades en el problema, las tablas están compuestas por una o dos claves primarias que representan a cada objeto de dicha tabla, y se conectan entre sí a través de claves foráneas.

Además, cada tabla contiene un número de columnas que son los atributos correspondientes a la entidad o relación que representan.

En la Figura 7 se muestra este concepto de manera gráfica:

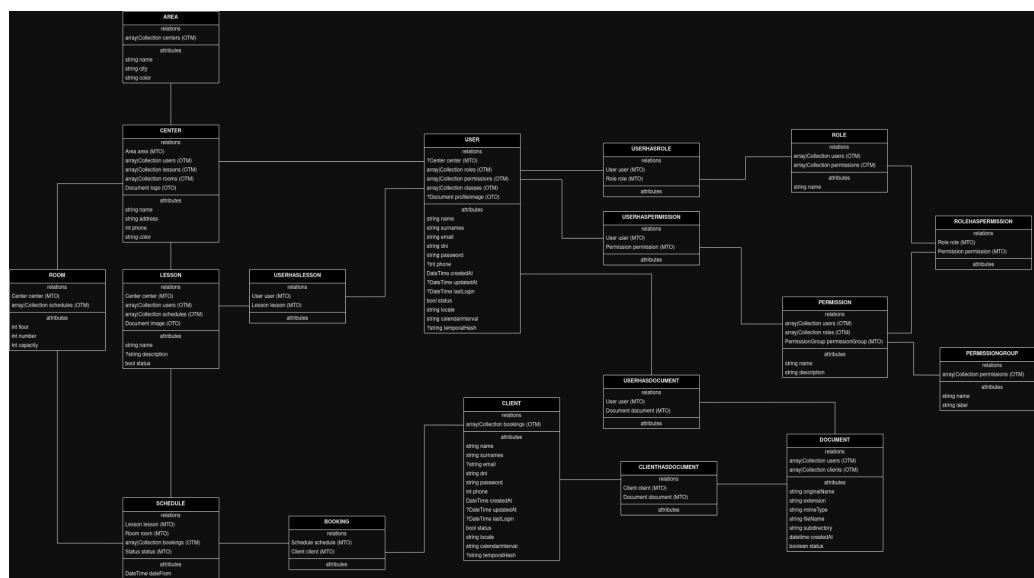


Figura 4.- Base de datos SQL en documentos

A continuación, se detallan las entidades y relaciones usadas para la creación de la base de datos.

8.1.1 Tabla Users

La tabla “Users” representa a la entidad usuario, se usará para guardar información sobre los usuarios que usan la aplicación.

Los atributos que la componen son:

- **id**

Cadena única que identifica a cada usuario.

- **center**

Clave foránea que se corresponde con el id del centro cívico al que pertenece el usuario. Si el usuario es el superadministrador, este campo está vacío.

- **roles**

Vector que contiene los roles del usuario. Solamente va a tener uno, pero por si en algún momento esto cambiaba decidí dejarlo así.

- **permissions**

Vector que contiene los permisos que tiene cada usuario.
Normalmente, cada usuario tendrá los permisos que se le asignan cuando se crea con un rol determinado (superadmin, admin, teacher). El superadministrador puede cambiar los permisos de cada usuario individualmente.

- **classes**

Vector que contiene los id de las clases que pertenecen a cada profesor. Si es un superadministrador o administrador el vector estará vacío.

- **profileImage**

Clave foránea que se corresponde con el id del documento que será la imagen de perfil del usuario. Puede estar vacío.

- **name**

Cadena que almacena el nombre del usuario.

- **surnames**

Cadena que almacena los apellidos del usuario.

- **email**

Cadena que almacena el correo electrónico del usuario. Será el username de los usuarios de la aplicación web

- **dni**

Cadena que almacena el dni del usuario.

- **password**

Cadena que almacena la contraseña del usuario. Esta cadena estará cifrada en la base de datos y no se descifra en ningún momento en el código. Para comparar se cifra la escrita por el usuario con el mismo algoritmo y se comparan cifradas.

- **phone**
Cadena que almacena el teléfono del usuario. Puede estar vacía.
- **createdAt**
Fecha de creación del usuario.
- **updatedAt**
Fecha de actualización del usuario. Comienza vacía y se actualiza cada vez que se modifica cualquier atributo del mismo.
- **lastLogin**
Fecha del último login del usuario. Comienza vacío.
- **status**
Booleano que nos dice el estado del usuario. (Dado de alta o dado de baja)
- **locale**
Cadena que almacena el idioma del dispositivo del usuario. Esta cadena se usará para las traducciones. Ahora mismo solo está en español o inglés.
- **temporalHash**
Cadena que almacena un hash temporal en caso de petición de cambio de contraseña. Este hash se comprobará al hacer click en el correo de cambio de contraseña y se borrará una vez cambiada.

8.1.2 Tabla Client

La tabla “Client” representa a la entidad cliente, se usará para guardar información sobre los usuarios de la aplicación móvil o “clientes”.

Los atributos que la componen son:

- **bookings**

Vector que contiene la lista de reservas del cliente.

El resto de atributos son los mismos que los del usuario a partir de **name**. La única diferencia con los usuarios es que el username de los clientes es el **dni** y el email puede dejarse vacío.

8.1.3 Tabla Area

La tabla “Area” representa a la entidad área, se usará para guardar información sobre las áreas geográficas.

Los atributos que la componen son:

- **id**
Cadena única que identifica a cada área.
- **centers**
Vector que contiene los centros que pertenecen a esta área. En caso de eliminar el área, todos los centros que pertenecen a esta también serán borrados.
- **name**
Cadena que representa el nombre del área.
- **city**
Cadena que representa la ciudad a la que pertenece el área.
- **color**
Cadena que representa el color asignado al área. Está en formato hexadecimal.

8.1.4 Tabla Center

La tabla “Center” representa a la entidad centro cívico, se usará para guardar información sobre los centros cívicos.

Los atributos que la componen son:

- **id**
Cadena única que identifica a cada centro cívico.
- **area**
Clave foránea que se corresponde con el id del área a la que pertenece el centro cívico.
- **users**
Vector que contiene la lista de usuarios que pertenecen al centro (administradores y profesores del mismo).
- **lessons**
Vector que contiene la lista de clases que se imparten en este centro. En caso de borrar el centro, se borrarían todas las clases del mismo.
- **rooms**
Vector que contiene la lista de habitaciones del centro donde se pueden impartir clases. En caso de borrar el centro, se borrarían todas las habitaciones del mismo.
- **logo**
Clave foránea que se corresponde con el id del documento que es el logo del centro cívico.
- **name**
Cadena que representa el nombre del área.

- **address**

Cadena que representa la dirección del centro cívico.

- **openingTime**

Fecha de apertura del centro cívico.

- **closingTime**

Fecha de cierre del centro cívico.

- **phone**

Cadena que representa el teléfono de contacto del centro cívico.

- **color**

Cadena que representa el color asignado al centro cívico. Está en formato hexadecimal.

8.1.5 Tabla Lesson

La tabla “Lesson” representa a la entidad clase, se usará para guardar información sobre las diferentes clases que se ofertan en el centro cívico.

Los atributos que la componen son:

- **id**
Cadena única que identifica a cada clase.
- **center**
Clave foránea que se corresponde con el id del centro cívico al que pertenece.
- **users**
Vector que contiene la lista de usuarios que imparten la clase. Puede haber más de un profesor por clase.
- **schedules**
Vector que contiene la lista de horarios en los que se imparte la clase. Si se borra la clase, se borrarán los horarios.
- **image**
Clave foránea que se corresponde con el id del documento que es la imagen de la clase.
- **name**
Cadena que representa el nombre de la clase.
- **duration**
Float que representa la duración de la clase. (1 – 1.5 – 2 horas)
- **description**
Cadena que representa la descripción de la clase. Puede ser nulo.
- **status**
Booleano que representa si una clase esta activa o no.
- **color**
Cadena que representa el color asignado a la clase en hexadecimal.

8.1.6 Tabla Room

La tabla “Room” representa a la entidad habitación, se usará para guardar información sobre las habitaciones del centro cívico.

Los atributos que la componen son:

- **id**
Cadena única que identifica a cada habitación.
- **center**
Clave foránea que se corresponde con el id del centro cívico al que pertenece.
- **schedules**
Vector que contiene una lista de los horarios que usan la habitación. Si se borra la habitación, se borrarán los horarios.
- **floor**
Entero que representa la planta en la que se encuentra la habitación.
- **number**
Entero que representa el número de la habitación.
- **capacity**
Entero que representa la capacidad máxima de alumnos o “clientes” por habitación.

8.1.7 Tabla Schedule

La tabla “Schedule” representa a la entidad horario, se usará para guardar información sobre los horarios de las clases.

Los atributos que la componen son:

- **id**
Cadena única que identifica a cada horario.
- **lesson**
Clave foránea que se corresponde con el id de la clase que se va a impartir.
- **status**
Clave foránea que se corresponde con el id del estado en el que se encuentra el horario.
- **room**
Clave foránea que se corresponde con el id de la habitación donde se va a impartir la clase durante este horario.
- **teacher**
Clave foránea que se corresponde con el id del profesor (User) que va a impartir la clase durante este horario.
- **Bookings**
Vector que contiene la lista de reservas que tiene este horario. Si se borra el horario, se borrarán las reservas sobre el mismo.
- **dateFrom**
Fecha y hora en la que comienza el horario.
- **dateTo**
Fecha y hora en la que acaba el horario.

8.1.8 Tabla Status

La tabla “Status” representa la entidad de los distintos estados posibles, se usará para guardar información sobre los estados posibles. En nuestro caso solamente se usará para los estados de los horarios, pero es escalable para que cualquier entidad que necesite más de dos estados y que no se pueda representar con un booleano.

Los atributos que la componen son:

- **id**
Entero único que identifica a cada estado.
- **schedules**
Vector que contiene a la lista de horarios a los que están asignados este estado.
- **name**
Cadena que representa el nombre del estado.
- **color**
Cadena que representa el color asignado al estado en hexadecimal.
- **entityType**
Cadena que representa la entidad a la que se puede asignar este estado.
- **statusOrder**
Entero que representa el orden en el que se creó el estado en la base de datos.

8.1.9 Tabla Booking

La tabla “Booking” representa la relación N:N entre los clientes y los horarios, se usará para guardar información sobre las reservas que hacen los clientes de los distintos horarios.

Los atributos que la componen son:

- **client**

Clave primaria y foránea que se corresponde con el id del cliente que hace la reserva.

- **schedule**

Clave primaria y foránea que se corresponde con el id del horario que se ha reservado.

8.1.10 Tabla Document

La tabla “Document” representa la entidad documento, se usará para guardar información sobre los documentos que hay en la aplicación web.

Los atributos que la componen son:

- **id**
Cadena única que identifica a cada documento.
- **clients**
Vector que contiene la lista de clientes que comparten el documento.
- **users**
Vector que contiene la lista de usuarios que comparten el documento.
- **originalName**
Cadena que representa el nombre original del documento cuando se subió.
- **extension**
Cadena que representa la extensión del archivo.
- **contentType**
Cadena que representa el tipo MIME del archivo.
- **fileName**
Cadena que representa el nombre único con el que se guarda el archivo en el servidor.
- **Subdirectory**
Cadena que representa el nombre de la carpeta donde se ha guardado el documento.
- **createdAt**
Fecha en la que se subió el archivo al servidor.
- **status**
Booleano que representa si el documento está activo o no.

8.1.11 Tabla UserHasDocument

La tabla “UserHasDocument” representa la relación N:N entre documentos y usuarios, se usará para guardar información sobre los documentos que tienen o usan los usuarios.

Los atributos que la componen son:

- **user**

Clave primaria y foránea que se corresponde con el id del usuario que tiene/usa el documento.

- **document**

Clave primaria y foránea que se corresponde con el id del documento.

8.1.12 Tabla ClientHasDocument

La tabla “ClientHasDocument” representa la relación N:N entre documentos y clientes, se usará para guardar información sobre los documentos que tienen o usan los clientes.

Los atributos que la componen son:

- **client**

Clave primaria y foránea que se corresponde con el id del cliente que tiene/usa el documento.

- **document**

Clave primaria y foránea que se corresponde con el id del documento.

8.1.13 Tabla Permission

La tabla “Permission” representa la entidad permiso, se usará para guardar información sobre los permisos posibles.

Los atributos que la componen son:

- **id**
Entero único que representa el permiso.
- **group**
Clave foránea que corresponde con el grupo al que pertenece el permiso.
- **permissionUsers**
Vector que contiene la lista de los conjuntos permiso-usuario.
- **permissionRoles**
Vector que contiene la lista de los conjuntos permiso-rol.
- **label**
Cadena que representa la etiqueta que se le asigna al permiso.
- **action**
Cadena que representa la acción del permiso.
- **description**
Cadena que representa la descripción del permiso. Puede ser nulo.
- **adminManaged**
Booleano que indica si el permiso es solamente para administradores.

8.1.14 Tabla PermissionGroup

La tabla "PermissionGroup" representa la entidad grupo de permisos, se usará para guardar información sobre los diferentes grupos de permisos.

Los atributos que la componen son:

- **id**
Entero único que representa el grupo de permisos.
- **name**
Cadena que representa el nombre del grupo de permisos.
- **label**
Cadena que representa la etiqueta que se le asigna al grupo de permisos.

8.1.15 Tabla Role

La tabla “Role” representa la entidad rol, se usará para guardar información sobre los roles posibles.

Los atributos que la componen son:

- **id**
Entero único que representa el rol.
- **users**
Vector que contiene la lista de los conjuntos rol-usuario.
- **permissions**
Vector que contiene la lista de los conjuntos permiso-rol.
- **name**
Cadena que representa el nombre del rol.
- **color**
Cadena que representa el color del rol en hexadecimal.
- **admin**
Booleano que representa si el rol es de administrador o no.
- **description**
Cadena que representa la descripción del permiso. Puede ser nulo.

8.1.16 Tabla RoleHasPermission

La tabla “RoleHasPermission” representa la relación N:N entre roles y permisos, se usará para guardar información sobre los permisos que tendrá cada rol.

Los atributos que la componen son:

- **role**

Clave primaria y foránea que se corresponde con el id del rol que tiene dicho permiso.

- **permission**

Clave primaria y foránea que se corresponde con el id del permiso que se le asigna al rol.

8.1.17 Tabla UserHasLesson

La tabla “UserHasLesson” representa la relación N:N entre usuarios y clases, se usará para guardar información sobre los usuarios (profesores) y las clases que imparten.

Los atributos que la componen son:

- **user**

Clave primaria y foránea que se corresponde con el id del usuario que imparte la clase.

- **lesson**

Clave primaria y foránea que se corresponde con el id de la clase que se le asigna al usuario.

8.1.18 Tabla UserHasPermission

La tabla “UserHasPermission” representa la relación N:N entre permisos y usuarios, se usará para guardar información sobre los permisos que tienen los usuarios.

Los atributos que la componen son:

- **user**

Clave primaria y foránea que se corresponde con el id del usuario que tiene el permiso.

- **permission**

Clave primaria y foránea que se corresponde con el id del permiso asignado al usuario.

8.1.19 Tabla UserHasRole

La tabla “UserHasRole” representa la relación N:N entre roles y usuarios, se usará para guardar información sobre los roles que tienen los usuarios.

Los atributos que la componen son:

- **user**

Clave primaria y foránea que se corresponde con el id del usuario que tiene el rol.

- **rol**

Clave primaria y foránea que se corresponde con el id del rol que se le asigna al usuario.

8.2 Estructura de la aplicación web

En este apartado se verá la estructura de clases y recursos de la aplicación web.

8.2.1 Infrastructure

Directorio que contiene:

- Archivos de configuración de nginx [19] y symfony.
- Dockerfile definiendo la versión de php y los componentes a instalar.
- El archivo .env [20] con las constantes del usuario mysql y las claves.
- Los archivos Docker-compose tanto para dev, que se usa en local para poder tener un tracking de los errores al implementar el código, como para prod, que es el que se usará en el servidor, así evitando que los usuarios finales vean en qué archivos se encuentran los errores en caso de que se implementase una nueva versión con errores o se encontrasen errores de la actual.
- Makefile para crear shortcuts y no tener que memorizar todos los comandos de Docker.

8.2.2 Backend

Directorio que contiene:

- El archivo .env con la configuración de la url de la base de datos y de las secret y public keys de JWT.
- El resto de archivos con las funcionalidades del backend completo.

8.2.3 Resources

Directorio que contiene todos los recursos de la aplicación web. Cada vez que se crea una clase que puede tener una imagen de perfil o representativa se guarda en un subdirectorio con el nombre de la clase a la que se refiere.

8.2.4 Migrations

Directorio que contiene todas las migraciones [21] que se han creado al crear la infraestructura de la base de datos con las entidades y sus relaciones.

8.2.5 Templates

Directorio que contiene el “Frontend” de la aplicación web. Al usar Twig, simplemente con pasarle variables con los datos directamente desde el backend podemos generar funcionalidades en la misma vista, lo que simplifica mucho el desarrollo del frontend.

8.2.6 Src

Directorio que contiene todos los recursos del backend de la aplicación web. Se dividen en subdirectorios:

- Entity

Contiene las entidades de la base de datos. Cada entidad o relación de entidades es un archivo y a su vez una tabla en la base de datos.

- Controller

Contiene los controladores de cada entidad. El controlador recibe las peticiones html y las redirecciona al servicio correspondiente.

- Service

Contiene los servicios de cada entidad. El servicio recibe una llamada desde el controlador y llama a los repositorios disponibles para recoger la información necesaria, la trata y envía la necesaria a la plantilla Twig que la muestra.

- Repository

Contiene las llamadas a la base de datos para recoger los datos que se le pide desde los servicios.

- Shared

Contiene las clases que serán usadas por cualquier otra clase con funcionalidades útiles para cualquier situación.

- Twig

Contiene las clases que definen funciones que se podrán usar en todas las plantillas Twig.

8.3 Estructura de la aplicación móvil

En este apartado se verá la estructura de clases y recursos de la aplicación móvil.

8.3.1 Android & IOS

Directorios generados automáticamente al crear el proyecto Flutter desde Android Studio. Se puede modificar versión de Android o IOS dentro de estas carpetas.

8.3.2 Lib

Directorio en el que se encuentran los archivos .dart que crean las vistas y contienen las funcionalidades de las mismas a través de widgets. El código de estas clases puede verse con más detalle en el documento “Manual de Código” del proyecto.

- main.dart

Archivo principal que contiene el widget root que se ejecuta al iniciar la aplicación. Este widget reenvía a la pantalla de login o la home dependiendo de si el usuario ha hecho login anteriormente y no cerró sesión antes de salir de la aplicación.

- api_connection.dart

Archivo que contiene la clase API. Esta clase contiene las cadenas con las direcciones de los controladores del backend que devuelve la información que necesita la app.

- base64_to_image.dart

Archivo que contiene el widget para construir una imagen a partir de la misma en base64 [22] (así se reciben desde el backend).

- login_screen.dart

Archivo que contiene el widget que crea la vista del inicio de sesión. Contiene la función para iniciar sesión y el link que la conecta con la vista de registro.

- signup_screen.dart

Archivo que contiene el widget que crea la vista del registro en la app. Contiene la función para registrarse y el link que la conecta con la vista de inicio de sesión. Se ha deshabilitado esta opción por motivos de seguridad, solamente presencialmente en cualquier centro cívico se podrá un cliente registrar entregando su DNI a un administrador para que lo haga desde la aplicación web.

- home_screen.dart

Archivo que contiene el widget que crea la vista del home. Esta vista muestra la lista de los centros cívicos disponibles. Si elegimos cualquiera de ellos entramos en la vista lesson_screen.dart. Si hacemos click en el icono de arriba a la izquierda accedemos en la vista client_profile_screen.dart, si lo hacemos en el icono de cerrar sesión volveremos a la vista login_screen.dart.

- client_profile_screen.dart

Archivo que contiene el widget que crea la vista del perfil de usuario. Aquí podemos ver la lista de los horarios reservados por el usuario y podremos cancelar cualquier reserva que no haya ocurrido aún.

- lessons_screen.dart

Archivo que contiene el widget que crea la vista de la lista de clases con horarios disponibles. Si elegimos cualquiera de ellas entramos en la vista schedules_sceen.dart. También contiene un botón de volver a la vista anterior.

- `schedules_screen.dart`

Archivo que contiene el widget que crea la vista de la lista de horarios disponibles. Los que ya tengamos reservados tendrán un icono distinto y no se podrán clicar. El resto se podrán reservar al hacer click.

8.3.3 Images

Directorio en el que se encuentran las imágenes que son exclusivas de la app móvil.

Las imágenes del directorio son:

- `login.jpg`

Imagen que se muestra en la vista de inicio de sesión.

- `register.png`

Imagen que se muestra en la vista de registro de usuario.

- `empty_image.png`

Imagen que se muestra cada vez que una vista de listas devuelve una lista vacía.

8.4 Diseño de la interfaz de usuario aplicación web

8.4.1 Pantalla de login

En esta pantalla el usuario podrá iniciar sesión en la aplicación introduciendo su email y contraseña.

El resultado se puede observar en la siguiente figura:

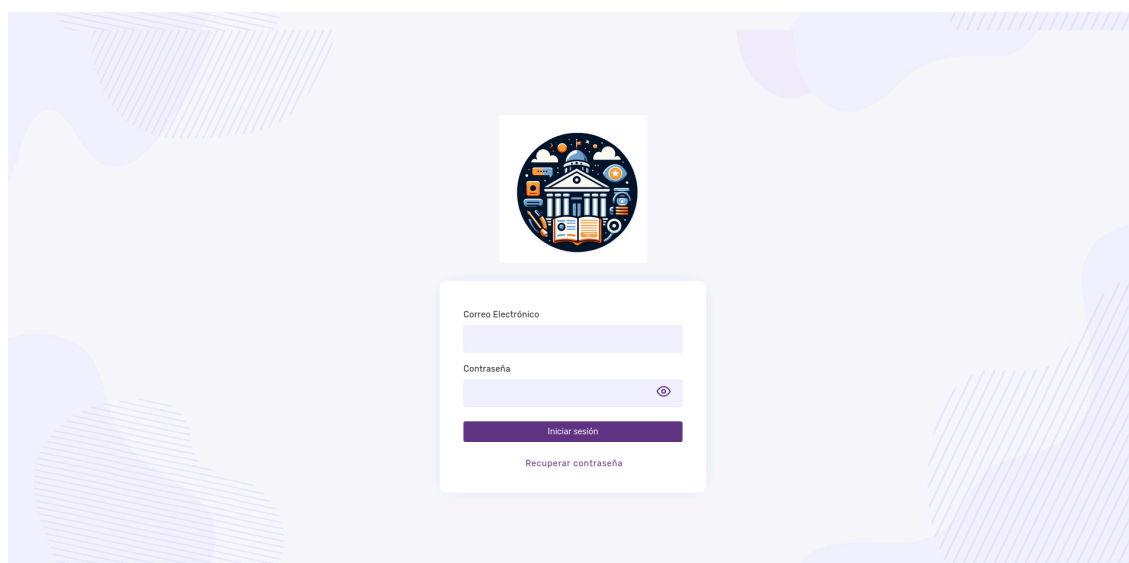


Figura 5.- Pantalla de Inicio de Sesión

8.4.2 Pantalla de horarios

En esta pantalla el usuario podrá ver los horarios de las diferentes clases disponibles. Por defecto se verá el calendario del mes, pero está la opción de la semana, del día y a modo de lista.

Existirá el botón de crear horario en la barra de arriba. La pantalla de creación mostrará las opciones disponibles.

Si se hace click en el horario, veremos la pantalla detallada del mismo. Esta contiene los botones de editar y borrar horario siempre y cuando no haya pasado la fecha de comienzo.

El resultado se puede observar en la siguiente figura:

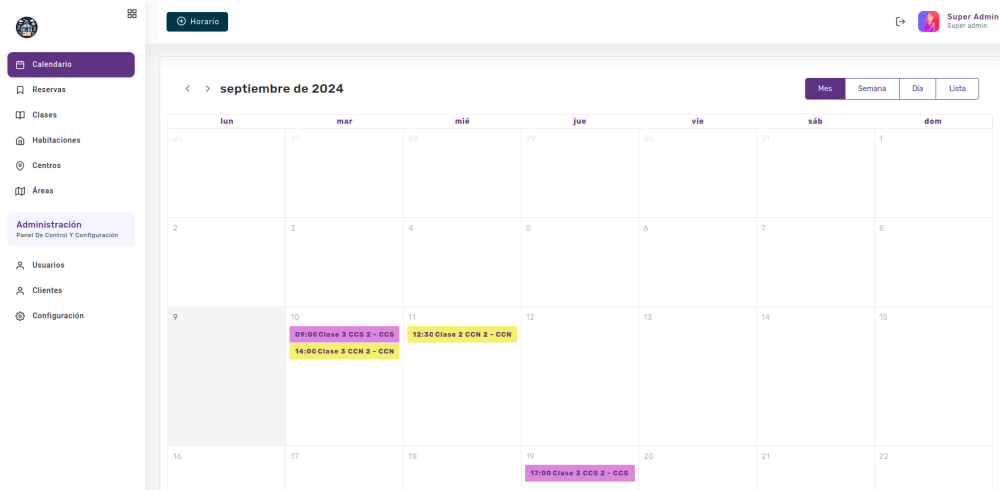


Figura 6.- Pantalla de horarios

8.4.3 Pantalla de clases

En esta pantalla el usuario podrá ver todas las clases que tenga permiso para ver. Un superadministrador verá todas las clases, un administrador verá todas las clases de su centro y un profesor verá todas sus clases.

Existirá el botón de crear clase en la barra de arriba. La pantalla de creación mostrará las opciones disponibles.

Existen tres botones para cada clase de la lista: ver, editar y dar de baja.

El resultado se puede observar en la siguiente figura:

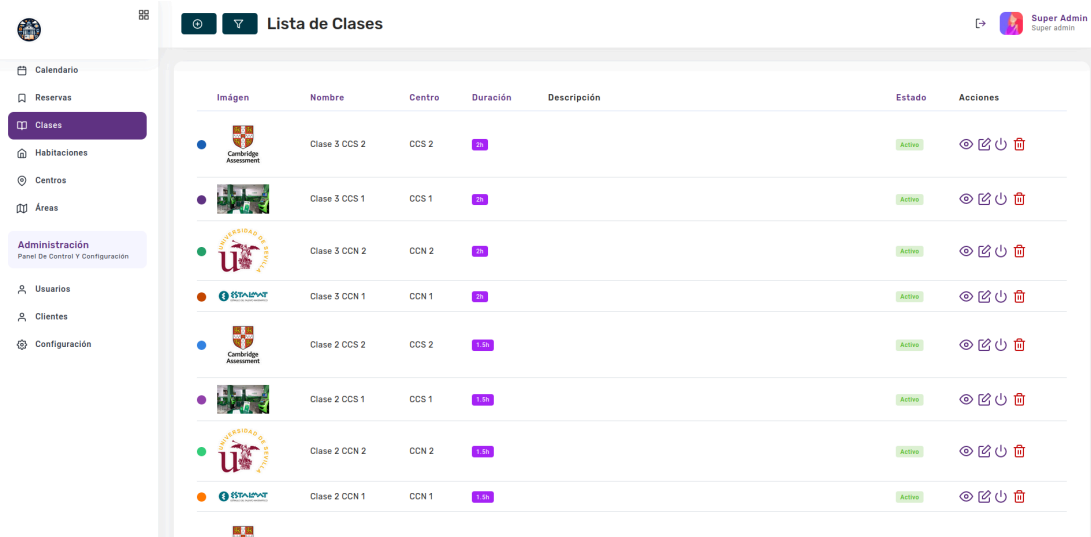


Figura 7.- Pantalla de clases

8.4.4 Pantalla de habitaciones

En esta pantalla el usuario podrá ver todas las habitaciones. Un superadministrador verá todas las habitaciones y un administrador verá todas las habitaciones de su centro.

Existirá el botón de crear habitación en la barra de arriba. La pantalla de creación mostrará las opciones disponibles.

Existen tres botones para cada habitación de la lista: ver, editar y borrar.

El resultado se puede observar en la siguiente figura:

Centro	Planta	Número	Capacidad	Acciones
CCS 2	Baja	# 2	10	
CCS 2	Baja	# 1	1	
CCS 1	Baja	# 1	1	
CCS 1	Baja	# 2	10	
CCN 2	Baja	# 1	1	
CCN 2	Baja	# 2	10	
CCN 1	1*	# 1	1	
CCN 1	Baja	# 1	1	
CCN 1	Baja	# 2	10	

Resultados: 9

Mostrar: 25

Figura 8.- Pantalla de habitaciones

8.4.5 Pantalla de áreas

Solamente el superadministrador podrá ver esta pantalla.

Existirá el botón de crear área en la barra de arriba. La pantalla de creación mostrará las opciones disponibles.

Existen tres botones para cada área de la lista: ver, editar y borrar.

El resultado se puede observar en la siguiente figura:

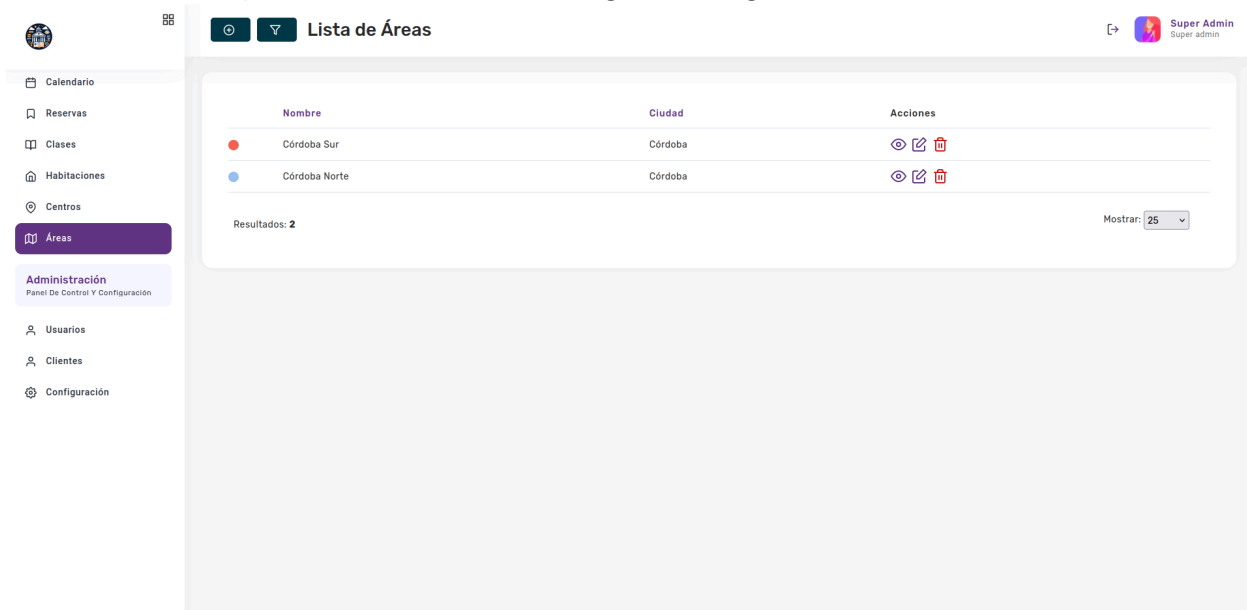


Figura 9.- Pantalla de áreas

8.4.6 Pantalla de centros

Esta pantalla se verá distinta para cada usuario. El superadministrador podrá ver una lista de todos los centros y los administradores y profesores solamente verán los datos del centro al que están asignados.

Para el superadministrador existirá un botón para crear un centro en la barra de arriba, cuya pantalla de creación mostrará las opciones disponibles.

Existen tres botones para cada centro para el superadministrador de la lista: ver, editar y dar de baja.

El resultado se puede observar en la siguiente figura:

Logo	Nombre	Dirección	Área	Teléfono	Hora de Apertura	Hora de Cierre	Acciones
	CCS 2	Calle CSS2	Córdoba Sur	+34 957 444 444	09:00	21:30	
	CCS 1	Calle CCS1	Córdoba Sur	+34 957 333 333	09:00	21:00	
	CCN 2	Calle CCN2	Córdoba Norte	+34 957 222 222	08:00	20:30	
	CCN 1	Calle CCN1	Córdoba Norte	+34 957 111 111	08:00	20:00	

Resultados: 4

Mostrar: 25

Figura 10.- Pantalla de centros

8.4.7 Pantalla de usuarios

Esta pantalla mostrará una lista de los usuarios de la aplicación web. El superadministrador podrá ver todos los usuarios y los administradores podrán ver todos los usuarios de su centro.

Existirá el botón de registro de usuario en la barra de arriba. La pantalla de registro mostrará las opciones disponibles. Los administradores solo podrán registrar usuarios para su centro.

Existen tres botones para cada usuario de la lista: ver, editar y dar de baja.

El resultado se puede observar en la siguiente figura:

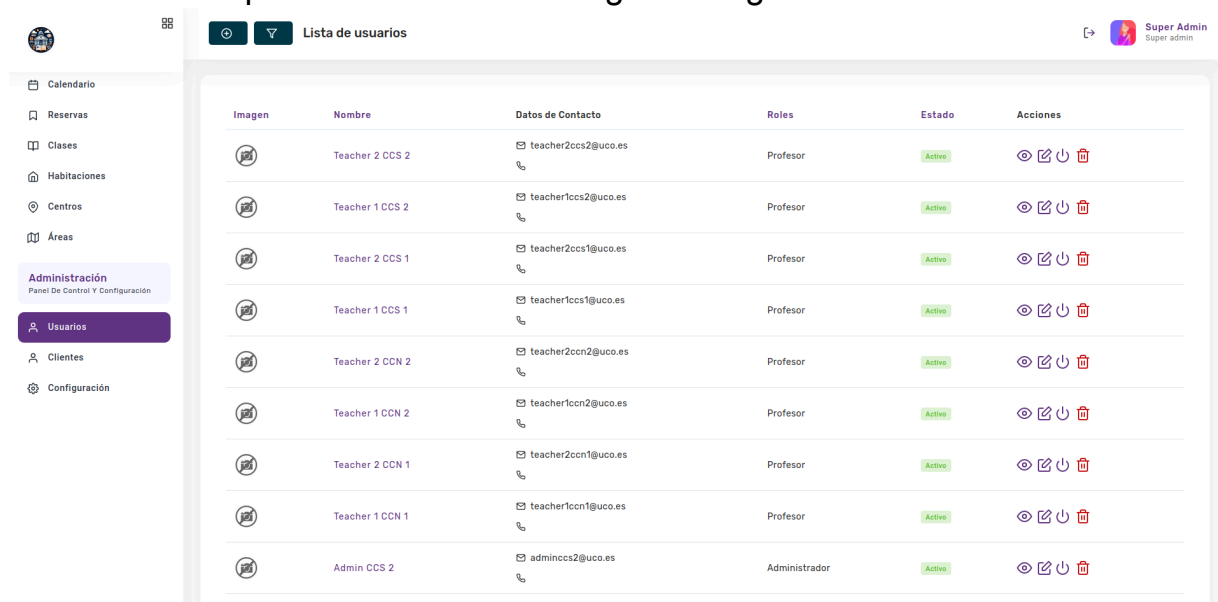






































Imagen	Nombre	Datos de Contacto	Roles	Estado	Acciones
	Teacher 2 CCS 2	teacher2ccs2@uco.es	Profesor	Activo	  
	Teacher 1 CCS 2	teacher1ccs2@uco.es	Profesor	Activo	  
	Teacher 2 CCS 1	teacher2ccs1@uco.es	Profesor	Activo	  
	Teacher 1 CCS 1	teacher1ccs1@uco.es	Profesor	Activo	  
	Teacher 2 CCN 2	teacher2ccn2@uco.es	Profesor	Activo	  
	Teacher 1 CCN 2	teacher1ccn2@uco.es	Profesor	Activo	  
	Teacher 2 CCN 1	teacher2ccn1@uco.es	Profesor	Activo	  
	Teacher 1 CCN 1	teacher1ccn1@uco.es	Profesor	Activo	  
	Admin CCS 2	adminccs2@uco.es	Administrador	Activo	  

Figura 11.- Pantalla de usuarios







8.4.8 Pantalla de clientes

Esta pantalla la podrán ver tanto el superadministrador como los administradores. Se trata de una lista de los clientes de aplicación móvil.

Existirá el botón de registrar cliente en la barra de arriba. La pantalla de registro mostrará las opciones disponibles.

Existen tres botones para cada cliente de la lista: ver, editar y dar de baja.

El resultado se puede observar en la siguiente figura:

Nombre	Datos de Contacto	Estado	Acciones
Cliente Segundo	653 118 018	Activo	  
Cliente Primero	653 118 019	Activo	  

Resultados: 2

Mostrar: 25

Figura 12.- Pantalla de clientes

8.4.9 Pantalla de configuración

Solamente el superadministrador podrá ver esta pantalla.

La pantalla de configuración mostrará las opciones disponibles. Podemos cambiar tanto los colores, el nombre de la aplicación, el logo, etc.

El resultado se puede observar en la siguiente figura:

The screenshot displays the 'Configuración' (Configuration) page of a web application. On the left is a sidebar menu with options: Calendario, Reservas, Clases, Habitaciones, Centros, Áreas, Administración (highlighted), Usuarios, Clientes, and Configuración. The main content area is titled 'Panel De Control Y Configuración' and contains the following settings:

- Nombre de la Aplicación:** Reserva Cultura
- Nombre Corto de la Aplicación:** CulturReserva
- Logotipo Grande:** Browse... No file selected.
- Logotipo Pequeño:** Browse... No file selected.
- Favicon:** Browse... No file selected.
- Color Principal:** A purple color picker bar.
- Color Secundario:** A yellow color picker bar.
- Días para Notificar:** 1
- Intervalo en el Calendario:** 12:00 AM

An 'Actualizar' (Update) button is located at the bottom right of the configuration panel. The top right corner shows the user 'Super Admin' with a profile icon.

Figura 13.- Pantalla de configuración

8.4.10 Pantalla de perfil de usuario

Cada usuario podrá ver su perfil. Todos los usuarios podrán cambiar su contraseña en esta pantalla.

El superadministrador y los administradores pueden ver sus datos y los profesores podrán ver una lista de sus clases.

El resultado se puede observar en la siguiente figura:

The screenshot shows a web application interface for a user profile. On the left is a sidebar menu with icons and labels: Calendario, Reservas, Clases, Habitaciones, Centros, Áreas, Administración (Panel De Control Y Configuración), Usuarios (highlighted in purple), Clientes, and Configuración. The main content area is titled 'Perfil' with a back arrow. It features a profile card for 'Super Admin' with a circular avatar, a grid of personal details (Nombre: Super, Apellidos: Admin, Teléfono, Correo Electrónico: i92cajaa@uco.es, Rol: Super admin), and a section for updating the password. The password update section includes labels 'Contraseña *' and 'Repetir Contraseña *', corresponding input fields, and an 'Actualizar' button. The top right corner shows a user profile icon and the text 'Super Admin Super admin'.

Figura 14.- Pantalla de perfil

8.5 Diseño de la interfaz de usuario aplicación móvil

Este apartado desarrollará el proceso de creación de la interfaz de la app, se han diseñado dos temas, claro y oscuro.

8.5.1 Pantalla de carga

Esta pantalla aparecerá cada vez que iniciemos la aplicación, se mostrará mientras se carga la aplicación. Cuando la carga termine si todo ha salido bien se redirigirá al usuario a su correspondiente interfaz.

Se incluye una barra de carga para que el usuario sepa que la app está en funcionamiento y no se ha detenido. Se muestra el resultado en la siguiente figura:

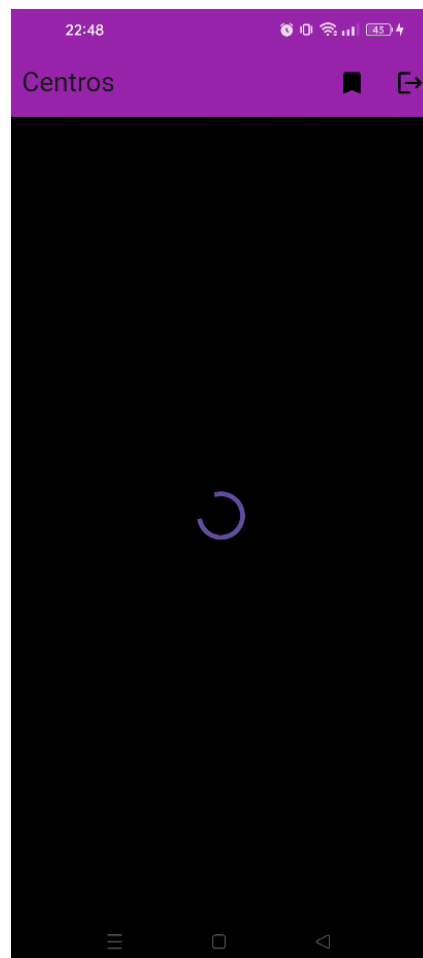


Figura 15.- Pantalla de carga ReservaCultura.

8.5.2 Pantalla de login

En esta pantalla el usuario podrá iniciar sesión en la aplicación introduciendo su DNI y contraseña.

Aparecerá un icono en el campo de contraseña para que el usuario pueda observar si está bien escrita

Existirá un único botón, iniciar sesión.

El resultado se puede observar en la siguiente figura:

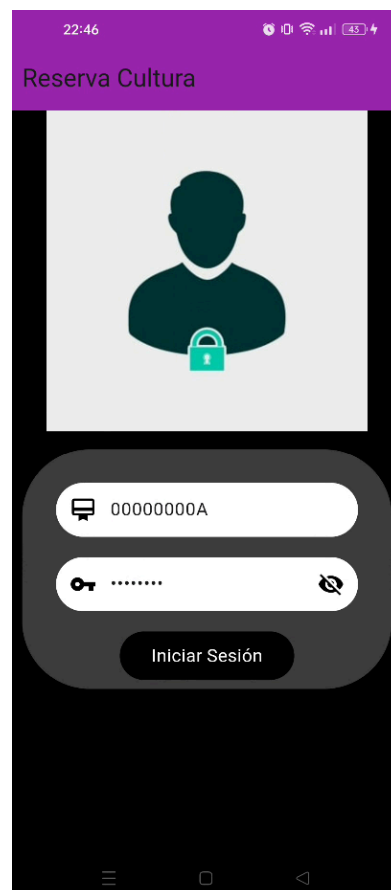


Figura 16.- Pantalla de login ReservaCultura.

8.5.3 Pantalla de centros

Esta pantalla sólo aparecerá cuando el usuario haya iniciado sesión en la parte superior habrá un texto explicativo y dos botones, el de perfil y el de cerrar sesión.

En la pantalla habrá una lista de todos los centros que tienen horarios activos que no hayan ocurrido aún.

En la siguiente figura se muestra el resultado final.



Figura 17.- Pantalla de bienvenida ReservaCultura.

8.5.4 Pantalla de clases

Esta es la pantalla principal de cada centro. En esta pantalla se verá una lista de todas las clases que tengan horarios activos que no hayan ocurrido aún.

El resultado final se puede observar en la siguiente figura:

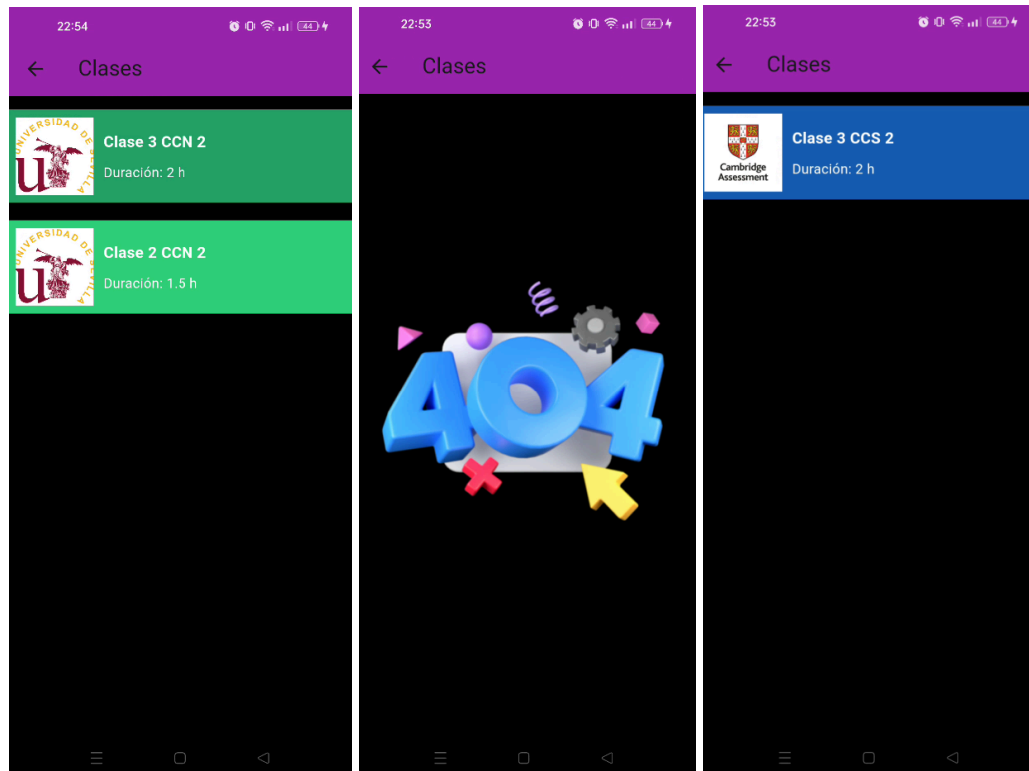


Figura 18.- Pantallas de clases ReservaCultura.

8.5.5 Pantalla de horarios

En esta pantalla deslizable se verán listados todos los horarios de la clase seleccionada en la pantalla anterior.

Se podrán reservar los horarios que no se hayan reservado previamente y solo se mostrarán los que no estén completos o ya hayan empezado.

En la siguiente figura se puede observar el resultado final de esta pantalla:

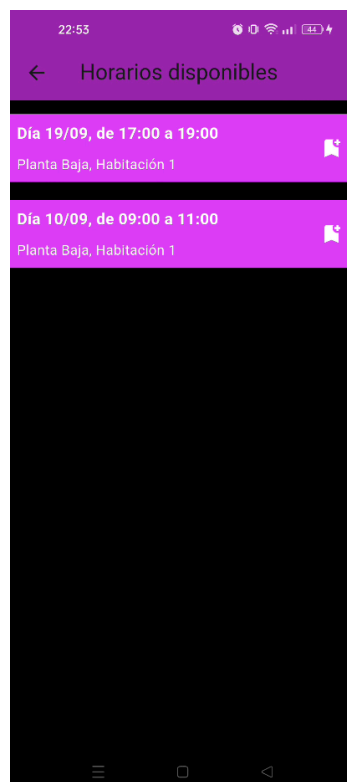


Figura 19.- Pantalla de horarios.

8.5.6 Pantalla perfil cliente

En esta pantalla se verán los datos del cliente.

En la pantalla habrá una lista de todos los horarios que el cliente ha reservado. Se podrán cancelar todos los horarios que no hayan comenzado aún.

En la siguiente figura se muestra el resultado final.



Figura 20.- Pantalla perfil cliente

8.6 Desarrollo de las funciones de la app web

En este apartado se mostrará cómo se han desarrollado las diferentes funciones con las que cuenta la aplicación móvil y que se ajustan a la especificación de requisitos.

Cada función se describirá mediante casos de uso con lenguaje de alto nivel.

8.6.1 Sistema de autenticación dedicado

La app permitirá iniciar sesión y cerrar sesión.

El sistema dedicado utiliza usuario y contraseña, en ReservaCultura el usuario equivale al correo electrónico. En las siguientes tablas pueden verse los casos de uso:

Nombre	Inicio de sesión
Descripción	El usuario introduce los datos necesarios para iniciar sesión en la aplicación web.
Actores	Usuario
Precondiciones	El usuario debe estar en la pantalla de login.
Flujo	<ol style="list-style-type: none">1. El usuario introduce los datos en los campos del formulario.2. El usuario pulsa el botón de iniciar sesión.<ol style="list-style-type: none">a. Si el usuario cometió algún error al introducir los datos se muestra en la pantalla.b. El usuario inicia sesión y se redirecciona a la pantalla objetivo.
Postcondiciones	El usuario inicia sesión y se encuentra en la pantalla de calendario de horarios.

Tabla 2.- Caso de uso, Inicio de sesión en aplicación web

Nombre	Cerrar sesión
Descripción	El usuario pulsa el botón de cerrar la sesión actual.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de cerrar sesión. 2. Se limpian los datos de sesión. 3. El usuario es redirigido a la pantalla de login.
Postcondiciones	El usuario no está logueado y se encuentra en la pantalla de login.

Tabla 3.- Caso de uso, Cerrado de sesión en la aplicación web

8.6.2 Sistema de áreas

La app permitirá listar, crear, editar y borrar áreas.

En las siguientes tablas pueden verse los casos de uso:

Nombre	Listar áreas
Descripción	La aplicación web lista todas las áreas existentes.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y ser un superadministrador.
Flujo	<ol style="list-style-type: none">1. El usuario pulsa el botón de áreas en el menú lateral.2. El usuario es redirigido a la pantalla de áreas.
Postcondiciones	El usuario se encuentra en la pantalla de áreas.

Tabla 4.- Caso de uso, Listar áreas

Nombre	Crear área
Descripción	Se crea un área.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y ser superadministrador.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de crear área. 2. El usuario es redirigido a la pantalla de creación de área. 3. El usuario completa todos los campos necesarios. <ol style="list-style-type: none"> a) Si hay conflictos con los datos salta error b) Si no hay conflictos, se crea el área
Postcondiciones	Se crea un área y se redirige al usuario a la pantalla de listar áreas.

Tabla 5.- Caso de uso, Creación de área

Nombre	Editar área
Descripción	Se edita el área seleccionada.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y ser un superadministrador.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de editar área. 2. El usuario es redirigido a la pantalla de edición de área. 3. El usuario completa todos los campos necesarios.

	<ul style="list-style-type: none"> • Si hay conflictos con los datos salta error • Si no hay conflictos, se edita el área
Postcondiciones	El área es editada y se redirige al usuario a la pantalla de listado de áreas.

Tabla 6.- Caso de uso, Editar área

Nombre	Borrar área
Descripción	Se borra el área seleccionada.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y ser un superadministrador.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de borrar área. 2. El usuario confirma su elección de borrar el área. 3. Se borra el área
Postcondiciones	El área y sus correspondientes centros son borrados y se redirige al usuario a la pantalla de listado de áreas.

Tabla 7.- Caso de uso, Borrar área

8.6.3 Sistema de centros

La app permitirá listar, crear, editar y borrar centros.

En las siguientes tablas pueden verse los casos de uso:

Nombre	Listar centros
Descripción	La aplicación web lista todas las áreas existentes.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y ser un superadministrador.
Flujo	<ol style="list-style-type: none">1. El usuario pulsa el botón de centros en el menú lateral.2. El usuario es redirigido a la pantalla de centros.
Postcondiciones	El usuario se encuentra en la pantalla de centros.

Tabla 8.- Caso de uso, Listar centros

Nombre	Crear centro
Descripción	Se crea un centro.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y ser superadministrador.

Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de crear centro. 2. El usuario es redirigido a la pantalla de creación de centro. 3. El usuario completa todos los campos necesarios. <ol style="list-style-type: none"> a. Si hay conflictos con los datos salta error b. Si no hay conflictos, se crea el centro
Postcondiciones	Se crea un centro y se redirige al usuario a la pantalla de listar centros.

Tabla 9.- Caso de uso, Creación de centro

Nombre	Editar centro
Descripción	Se edita el centro seleccionado.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y ser un superadministrador.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de editar centro. 2. El usuario es redirigido a la pantalla de edición de centro. 3. El usuario completa todos los campos necesarios. <ol style="list-style-type: none"> a. Si hay conflictos con los datos salta error b. Si no hay conflictos, se edita el centro
Postcondiciones	El centro es editado y se redirige al usuario a la pantalla de listado de centros.

Tabla 10: Editar centro

Nombre	Borrar centro
Descripción	Se borra el centro seleccionado.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y ser un superadministrador.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de borrar centro. 2. El usuario confirma su elección de borrar el centro. 3. Se borra el centro
Postcondiciones	El centro es borrado y se redirige al usuario a la pantalla de listado de centros.

Tabla 11.- Borrar centro

8.6.4 Sistema de usuarios

La app permitirá listar, registrar, editar y dar de baja a usuarios.

En las siguientes tablas pueden verse los casos de uso:

Nombre	Listar usuarios
Descripción	La aplicación web lista todos los usuarios existentes.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y ser un superadministrador o administrador.
Flujo	<ol style="list-style-type: none">1. El usuario pulsa el botón de usuarios en el menú lateral.2. El usuario es redirigido a la pantalla de usuarios.
Postcondiciones	El usuario se encuentra en la pantalla de usuarios.

Tabla 12: Caso de uso, Listar usuarios

Nombre	Registrar usuario
Descripción	Se registra un usuario.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y ser superadministrador o administrador.

Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de registrar usuario. 2. El usuario es redirigido a la pantalla de registro de usuario. 3. El usuario completa todos los campos necesarios. <ol style="list-style-type: none"> a. Si hay conflictos con los datos salta error b. Si no hay conflictos, se crea el usuario
Postcondiciones	Se registra un usuario y se redirige al usuario creador a la pantalla de listar usuarios.

Tabla 13.- Caso de uso, Registro de usuario

Nombre	Editar usuario
Descripción	Se edita el usuario seleccionado.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y ser un superadministrador o administrador.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de editar usuario. 2. El usuario es redirigido a la pantalla de edición de usuario. 3. El usuario completa todos los campos necesarios. <ol style="list-style-type: none"> a. Si hay conflictos con los datos salta error b. Si no hay conflictos, se edita el cliente
Postcondiciones	El usuario es editado y se redirige al usuario que edita a la pantalla de listado de usuarios.

Tabla 14.- Caso de uso, Editar usuario

Nombre	Dar de baja/alta a usuario
Descripción	Se da de baja/alta al usuario seleccionado.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada, ser un superadministrador o administrador y debe estar activado/desactivado.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de dar de baja/alta a usuario. 2. El usuario confirma su elección de dar de baja/alta al usuario. 3. Se da de baja/alta al usuario
Postcondiciones	El usuario es dado de baja/alta y se redirige al usuario que da de baja a la pantalla de listado de usuarios.

Tabla 15.- Caso de uso, Dar de baja/alta a un usuario

8.6.5 Sistema de clientes

La app permitirá listar, registrar, editar y dar de baja a clientes.

En las siguientes tablas pueden verse los casos de uso:

Nombre	Listar clientes
Descripción	La aplicación web lista todos los clientes existentes.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y ser un superadministrador o administrador.
Flujo	<ol style="list-style-type: none">1. El usuario pulsa el botón de clientes en el menú lateral.2. El usuario es redirigido a la pantalla de clientes.
Postcondiciones	El usuario se encuentra en la pantalla de usuarios.

Tabla 16.- Caso de uso, Listar clientes

Nombre	Registrar cliente
Descripción	Se registra un cliente.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y ser superadministrador o administrador.

Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de registrar cliente. 2. El usuario es redirigido a la pantalla de registro de cliente. 3. El usuario completa todos los campos necesarios. <ol style="list-style-type: none"> a. Si hay conflictos con los datos salta error b. Si no hay conflictos, se crea el cliente
Postcondiciones	Se registra un cliente y se redirige al usuario a la pantalla de listar usuarios.

Tabla 17.- Caso de uso, Registro de cliente

Nombre	Editar cliente
Descripción	Se edita el cliente seleccionado.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y ser un superadministrador o administrador.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de editar cliente. 2. El usuario es redirigido a la pantalla de edición de cliente. 3. El usuario completa todos los campos necesarios. <ol style="list-style-type: none"> a. Si hay conflictos con los datos salta error b. Si no hay conflictos, se edita el cliente
Postcondiciones	El cliente es editado y se redirige al usuario a la pantalla de listado de clientes.

Tabla 18.- Caso de uso, Editar cliente

Nombre	Dar de baja/alta a cliente
Descripción	Se da de baja/alta al cliente seleccionado.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada, ser un superadministrador o administrador y el cliente estar activado/desactivado.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de dar de baja/alta cliente. 2. El usuario confirma su elección de dar de baja/alta al cliente. 3. Se da de baja/alta al cliente
Postcondiciones	El cliente es dado de baja/alta y se redirige al usuario a la pantalla de listado de usuarios.

Tabla 19.- Caso de uso, Dar de baja/alta a un cliente

8.6.6 Sistema de configuración

La app permitirá editar la configuración.

En las siguientes tablas pueden verse los casos de uso:

Nombre	Editar configuración
Descripción	La aplicación edita la configuración de la misma.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y ser un superadministrador.
Flujo	<ol style="list-style-type: none">1. El usuario pulsa el botón de configuración en el menú lateral.2. El usuario es redirigido a la pantalla de configuración.3. El usuario cambia la configuración y se aplican los cambios.
Postcondiciones	El usuario se encuentra en la pantalla de horarios.

Tabla 20.- Caso de uso, Editar configuración

8.6.7 Sistema de clases

La app permitirá listar, crear, editar y dar de baja/alta clases.

En las siguientes tablas pueden verse los casos de uso:

Nombre	Listar clases
Descripción	La aplicación web lista todas las clases existentes.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada.
Flujo	<ol style="list-style-type: none">4. El usuario pulsa el botón de clases en el menú lateral.5. El usuario es redirigido a la pantalla de clases.
Postcondiciones	El usuario se encuentra en la pantalla de clases.

Tabla 21.- Caso de uso, Listar clases

Nombre	Crear clase
Descripción	Se crea una clase.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada.

Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de crear clase. 2. El usuario es redirigido a la pantalla de creación de clase. 3. El usuario completa todos los campos necesarios. <ol style="list-style-type: none"> a. Si hay conflictos con los datos salta error b. Si no hay conflictos, se crea la clase
Postcondiciones	Se crea una clase y se redirige al usuario a la pantalla de listar clases.

Tabla 22.- Caso de uso, Creación de clase

Nombre	Editar clase
Descripción	Se edita la clase seleccionada.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y existir la clase.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de editar clase. 2. El usuario es redirigido a la pantalla de edición de clase. 3. El usuario completa todos los campos necesarios. <ol style="list-style-type: none"> a. Si hay conflictos con los datos salta error b. Si no hay conflictos, se edita la clase
Postcondiciones	La clase es editada y se redirige al usuario a la pantalla de listado de clases.

Tabla 23.- Caso de uso, Editar clase

Nombre	Dar de baja/alta clase
Descripción	Se da de baja la clase seleccionada.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y la clase estar activada/desactivada.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de dar de baja/alta clase. 2. El usuario confirma su elección de dar de baja/alta a la clase. 3. Se da de baja/alta a la clase
Postcondiciones	La clase es dada de baja/alta y se redirige al usuario a la pantalla de listado de clases.

Tabla 24.- Caso de uso, Dar de baja/alta una clase

8.6.8 Sistema de habitaciones

La app permitirá listar, crear, editar y dar de baja/alta habitaciones.

En las siguientes tablas pueden verse los casos de uso:

Nombre	Listar habitaciones
Descripción	La aplicación web lista todas las habitaciones existentes.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada.
Flujo	<ol style="list-style-type: none">1. El usuario pulsa el botón de habitaciones en el menú lateral.2. El usuario es redirigido a la pantalla de habitaciones.
Postcondiciones	El usuario se encuentra en la pantalla de habitaciones.

Tabla 25.- Caso de uso, Listar habitaciones

Nombre	Crear habitación
Descripción	Se crea una habitación.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada.

Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de crear habitación. 2. El usuario es redirigido a la pantalla de creación de habitación. 3. El usuario completa todos los campos necesarios. <ol style="list-style-type: none"> a. Si hay conflictos con los datos salta error b. Si no hay conflictos, se crea la habitación
Postcondiciones	Se crea una habitación y se redirige al usuario a la pantalla de listar habitaciones.

Tabla 26.- Caso de uso, Creación de habitación

Nombre	Editar habitación
Descripción	Se edita la habitación seleccionada.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y existir la habitación.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de editar habitación. 2. El usuario es redirigido a la pantalla de edición de habitación. 3. El usuario completa todos los campos necesarios. <ol style="list-style-type: none"> a. Si hay conflictos con los datos salta error b. Si no hay conflictos, se edita la habitación
Postcondiciones	La habitación es editada y se redirige al usuario a la pantalla de listado de habitaciones.

Tabla 27.- Caso de uso, Editar habitación

Nombre	Dar de baja/alta habitación
Descripción	Se da de baja/alta la habitación seleccionada.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y la habitación estar activada/desactivada.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de dar de baja/alta habitación. 2. El usuario confirma su elección de dar de baja/alta a la habitación. 3. Se da de baja/alta a la habitación
Postcondiciones	La habitación es dada de baja/alta y se redirige al usuario a la pantalla de listado de habitaciones.

Tabla 28.- Caso de uso, Dar de baja/alta una habitación

8.6.9 Sistema de horarios

La app permitirá listar, crear, editar y borrar horarios.

En las siguientes tablas pueden verse los casos de uso:

Nombre	Listar horarios
Descripción	La aplicación web lista todos los horarios existentes.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada.
Flujo	<ol style="list-style-type: none">1. El usuario pulsa el botón de calendario en el menú lateral.2. El usuario es redirigido a la pantalla de horarios.
Postcondiciones	El usuario se encuentra en la pantalla de horarios.

Tabla 29.- Caso de uso, Listar horarios

Nombre	Crear horario
Descripción	Se crea una horario.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada.

Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de crear horario. 2. El usuario es redirigido a la pantalla de creación de horario. 3. El usuario completa todos los campos necesarios. <ol style="list-style-type: none"> a. Si hay conflictos con los datos salta error b. Si no hay conflictos, se crea el horario
Postcondiciones	Se crea un horario y se redirige al usuario a la pantalla de listar horarios.

Tabla 30.- Caso de uso, Creación de horario

Nombre	Editar horario
Descripción	Se edita el horario seleccionado.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada, existir el horario y que no haya comenzado.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de editar horario. 2. El usuario es redirigido a la pantalla de edición de horario. 3. El usuario completa todos los campos necesarios. <ol style="list-style-type: none"> a. Si hay conflictos con los datos salta error b. Si no hay conflictos, se edita el horario
Postcondiciones	El horario es editado y se redirige al usuario a la pantalla de horarios.

Tabla 31.- Caso de uso, Editar horario

Nombre	Borrar horario
Descripción	Se borra el horario seleccionado.
Actores	Usuario
Precondiciones	El usuario debe estar con la sesión iniciada y el horario debe existir y no haber comenzado.
Flujo	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de borrar horario. 2. El usuario confirma su borrar el horario. 3. Se borra el horario.
Postcondiciones	El horario se borra y se redirige al usuario a la pantalla de horarios.

Tabla 32.- Caso de uso, Borrar un horario

8.7 Desarrollo de las funciones de la app móvil

En este apartado se mostrará cómo se han desarrollado las diferentes funciones con las que cuenta la aplicación móvil y que se ajustan a la especificación de requisitos.

Cada función se describirá mediante casos de uso con lenguaje de alto nivel.

8.7.1 Sistema de autenticación dedicado

La app permitirá iniciar sesión y cerrar sesión de cliente.

El sistema dedicado utiliza usuario y contraseña, en ReservaCultura el usuario equivale al DNI del cliente al registrarse presencialmente en un centro cívico. En las siguientes tablas pueden verse los casos de uso:

Nombre	Inicio de sesión
Descripción	El cliente introduce los datos necesarios para iniciar sesión en la app móvil.
Actores	Cliente
Precondiciones	El cliente debe estar en la pantalla de login.
Flujo	<ol style="list-style-type: none">1. El cliente introduce los datos en los campos del formulario.2. El cliente pulsa el botón de iniciar sesión.<ol style="list-style-type: none">a. Si el cliente cometió algún error al introducir los datos se muestra en la pantalla.b. El cliente inicia sesión y se redirecciona a la pantalla centros.
Postcondiciones	El cliente inicia sesión y se encuentra en la pantalla centros.

Tabla 33.- Caso de uso, iniciar sesión en app móvil

Nombre	Cerrar sesión
Descripción	El cliente pulsa el botón de cerrar la sesión actual.
Actores	Cliente
Precondiciones	El cliente debe estar en la pantalla centros.
Flujo	<ol style="list-style-type: none"> 1. El cliente pulsa el botón de cerrar sesión. 2. Se limpian los datos de sesión. 3. El cliente es redirigido a la pantalla de login.
Postcondiciones	El cliente no está logueado y se encuentra en la pantalla de login.

Tabla 34.- Caso de uso, cerrar sesión en app móvil

8.7.2 Sistema para recordar al cliente

La app permitirá recordar al cliente que inició sesión, así, el cliente no tiene que introducir las credenciales cada vez que abre la aplicación de nuevo.

Al cerrar la sesión se limpiarán los archivos locales y se podrá volver a iniciar sesión introduciendo las credenciales.

En la siguiente tabla puede verse el caso de uso correspondiente:

Nombre	Recordar cliente
Descripción	El cliente es recordado al abrir la app móvil.
Actores	Cliente
Precondiciones	El cliente debe abrir la aplicación móvil.
Flujo	<ol style="list-style-type: none">1. El cliente abre la aplicación y accede a la pantalla de inicio.<ol style="list-style-type: none">a. Si existe registro en los ficheros locales, el cliente inicia sesión automáticamente.b. Si no, se carga la vista de la pantalla de inicio/login.
Postcondiciones	El cliente ha iniciado sesión automáticamente y está en la pantalla centro.

Tabla 35.- Caso de uso, recordar cliente en app móvil

8.7.3 Elección de centro

La app permite al cliente que se ha iniciado sesión elegir el centro en el que querrá reservar un horario. En la siguiente tabla está representado su caso de uso:

Nombre	Elección de centro
Descripción	El cliente elige centro cada vez que quiere reservar un horario en dicho centro a través de la app móvil.
Actores	Cliente
Precondiciones	El cliente ha iniciado sesión.
Flujo	<ol style="list-style-type: none">1. El cliente es redirigido a la pantalla de centros para elegir uno.<ol style="list-style-type: none">a. El cliente escoge uno de los centros y accede a la pantalla de clases.b. Si no escoge y sale, la próxima vez que inicie sesión volverá al mismo punto.
Postcondiciones	El cliente ha elegido centro y se encuentra en la pantalla de clases correspondiente.

Tabla 36.- Caso de uso, elección de centro en app móvil

8.7.4 Elección de clases

La app permite al cliente que se ha iniciado sesión elegir la clase del centro que ha elegido previamente en la que querrá reservar un horario. En la siguiente tabla está representado su caso de uso:

Nombre	Elección de clase
Descripción	El cliente elige clase cada vez que quiere reservar un horario en dicha clase del centro elegido previamente a través de la app móvil.
Actores	Cliente
Precondiciones	El cliente ha iniciado sesión y ha elegido un centro.
Flujo	<ol style="list-style-type: none">1. El cliente es redirigido a la pantalla de clases para elegir una.<ol style="list-style-type: none">a. El cliente escoge una de las clases y accede a la pantalla de horarios.b. Si no escoge y sale, la próxima vez que inicie sesión volverá a la pantalla de centros.
Postcondiciones	El cliente ha elegido clase y se encuentra en la pantalla de horarios correspondiente.

Tabla 37.- Caso de uso, elección de clase

8.7.5 Reserva de horario

La app permite al cliente que se ha iniciado sesión elegir el horario a reservar de la clase del centro que han elegido previamente. En la siguiente tabla está representado su caso de uso:

Nombre	Reserva de horario
Descripción	El cliente elige clase cada vez que quiere reservar un horario en la clase elegida previamente a través de la app móvil.
Actores	Cliente
Precondiciones	El cliente ha iniciado sesión y ha elegido una clase.
Flujo	<ol style="list-style-type: none">1. El cliente es redirigido a la pantalla de horarios para reservar uno.<ol style="list-style-type: none">a. El cliente escoge uno de los horarios y lo reserva.b. Si no escoge y sale, la próxima vez que inicie sesión volverá a la pantalla de centros.
Postcondiciones	El cliente ha reservado horario y se redirecciona a la pantalla de centros.

Tabla 38.- Caso de uso, reserva de horario

8.7.6 Ver perfil de cliente

La app permitirá al cliente ver su perfil. En la siguiente tabla se muestra el caso de uso correspondiente:

Nombre	Ver perfil
Descripción	El cliente quiere ver sus datos.
Actores	Cliente
Precondiciones	El cliente está ubicado en la pantalla principal y hace click en el botón de ver perfil.
Flujo	<ol style="list-style-type: none">1. El cliente pulsa el botón de ver perfil.2. El cliente es redirigido a la pantalla de su perfil.
Postcondiciones	El cliente está en la pantalla de perfil.

Tabla 39.- Caso de uso, ver perfil

8.7.7 Cancelar reserva de horario

La app permitirá al cliente cancelar la reserva de cualquier horario que no haya comenzado aún. En la siguiente tabla se muestra el caso de uso correspondiente:

Nombre	Cancelar reserva
Descripción	El cliente quiere cancelar una reserva de cualquier horario que aún no haya empezado.
Actores	Cliente
Precondiciones	El cliente está ubicado en la pantalla de su perfil y hace click en el botón de cancelar reserva en cualquier reserva que aún no haya comenzado.
Flujo	<ol style="list-style-type: none">1. El cliente pulsa el botón de cancelar reserva.2. El cliente es redirigido a la pantalla de centros.
Postcondiciones	El cliente ha cancelado una reserva y se ha redirigido a la pantalla de centros.

Tabla 40.- Caso de uso, cancelar reserva

Capítulo 9

Pruebas del sistema

En este capítulo se desarrollarán y expondrán las pruebas realizadas al sistema para verificar que su funcionamiento es el esperado, comprobar la integridad ante los fallos y comprobar su robustez y calidad.

9.1 Pruebas en el entorno de desarrollo

En este apartado se verá el resultado de las pruebas realizadas en el entorno de desarrollo Android Studio.

Prueba 1 - Pantalla de carga	
Pasos a realizar	Ejecutar la aplicación sin conexión y con conexión a internet.
Salida esperada	La aplicación debe mostrar un mensaje de error al no haber conexión, si hubiese conexión debe funcionar correctamente.
Salida obtenida	La aplicación muestra un mensaje de error cuando no hay conexión y redirige a la pantalla de login cuando existe algún tipo de conexión a internet, ya sea por datos o wifi.
Resultado	Prueba ejecutada con éxito.

Tabla 41.- Prueba 1, pantalla de carga

Prueba 2 - Pantalla de inicio de sesión	
Pasos a realizar	Introducción de datos erróneos, introducción de credenciales erróneas.
Salida esperada	La aplicación móvil debe mostrar un mensaje de error al introducir los datos erróneamente antes de enviar la solicitud de login. Si no existe ningún error también deberá comprobar que las credenciales son correctas.
Salida obtenida	La aplicación muestra un mensaje de error cuando los datos no tienen el formato deseado. La aplicación muestra un error cuando las credenciales enviadas no son correctas.
Resultado	Prueba ejecutada con éxito.

Tabla 42.- Prueba 2, pantalla de inicio de sesión

Prueba 3 – Elegir centro	
Pasos a realizar	Elegir un centro y cargar la pantalla de clases.
Salida esperada	La aplicación debe mostrar la pantalla de clases.
Salida obtenida	La aplicación muestra la pantalla de clases.
Resultado	Prueba ejecutada con éxito.

Tabla 43.- Prueba 3, elegir centro

Prueba 4 - Elegir clase	
Pasos a realizar	Elegir una clase y cargar la pantalla de horarios.
Salida esperada	La aplicación debe mostrar la pantalla de horarios.
Salida obtenida	La aplicación muestra la pantalla de horarios.
Resultado	Prueba ejecutada con éxito.

Tabla 44.- Prueba 4, elegir clase

Prueba 5 – Reservar Horario	
Pasos a realizar	Se selecciona un horario y se reserva.
Salida esperada	La aplicación debe reservar un horario y redireccionar a la pantalla de centros.
Salida obtenida	La aplicación reserva el horario y redirecciona a la pantalla de centros.
Resultado	Prueba ejecutada con éxito.

Tabla 45.- Prueba 5, reservar horario

Prueba 6 – Ver Perfil	
Pasos a realizar	Se hace click en el botón del perfil.
Salida esperada	La aplicación debe redireccionar a la pantalla de perfil del cliente.
Salida obtenida	La aplicación redirecciona a la pantalla de perfil del cliente.
Resultado	Prueba ejecutada con éxito

Tabla 46.- Prueba 6, ver perfil

Prueba 7 – Cancelación de Reserva	
Pasos a realizar	Se hace click en el botón del perfil y se hace click en el botón de cancelar reserva de la reserva que queramos.
Salida esperada	La aplicación debe mostrar la pantalla de perfil que tiene la lista de reservas y al hacer click en cancelar y se debe cancelar.
Salida obtenida	La aplicación muestra la pantalla de perfil con la lista de reservas y al hacer click en cancelar y se cancela y se redirige a la pantalla de centros.
Resultado	Prueba ejecutada con éxito.

Tabla 47.- Prueba 7, cancelación de reserva

Prueba 8 – Cerrar Sesión	
Pasos a realizar	Hacer click en el botón de cerrar sesión de la pantalla de centros.
Salida esperada	La aplicación debe cerrar sesión y redirigir a la pantalla de login.
Salida obtenida	La aplicación cierra sesión y redirige a la pantalla de login.
Resultado	Prueba ejecutada con éxito.

Tabla 48.- Prueba 8, cierre de sesión

9.2 Pruebas en Aplicación Web

En este apartado se verá el resultado de las pruebas realizadas en la aplicación web.

Prueba 9 - Pantalla de inicio de sesión	
Pasos a realizar	Introducción de datos erróneos, introducción de credenciales erróneas.
Salida esperada	La aplicación debe mostrar un mensaje de error al introducir los datos erróneamente antes de enviar la solicitud de login. Si no existe ningún error también deberá comprobar que las credenciales son correctas.
Salida obtenida	La aplicación muestra un mensaje de error cuando los datos no tienen el formato deseado. La aplicación muestra un error cuando las credenciales enviadas no son correctas.
Resultado	Prueba ejecutada con éxito.

Tabla 49.- Prueba 9, pantalla de inicio de sesión app web

Prueba 10 – Listar centros	
Pasos a realizar	Hacer click en la opción de centros siendo superadministrador.
Salida esperada	La aplicación debe mostrar la lista de centros si es superadministrador. Si es administrador o profesor la opción debe ser mi centro y debe mostrar la información del mismo.
Salida obtenida	La aplicación muestra la lista de centros si es superadministrador. Si es administrador o profesor la opción es mi centro y muestra la información del mismo.
Resultado	Prueba ejecutada con éxito.

Tabla 50.- Prueba 10, listar centros app web

Prueba 11 – Crear Clase	
Pasos a realizar	Estar en la vista de listado de clases, darle al botón de crear clase y rellenar la información necesaria. Se rellena la información de forma errónea y después correctamente.
Salida esperada	La aplicación debe mostrar un error al no rellenar algún campo y otro al rellenarlo erróneamente. Si no hay errores la debe crear sin problemas.
Salida obtenida	La aplicación muestra un error al no rellenar algún campo y otro al rellenarlo erróneamente. Si no hay errores la crea sin problemas.
Resultado	Prueba ejecutada con éxito.

Tabla 51.- Prueba 11, crear clase

Prueba 12 – Registrar Cliente	
Pasos a realizar	Se selecciona la vista de lista de clientes, se cliquea registrar cliente y se rellena la información con el DNI proporcionado por el cliente.
Salida esperada	Si el DNI ya existe en la base de datos debe lanzar un error, al igual si las contraseñas no coinciden. Si no hay problema, debe registrarlo y el cliente puede iniciar sesión en la app móvil.
Salida obtenida	Si el DNI ya existe en la base de datos lanza un error, al igual si las contraseñas no coinciden. Si no hay problema, lo registra y el cliente puede iniciar sesión en la app móvil.
Resultado	Prueba ejecutada con éxito.

Tabla 52.- Prueba 12, registrar cliente

Capítulo 10

Conclusiones

En este capítulo se exponen las conclusiones obtenidas a lo largo del desarrollo del proyecto teniendo en cuenta los objetivos iniciales propuestos, el problema original y los diseños que se eligieron. Finalmente se detallarán las futuras mejoras para el proyecto.

Este proyecto de fin de grado pretendía solucionar el problema de digitalización en las escuelas infantiles, mediante un software de calidad, accesible y gratuito a los usuarios.

10.1 Conclusiones generales

En este apartado se verá el resultado de las pruebas realizadas en el entorno de desarrollo Android Studio y el de PHPStorm.

Para cumplir esto, se desarrolló una aplicación móvil multiplataforma con Flutter y una aplicación web que cumplió los siguientes objetivos propuestos:

1. La aplicación móvil permitirá iniciar sesión y cerrar sesión.
2. La aplicación móvil recordará al usuario cuando vuelva a iniciar la sesión.
3. La aplicación móvil permitirá a los clientes hacer reservas.
4. La aplicación móvil permitirá a los clientes visualizar todas sus reservas realizadas.

5. La aplicación móvil permitirá a los clientes cancelar reservas que no hayan ocurrido aún.
6. La aplicación web permitirá a los usuarios iniciar sesión y cerrar sesión.
7. La aplicación web permitirá al superadministrador dar de alta a usuarios y asignarlos a cualquier centro cívico y dar de alta a clientes.
8. La aplicación web permitirá al superadministrador cambiar cualquier aspecto de la configuración de la misma (nombre completo, nombre abreviado, logo, color principal y secundario, etc.).
9. La aplicación web permitirá al superadministrador crear, editar, listar y eliminar áreas.
10. La aplicación web permitirá al superadministrador crear, editar, listar y eliminar centros cívicos.
11. La aplicación web permitirá a un superadministrador crear, editar, listar y eliminar habitaciones para cualquier centro cívico.
12. La aplicación web permitirá a un superadministrador crear, editar, listar y eliminar clases para cualquier centro cívico y asignarlas a cualquier profesor del mismo centro.
13. La aplicación web permitirá a un superadministrador crear, editar, listar y eliminar horarios de las clases de cualquier centro en las habitaciones del mismo centro cívico.

14. La aplicación web permitirá a un superadministrador crear, editar, listar y eliminar reservas de cualquier cliente a cualquier horario de una clase de cualquier centro cívico.
15. La aplicación web permitirá a un administrador dar de alta a usuarios para su propio centro cívico (otro administrador o un profesor) y dar de alta a clientes.
16. La aplicación web permitirá a un administrador ver los datos de su centro cívico.
17. La aplicación web permitirá a un administrador crear, editar, listar y eliminar habitaciones de su centro cívico.
18. La aplicación web permitirá a un administrador y al superadministrador crear, editar, listar y eliminar clases del centro y asignarlas a cualquier profesor del centro cívico.
19. La aplicación web permitirá a un administrador crear, editar, listar y eliminar horarios de las clases de su centro cívico en una habitación del mismo.
20. La aplicación web permitirá a un administrador crear, editar, listar y eliminar reservas de cualquier cliente a cualquier horario de una clase del centro cívico.
21. La aplicación web permitirá a un profesor crear nuevas clases, y editar y eliminar sus propias clases.

22. La aplicación web permitirá a un profesor crear, editar y borrar horarios de una clase suya.
23. La aplicación web permitirá a los usuarios ver y modificar su perfil.

10.2 Conclusiones de las pruebas

Las pruebas realizadas tanto en el entorno de Android Studio como en el entorno de PHPStorm nos muestran que la aplicación desarrollada es robusta, será estable durante el tiempo que transcurra antes de su posible mantenimiento.

El proceso de corrección de futuros fallos se verá afectado directamente por las acciones de los usuarios, ya que, estos deben reportar los fallos que se produzcan y proponer nuevas funcionalidades.

10.3 Futuras mejoras

La aplicación puede recibir mejoras en sus servicios añadiendo nuevas funcionalidades, podría añadirse emails de confirmación de registro, de cancelación de clases, etc.

Debe comprarse un servidor donde almacenar la aplicación web y subir la aplicación móvil a Google Play y al Apple Store.

Bibliografía

- [1] Android Developers. Introducción a Android Studio.
<https://developer.android.com/studio/intro?hl=es-419>. Visitado el 21 de Junio de 2024.

- [2] Flutter Documentation.
<https://docs.flutter.dev/>. Visitado el 21 Junio de 2024.

- [3] Dart guides.
<https://dart.dev/guides>. Visitado el 21 de Junio de 2024.

- [4] PHPStorm. Getting started
<https://www.jetbrains.com/help/phpstorm/getting-started.html>. Visitado el 21 de Junio de 2024.

- [5] PHP Documentation. <https://www.php.net/docs.php>. Visitado el 21 de Junio de 2024.

- [6] Symfony Documentation. <https://symfony.com/doc/current/index.html>.
Visitado el 22 de Junio de 2024.

- [7] Universidad Complutense de Madrid. Diseño de base de datos relaciones.
https://www.fdi.ucm.es/profesor/fernand/MTIG_/Tema%202%20Dise%C3%B1o%20de%20base%20de%20datos.pdf. Visitado el 22 de Junio de 2024.

- [8] JSON Web Token Introduction. <https://jwt.io/introduction>. Visitado el 24 de Junio de 2024

[9] SQL. Tutorial and Exercises. <https://www.w3schools.com/sql/>. Visitado el 24 de Junio de 2024

[10] PHPMyAdmin. Documentation. <https://docs.phpmyadmin.net/en/latest/>. Visitado el 28 de Junio de 2024

[11] Gympass Wellhub. Wellhub is now Gympass.
https://play.google.com/store/apps/details?id=com.gympass&hl=pt_BR&pli=1. Visitado el 29 de Junio de 2024.

[12] Padel Manager.
<https://play.google.com/store/apps/details?id=com.padelmanager.padelmanager&hl=es>. Visitado el 29 de Junio de 2024.

[13] Framework. What is it?
<https://www.codecademy.com/resources/blog/what-is-a-framework/>. Visitado el 29 de Junio de 2024.

[14] Wikipedia. Xcode. <https://es.wikipedia.org/wiki/Xcode>. Visitado el 30 de Junio de 2024.

[15] Twig. What is it? <https://twig.symfony.com/doc/3.x/intro.html>. Visitado el 30 de Junio de 2024.

[16] Wikipedia. Jakarta Server Pages.
https://en.wikipedia.org/wiki/Jakarta_Server_Pages. Visitado el 30 de Junio de 2024.

[17] Platzi. Backend and Frontend.
<https://platzi.com/blog/que-es-frontend-y-backend/>. Visitado el 30 Junio de 2024.

[18] Docker. Guides.

<https://docs.docker.com/guides/>. Visitado el 30 de Junio de 2024.

[19] Nginx. Documentation.

<https://docs.nginx.com/>. Visitado el 30 de Junio de 2024.

[20] Arsys. Variables entorno PHP en archivos .env.

<https://www.arsys.es/blog/variables-entorno-php-archivos-env>. Visitado el 30 de Junio de 2024.

[21] Symfonycasts. Migraciones doctrine.

<https://symfonycasts.com/es/screencast/symfony-doctrine/migrations>.

Visitado el 30 de Junio de 2024.

[22] Wikipedia. Base64. <https://en.wikipedia.org/wiki/Base64>. Visitado el

30 de Junio de 2024.