

# MEMORIA ESCRITA DEL PROYECTO

CFGS Desarrollo de Aplicaciones Web

advisoryLLC

**Autor:** Montserrat Álvarez

**Convocatoria:** 2S – 2122



## Contenido

<b>1. INTRODUCCIÓN.....</b>	<b>4</b>
1.1. MOTIVACIÓN .....	4
1.2. ABSTRACT .....	5
1.3. OBJETIVOS PROPUESTOS.....	5
1.3.1. OBJETIVOS GENERALES.....	5
1.3.2. OBJETIVOS ESPECÍFICOS .....	6
<b>2. METODOLOGIA .....</b>	<b>6</b>
2.1. CICLO DE VIDA DEL PROYECTO .....	6
2.2. FASES EN LAS QUE SE HA DISEÑADO EL PROYECTO .....	7
2.3. METODOLOGIA DE GESTIÓN USADA.....	8
<b>3. TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN EL PROYECTO.....</b>	<b>9</b>
<b>4. ESTIMACIÓN DE RECURSOS Y PLANIFICACIÓN .....</b>	<b>11</b>
<b>5. ANÁLISIS DEL PROYECTO.....</b>	<b>13</b>
5.1. REQUISITOS .....	13
5.1.1. REQUISITOS FUNCIONALES.....	13
5.1.2. REQUISITOS NO FUNCIONALES: .....	16
5.2. BASE DE DATOS .....	16
5.2.1. ENTIDADES .....	17
5.2.2. RELACIONES, CARDINALIDADES Y PARTICIPACIONES.....	18

---

5.2.3. MODELO ENTIDAD - RELACIÓN .....	18
5.2.4. MODELO RELACIONAL .....	19
5.2.5. MODELO FÍSICO .....	21
5.3. MODELADO DEL SISTEMA .....	22
5.3.1. CASOS DE USO.....	22
5.3.1.1. DIAGRAMA DE CONTEXTO .....	22
5.3.1.2. PLANTILLAS DE DESCRIPCIÓN TEXTUAL.....	23
5.3.1.3. MODELO DE CASOS DE USO O DIAGRAMA ESTRUCTURADO.....	26
5.3.2. DIAGRAMA DE CLASES .....	27
6. DISEÑO DEL PROYECTO .....	28
6.1. DISEÑO DEL SITE (index.php, Services.php y Login.php).....	29
6.1.1. Index.php.....	31
6.1.2. Services.php.....	31
6.1.3. Login.php .....	32
6.1.4. RegistrationForm.php.....	34
6.2. La conexión a la BBDD: archivo Connection.php.....	35
6.3. Vistas de trabajo para usuarios, empleados y administrador .....	35
6.3.1. UserAdvisoryHome.php.....	36
6.3.2. EmployeeAdvisoryHome.php.....	36
6.3.3. AdminAdvisoryHome.php.....	37
6.4. Funcionalidad INSERT del CRUD .....	37
6.4.1. Formularios para las inserciones .....	37
6.4.1.1. RegistrationForm.php.....	37
6.4.1.2. FormAppointment.php .....	37

---

6.4.2. Inserciones en las tablas user, employee y appointment (Insert.php) .....	38
6.5. Funcionalidad READ del CRUD .....	39
6.6. Funcionalidad UPDATE del CRUD.....	40
6.7. Funcionalidad DELETE del CRUD .....	41
7. DESPLIEGUE Y PRUEBAS .....	41
7.1. DESPLIEGUE .....	41
7.2. PRUEBAS.....	42
8. CONCLUSIONES.....	45
8.1. OBJETIVOS ALCANZADOS .....	45
8.2. CONCLUSIONES DEL PROYECTO REALIZADO .....	46
9. VÍAS FUTURAS .....	47
10. GLOSARIO .....	48
11. BIBLIOGRAFÍA / WEBGRAFÍA.....	49
12. ANEXOS.....	50
12.1 ANEXO I: MANUAL DE INSTALACIÓN.....	50
12.2 ANEXO II: MANUAL DE USUARIO Y CREDENCIALES ADMIN .....	51
12.3 ANEXO III: Tabla previa al Diagrama de Gantt .....	52
12.4. ANEXO IV: Diagrama de Gantt .....	52
12.5 ANEXO V: Modelo Entidad-Relación .....	53
12.6 ANEXO VI: MODELO RELACIONAL.....	53
12.7. ANEXO VII: MODELO FÍSICO .....	53

---

<b>12.8. ANEXO VIII: ESTRUCTURA BASE DE DATOS proyecto_daw .....</b>	<b>54</b>
<b>12.9. ANEXO IX: DIAGRAMA DE CASOS DE USO .....</b>	<b>54</b>
<b>12.10. ANEXO X: MAPA DE NAVEGACIÓN DE LA APLICACIÓN Y DISEÑO ESBOZO INICIAL...</b>	<b>55</b>
<b>12.11. ANEXO XI: VISTAS OBTENIDAS EN PRUEBAS REALIZADAS.....</b>	<b>55</b>

## **1. INTRODUCCIÓN**

AdvisoryLLC es una aplicación web que nace para dar visibilidad en internet a una asesoría jurídica vecinal y también para ayudar a la administración de la agenda de servicios ofertados por ésta, promoviendo a una mayor conciliación familiar de sus empleados.

La idea es que un cliente pueda reservar servicios de la asesoría mediante citas, tanto presenciales como telemáticas, a través de una interfaz de usuario, y una vez se haya identificado como cliente registrado.

La aplicación web también ayudará a la gestión de su cartera de clientes, de manera que el usuario administrador podrá listarlos, modificar sus datos, o eliminarlos en caso necesario, a través de una interfaz de administrador, una vez se haya identificado como empleado y siempre que tenga el perfil de administrador, lo que aportará un mayor control de usuarios y de sus datos.

También se pretende con esta aplicación, una mejora en la dinámica diaria de los empleados, de manera que un empleado, una vez se haya registrado como tal, y una vez el administrador le haya dado acceso a la interfaz de empleado, podrá visualizar las citas de los servicios que atenderá la asesoría, y también modificarlas o incluso, eliminarlas en caso necesario.

Y como herramienta para el departamento de recursos humanos, esta aplicación ayudará en la configuración y administración de los empleados que forman parte de la empresa. El administrador, a través de la funcionalidad CRUD (Create, Read, Update, Delete), podrá listar a todos los empleados, modificar el perfil de empleado para darle o arrebatárle el acceso a la interfaz de empleado, o modificar sus datos personales y en último extremo, eliminar empleados cuando cursen baja en la empresa.

### **1.1. MOTIVACIÓN**

Existen muchas empresas que llevan años dedicados a su negocio en nuestras ciudades y pueblos, con un enfoque personal, una sede social física, y un márketing bastante obsoleto basado en el boca a boca, páginas amarillas, etc. La falta de visibilidad en internet de estas empresas, impide que

puedan explotar eficazmente su potencial, perdiendo gran parte del negocio que se derivan a otras empresas sí visibles en la red. Esto ha hecho que vea una gran oportunidad para mejorar este extremo, y este proyecto nace de la motivación de ayudar a la visibilización de estas empresas.

Por otro lado, la falta de organización existente en las agendas de citas, dotan de un menor control en los servicios ofertados, perdiéndose datos en copias físicas de éstos, y también de un menor control en las agendas personales, que impiden una mayor conciliación familiar y personal. Mediante este proyecto también se ayudará a mejorar el control en la gestión diaria.

## 1.2. ABSTRACT

Based on the interest in bringing to personal neighborhood businesses the possibility of having visibility on the Internet that increases their commercial potential, this project will consist of the creation of a web page to give visibility on the internet to a neighborhood advisory, and help in the administration of its agenda of services offered helping a greater family conciliation of the employees.

It will manage, through CRUD functionality (Create, Read, Update and Delete), its different services offered:

- It will be possible to insert data, modify data, and delete data, with a certain user profile;
- The data can be viewed with all user profiles that are registered;
- but, on the other hand, an unregistered user will only be able to navigate through the descriptive views of the web page: Home, services offered, contact...

Its software architecture pattern will be the MVC, and will have a mySql Database, with 4 tables (user, service, appointment, employee).

After validating the project, we can conclude that its deployment will benefit not only the company itself in its visibility and management, but also the users, who will have more options for locating consultancies that can resolve the different administrative and legal procedures, not only in person by appointment with one of our managers in our office, but also telematically through online appointments too

## 1.3. OBJETIVOS PROPUESTOS

### 1.3.1. OBJETIVOS GENERALES

- El primer objetivo que me he planteado con este proyecto es visibilizar AdvisoryLLC para que pueda llegar a muchas más personas y que le permita dar un salto cualitativo en número de clientes y le genere un mayor número de peticiones de servicios ofertados, tanto de proximidad, como online.

- También persigo como objetivo con esta aplicación web, desarrollar un sistema que facilite a AdvisoryLLC la gestión de sus citas, usuarios, y empleados, para una mayor organización y regulación del trabajo diario, que englobarán varios objetivos específicos que desglosaré a continuación en el siguiente punto.

### 1.3.2. OBJETIVOS ESPECÍFICOS

- Desarrollar una aplicación web que permita que los usuarios y los empleados puedan crear una cuenta introduciendo sus datos en el sistema.
- Desarrollar una aplicación web que tenga una vista de escritorio, a la que usuario, empleado y administrador (una vez registrados) puedan conectarse a través de un navegador web.
- Desarrollar una aplicación web que permita que los usuarios puedan reservar citas tanto telemáticas como presenciales para los servicios jurídicos que necesiten.
- Desarrollar una aplicación web que permita que los empleados y el administrador puedan visibilizar las citas reservadas por los clientes, modificarlas o eliminarlas.
- Desarrollar una aplicación web que permita que el administrador pueda visibilizar a los clientes de la empresa, modificar sus datos personales, o en última instancia, pueda eliminar un cliente.
- Desarrollar una aplicación web que permita que el administrador pueda visibilizar a los empleados de la empresa, así como modificar sus datos y también su acceso a la interfaz de empleado, o en última instancia, eliminarlo.

## 2. METODOLOGIA

### 2.1. CICLO DE VIDA DEL PROYECTO

El ciclo de vida de este proyecto serán las diferentes etapas desde el concepto, a lo largo de todas las etapas de desarrollo, y hasta la entrega de éste. Para el diseño del ciclo de vida de este proyecto he elegido el **Modelo en cascada con realimentación**, por varios motivos:

- En primer lugar, desde el inicio conoceremos los requisitos puesto que yo misma seré quien los crearé.
- Además, en caso de detección de alguna deficiencia o fallo en alguna etapa anterior, puedo volver a la etapa anterior e implementar las correcciones o los ajustes que sean necesarios, dado que se trata de un modelo que acepta la realimentación de etapas anteriores.
- Es un modelo secuencial, con una planificación muy natural.
- La calidad resultante utilizando este ciclo de vida del producto final es de la más alta.

El modelo en cascada implementa las etapas del proceso para el desarrollo del software de manera secuencial, de forma que hasta la finalización de una etapa no comienza la siguiente, y con el modelo en cascada con realimentación nos permitirá hacer comprobaciones para constatar si ya es posible el paso a la siguiente etapa, antes de pasar a la siguiente, y si requiere realimentar la anterior, o no.



*Figura 1: Fases del modelo en cascada con realimentación - Fuente: material didáctico MP5. Entornos de desarrollo (DAW)*

## 2.2. FASES EN LAS QUE SE HA DISEÑADO EL PROYECTO

Las fases del ciclo de vida de este proyecto serán: Análisis, Diseño, Codificación, Pruebas y Mantenimiento.

- **PREANÁLISIS:** Fase inicial, muy importante, donde he decidido qué idea desarrollaría, de entre otras ideas iniciales, y según los tiempos y recursos de que disponía, así como de los conocimientos previos y de los retos que me suponía desarrollarla. Además, durante esta fase, tras la elección del tema para el proyecto, he preparado físicamente el entorno de trabajo en el que lo desarrollaría para que fuera lo más funcional y organizado posible, y también el entorno digital (**Seiri**)
- **ANÁLISIS:** Para analizar el sistema, he realizado un estudio de sus requisitos, tanto funcionales como no funcionales, así como de su estructura dinámica, mediante casos de uso, y de su estructura estática, mediante su diagrama de clases. Para esta fase he preparado los repositorios y archivos, y priorizando ubicación en el escritorio de las herramientas que necesitaría utilizar a lo largo del desarrollo, según su frecuencia (**Seiton**).
- **DISEÑO:** Durante la fase de diseño decidí qué elementos serían comunes a todas las vistas, y creé unos esbozos y un mapa de navegación, para plasmar el diseño de las vistas que necesitaría desarrollar a continuación.
- **CODIFICACIÓN:** El árduo trabajo realizado durante el preanálisis y el análisis, me ayudó para que la codificación no arrojara más problemas que los propios del desarrollo, enfocándome en ir cumpliendo los requisitos uno a uno, aunque he tenido que buscar recursos para codificar en PHP utilizando clases de objetos, lo que por un lado, me ha retrasado en la fase de codificación, y por otro, me ha supuesto un reto muy interesante a nivel personal.



- **PRUEBAS:** Durante la fase de Pruebas, he testado las diferentes funcionalidades de la aplicación (pruebas de caja negra), de sus diferentes casos de uso y requerimientos, que permitiera una entrega óptima del producto final.
- **MANTENIMIENTO:** En vías futuras deberé tener en cuenta el mantenimiento de la aplicación, una vez haya sido entregado.

### 2.3. METODOLOGIA DE GESTIÓN USADA

Para este proyecto, **he descartado** para el proyecto la metodología **Scrum**, metodología muy útil para proyectos de desarrollo de software, pero no adecuada a este proyecto, dado que es más adecuada en proyectos multipersonales, y no para este proyecto unipersonal;

**He elegido** para la gestión del proyecto, por mi disponibilidad y mis circunstancias personales, la metodología de gestión **Lean**. Esta metodología nació en Japón, en la fábrica Toyota, porque querían optimizar los procesos de producción, reducir los desperdicios de tiempo en los procesos, reducir gastos en materias primas, y se enfocaron en buscar cómo optimizar los recursos y procesos productivos para eliminar la mayor parte de los desperdicios, también llamado **Muda**, para así optimizar el flujo del valor.

**Las 5 S:** Esta técnica de Lean ayuda a optimizar el flujo del valor y a transformar un entorno de trabajo lleno de cosas innecesarias en un entorno que minimice desperdicios de tiempo y recursos y genere el mayor valor posible. Para este proyecto trataré de encajarlas en las diferentes fases del ciclo de vida del desarrollo de esta aplicación:

- **Seiri** - que en japonés significa **Organizar**: donde separaré lo innecesario de lo necesario, limpiando el escritorio físico para evitar distracciones, y también el ordenador (archivos innecesarios), y determinando las soluciones de software, el IDE con el que realizaré el desarrollo, y las distintas plataformas que realmente podrán ayudarme a aportar valor al cliente, acotando al mínimo número el uso de éstos. Esta fase la realizaré al inicio del proyecto, durante la fase de **Preadánsis**.
- **Seiton** - que en japonés significa **Orden**: donde situaré lo necesario de una manera ordenada en un lugar según su frecuencia de uso y usando la información de una forma intuitiva y gráfica, evitando parrafadas escritas que puedan sustituirse con algún icono, gráfico, etc, más visual e intuitivo.

Esta fase la realizaré durante la fase de **Análisis** y también durante la creación del PowerPoint para la defensa del proyecto.

- **Seiso** - que en japonés significa **Limpieza**: donde eliminaremos lo innecesario en el flujo de trabajo, manteniendo los archivos y agenda de procesos limpia y clara sin posibles distracciones que puedan distorsionar la generación de valor.

Esta fase la haré coincidir con la fase de **Diseño y codificación**, enfocándome en ir cumpliendo uno a uno con los objetivos específicos marcados para el desarrollo de esta aplicación.

- **Seiketsu**- Estandarización y **automatización** del flujo de trabajo. Utilizaré un **tablero Kanban**, que es una herramienta muy visual que genera una visión estática del proyecto, aumenta la productividad y reduce muchísimo el ruido mental.

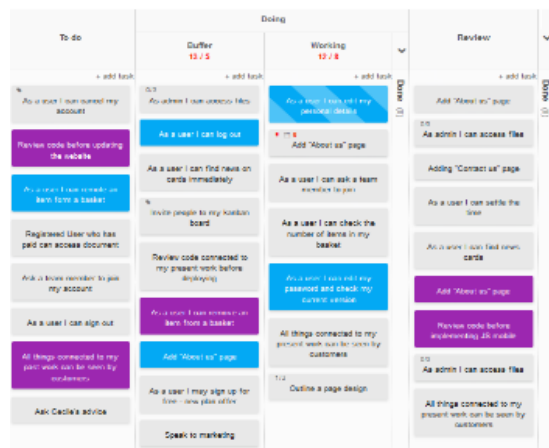


Figura 2: Ejemplo de tablero Kanban - Fuente: <https://kanbantool.com/es/guia-kanban/>

Esta fase la haré coincidir con el final de la fase de codificación y principio de **fase de Pruebas**, de manera que iré cumpliendo uno a uno cada uno de los objetivos propuestos al inicio del proyecto.

- **Shitsuke** - que en japonés significa **Disciplina**: última fase donde continuaremos manteniendo y mejorando la metodología de trabajo, revisándola. Podemos usar diferentes herramientas como por ejemplo, de tratarse de un proyecto multipersonal, haciendo una **caminata Gemba** consistente en revisar el trabajo del equipo para encontrar mejoras sin juzgar y con mucho respeto (que en este caso no procede, pues se trata de revisar durante su trabajo a un equipo que en este caso, no existe). En este proyecto sustituiré ésta por una revisión de todos los puntos de mi trabajo para ver dónde mejorar. Esta fase la haré coincidir con la fase de **Mantenimiento**.

### 3. TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN EL PROYECTO

Las tecnologías he utilizado a emplear en el proyecto serán:

- Base de datos:

- He usado lenguaje de consulta estructurada SQL, declarativo de alto nivel, para trabajar con los datos y con las relaciones entre éstos, en la base de datos relacional que he creado; así como para actualizar, recuperar y eliminar datos almacenados en ésta.

- He creado una BBDD relacional Mysql, con 4 tablas, mediante el sistema de gestión de bases de datos relacionales (RDBMS) de código abierto mySql.
- He utilizado las herramientas MySQL WorkBench y phpMyAdmin tanto para el diseño y modelado de mi base de datos, como para la realización de sus diagramas de Entidad-Relación y paso a modelo Relacional.

- Back End: La codificación de la aplicación se se ha basado en programación orientada a objetos y programación modular, además del uso de estructuras condicionales, bucles, arrays, excepciones, variables y métodos. He utilizado el lenguaje PHP, de código abierto, tanto para la conexión a la BBDD que crearé, como para las distintas funcionalidades CRUD que tendrá la aplicación.

- Front End:

- He utilizado el lenguaje de marcado HTML para implementar la estructura de las VISTAS, para la creación de diferentes elementos del DOM, como por ejemplo, los formularios y tablas HTML, y para la creación de la estructura básica de ésta, o "esqueleto" en el proceso de maquetado web de este proyecto.
- He utilizado también el lenguaje JavaScript para la validación del formato de correo electrónico del formulario de Registro.
- He utilizado CSS3 (Cascading Style Sheets), tanto para dar estilo a los diferentes elementos del DOM como para la obtención de un diseño responsive basado en el diseño mobile first que mejore la experiencia de usuario.

- Despliegue de la aplicación y Pruebas: el desarrollo se ha implementado en local, con el paquete de software libre XAMPP (Apache, MariaDB/MySQL, PHP, Perl), que contiene el servidor web Apache2 y el intérprete para el lenguaje de script PHP, el sistema de gestión de bases de datos mysql y phpMyAdmin.

- phpMyAdmin : Para gestionar MySQL/MariaDB, dado que es una aplicación web gratuita y libre, del paquete de XAMPP, está considerada actualmente como un estándar para la gestión de BBDD mySql en la World Wide Web (red informática mundial).
- Apache, servidor web HTTP, que escuchará en el puerto 80 nuestra conexión HTTP (posteriormente cuando contratemos hosting (así lo indicaré en el apartado de vías futuras de este proyecto) y certificado SSL, tendré que cambiarlo para que redirija ésta al puerto 443).
- mysql/ MariaDB, sistemas relacionales de administración de bases de datos de código abierto, con el que almacenaremos los datos de nuestro sistema.
- php (Hypertext Preprocessor), lenguaje que facilitará la incrustación del lenguaje HTML, el desarrollo web dinámico, y también a la conexión entre el Back-end y el Front-end.

- Prevención de riesgos: He usado los recursos aprendidos en materia de seguridad, posicionando la pantalla adecuadamente para evitar daños en la vista, tomando una posición postural correcta y ergonómica, y tomando los correspondientes descansos a lo largo del proyecto.

**Las herramientas de desarrollo he utilizado para la realización del proyecto son:**

- He utilizado el editor de código fuente totalmente gratuito y de código abierto, desarrollado por Microsoft, Visual Studio Code, como IDE de referencia, dado que es el que utilizo habitualmente y con el que me siento más cómoda.

- He utilizado Chrome, como navegador de referencia.

- He utilizado las herramientas para la creación, manipulación y gestión de la BBDD relacional mysql, para el almacenamiento de datos siguientes:

- MySQL Workbench como SGBD (sistema de gestión de bases de datos) para la creación y manipulación de la base de datos relacional mysql del proyecto, así como para el modelado de diagramas imprescindibles para la creación de la base de datos;
- y phpMyAdmin, como interfaz gráfica de usuario dada que es gratuita y accesible desde el panel de control de xampp;

- He utilizado el paquete office, tanto para la memoria (Word), como para el PowerPoint de soporte al vídeo (powerpoint).

- He utilizado la aplicación diagrams.net, para diferentes diagramas en la fase de Análisis (casos de uso, diagrama de clases).

- He utilizado el programa OBS para la creación y edición del vídeo para la defensa del proyecto.

## **4. ESTIMACIÓN DE RECURSOS Y PLANIFICACIÓN**

Para la estimación de recursos necesarios y planificación de procesos, he realizado en el inicio del proyecto, durante la fase de Preanálisis, un primer Diagrama de Gantt que, mediante el programa Excel del paquete Office, recoge la estimación inicial prevista de tiempos en días / fases del proyecto, para poder tener una planificación aproximada y no demasiado detallada, que me permitió inicialmente hacer una previsión del tiempo y recursos, para llegar a tiempo a las diferentes entregas con el máximo valor del producto.

Posteriormente, a lo largo del desarrollo del proyecto, a medida que he ido recorriendo las diferentes fases del proyecto, he calibrado y rectificado los tiempos, ajustado los iniciales previstos destinados

a cada fase del proyecto, a los tiempos realmente destinados a estos procesos, a fin de visibilizar el tiempo y recursos reales.

Este diagrama de Gantt realizado al final del proyecto, y recoge tanto la estimación inicial, como la planificación real final, aunando ambos datos a fin de poder cotejarlos.

Para la elaboración del diagrama, previamente he creado una tabla con las diferentes tareas a incluir (fases del proyecto), fecha de inicio, días dedicados a la tarea, y fecha de fin de a misma.

Item Proyecto Final	Fecha inicio	Duración	Fecha finalización
<b>Preanálisis - Seiri</b>			
Estimación Inicial	15/02/2022	7	22/02/2022
Tiempo real	15/02/2022	12	27/02/2022
<b>Análisis - Seiton</b>			
Estimación Inicial	21/02/2022	15	08/03/2022
Tiempo real	21/02/2022	21	14/03/2022
<b>Diseño - Seiso</b>			
Estimación Inicial	10/03/2022	45	24/04/2022
Tiempo real	14/03/2022	35	19/04/2022
<b>Pruebas y despliegue - Seiketsu</b>			
Estimación Inicial	25/04/2022	7	02/05/2022
Tiempo real	18/04/2022	7	25/04/2022
<b>Documentación (tema, memoria y powerpoint)</b>			
Estimación Inicial	21/02/2022	30	23/03/2022
Tiempo real durante fase de Análisis y diseño	21/02/2022	30	23/03/2022
Tiempo real durante fase de Pruebas y Powerp	18/04/2022	13	01/05/2022
<b>Video para la defensa del proyecto</b>			
Estimación Inicial	27/04/2022	10	07/05/2022
Tiempo real	02/05/2022	6	08/05/2022
<b>Mantenimiento - Shitsuke</b>			
Estimación	09/05/2022	15	24/05/2022

*Figura 3: Excel previo a la elaboración del Diagrama de Gantt.- Fuente: Elaboración propia*

y utilizando los datos de la tabla creada mediante la aplicación Excel, he insertado un gráfico de barras apiladas 2D que recogiera estos datos y los hiciera visibles, dando forma al Diagrama de Gantt.

- La propuesta me llevó algo más de lo que esperaba, dado que la idea inicial, de ampliar un proyecto anterior, fue descartada finalmente, para realizar un proyecto completo, desde inicio hasta fin, para este proyecto.
- Durante la fase de Análisis, tanto la creación de la base de datos como la realización de los diferentes diagramas, me llevaron más tiempo del previsto también, porque tuve que repasar documentación del módulo de Bases de datos para la realización de los diagramas de Entidad-Relación y paso a Relacional, y también repasar documentación del módulo de Entornos de Desarrollo para el de casos de uso.
- En cambio, tanto la fase de Diseño y codificación, como la de Pruebas, se ajustaron al tiempo previsto inicialmente.
- La realización de la Documentación (Memoria y Powerpoint) la he realizado en dos periodos de tiempo. Un primer periodo durante la fases de Preanálisis y Análisis, y otro periodo una vez realizadas las fases de Diseño y Pruebas.
- Finalmente la fase de la realización del vídeo de defensa me he visto obligada a recortarla para poder entregar el producto en la fecha prevista.

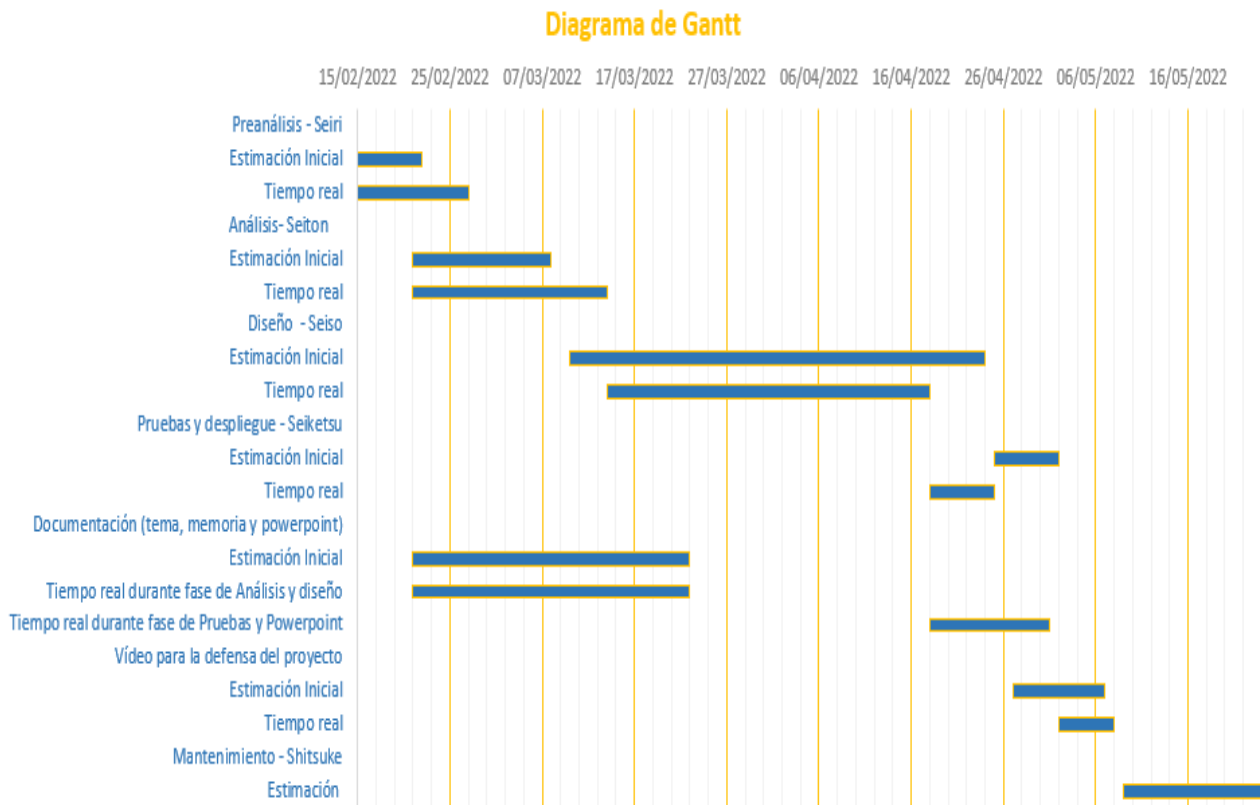


Figura 4: Diagrama de Gantt.- Fuente: Elaboración propia.

## 5. ANÁLISIS DEL PROYECTO

### 5.1. REQUISITOS

#### 5.1.1. REQUISITOS FUNCIONALES

Detallan las funcionalidades la aplicación AdvisoryLLC y cómo interactúa con los diferentes actores externos a él. Separadas según origen, serán:

#### De proceso:

**RF-001:** Un usuario o empleado que todavía no se haya registrado, debe poder navegar por las vistas estáticas HOME, SERVICIOS y LOGIN, pero no debe poder acceder a la home de usuario/empleado.

**RF-002:** Un usuario o empleado que todavía no se haya registrado, no debe poder loguarse y si lo intenta, la aplicación debe redirigirlo a la vista de Registro (formulario de registro) para que se registre.

**RF-003:** Un usuario no registrado debe poder registrarse en la aplicación mediante el formulario de registro y sus datos deberán almacenarse en la tabla usuario.

- RF-004: Un empleado no registrado debe poder registrarse en el sistema mediante el formulario de registro indicando que es empleado y sus datos deberán almacenarse en la tabla empleado asignándole por defecto perfil 0.
- RF-005: Un usuario registrado debe poder loguearse, mediante su e-mail y su contraseña.
- RF-006: Un usuario tras loguearse, debe poder acceder a la home de usuario (UserAdvisoryHome).
- RF-007: Un usuario que ha accedido a la home de usuario, debe poder reservar una cita.
- RF-008: Un usuario que ha accedido a la home de usuario, debe poder regresar a la HOME.
- RF-009: Un empleado registrado debe poder loguearse, mediante su e-mail y su contraseña.
- RF-010: Un empleado tras loguearse por primera vez cuando se registre, no debe poder acceder todavía a la home de empleado (EmployeeAdvisoryHome) sino únicamente a la home de usuario (UserAdvisoryHome ) hasta que el administrador revise sus datos y le asigne el perfil para ello (perfil 1).
- RF-011: Un empleado logueado al que ya le haya cambiado el administrador el perfil de 0 a 1, debe poder acceder a la home de empleado (EmployeeAdvisoryHome).
- RF-012: Un empleado que ha accedido a la home de empleado, debe poder visualizar la lista de citas.
- RF-013: Un empleado que ha accedido a la vista que muestra las citas, debe poder modificar una cita.
- RF-014: Un empleado que ha accedido a la vista que muestra las citas, debe poder eliminar una cita.
- RF-015: Un empleado que ha accedido a la vista que muestra las citas, debe poder regresar a la home de empleado.
- RF-016: Un empleado que ha accedido a la home de empleado, debe poder regresar a la HOME.
- RF-017: El Admin podrán acceder al sistema mediante el formulario Login, con su e-mail y su contraseña.
- RF-018: El Admin tras loguearse, debe poder acceder a la home de administrador (AdminAdvisoryHome), que tiene unas funcionalidades a las que sólo él tendrá acceso.
- RF-019: El Admin que ha accedido a la home de administrador (AdminAdvisoryHome), de debe poder visualizar la lista de citas.
- RF-020: El Admin que ha accedido la vista que muestra las citas, de debe poder modificar los datos de una cita.

**RF-021:** El Admin que ha accedido a la vista que muestra las citas, debe poder eliminar una cita.

**RF-022:** El Admin que ha accedido a la vista que muestra las citas, debe poder regresar a la home de administrador.

**RF-023:** El Admin que ha accedido a la home de administrador (AdminAdvisoryHome), debe poder visualizar la lista de usuarios.

**RF-024:** El Admin que ha accedido a la vista que muestra los usuarios, debe poder modificar los datos de un usuario.

**RF-025:** El Admin que ha accedido a la vista que muestra los usuarios debe poder eliminar un usuario.

**RF-026:** El Admin que ha accedido a la vista que muestra los usuarios, debe poder regresar a la home de administrador.

**RF-027:** El Admin que ha accedido a la home de administrador (AdminAdvisoryHome), debe poder visualizar la lista de empleados.

**RF-028:** El Admin que ha accedido a la vista que muestra los empleados, debe poder modificar los datos de un empleado.

**RF-029:** El Admin que ha accedido a la vista que muestra los empleados debe poder eliminar un empleado.

**RF-030:** El Admin que ha accedido a la vista que muestra los empleados, debe poder regresar a la home de administrador.

**RF-031:** El Admin que ha accedido a la home de administrador, debe poder regresar a la HOME.

**RF-032:** usuario, empleado y admin logueados, desde cualquier vista en la que se encuentren, deben poder navegar por las vistas estáticas HOME, SERVICIOS Y LOGIN desde el nav.

## **De interfaz**

### Formulario Login:

**RF-033:** El campo Password no debe permitir la visibilidad de los datos introducidos.

### Formulario de registro:

**RF-034:** El formulario de registro debe permitir que se indique mediante un tic si se es o no empleado.

**RF-035:** El campo DNI debe avisar si se introducen valores alfanuméricos de menos de 9 dígitos.

**RF-036:** El campo email debe avisar si se introducen valores con formato que no es de e-mail.

**RF-037:** El input correspondiente al perfil de empleado debe tener valor "0" y debe no ser visible, para que no pueda ser modificado.



### 5.1.2. REQUISITOS NO FUNCIONALES:

Describen las diferentes restricciones que impondrá el estándar del software a la aplicación AdvisoryLLC en sus funcionalidades, y que darán lugar a características que emergerán del sistema.

#### **Accesibilidad**

**RNF-001:** La aplicación deberá poderse visualizar desde cualquier navegador estándar actual.

#### **Rendimiento**

**RNF-002:** La aplicación responderá a las peticiones de los usuarios y empleados en menos de 3 segundos.

#### **Usabilidad**

**RNF-003:** La aplicación deberá tener una interface muy intuitiva y amigable, de manera que cualquier usuario, independientemente de sus conocimientos tecnológicos, pueda navegar por las diferentes vistas y cumplimentar formularios de login o de registro con facilidad.

#### **Disponibilidad**

**RNF-004:** El User|Employee|Admin deberá tener acceso a internet.

**RNF-005:** La página web deberá estar disponible 24/7 a través de internet, por lo que se alojará en un hosting, que deberá ser 24/7 y ofrecernos un mantenimiento también de 24/7 por si se diera algún problema.

#### **Requerimientos**

**RNF-006:** El usuario deberá tener acceso a internet desde cualquier dispositivo.

#### **Seguridad**

**RNF-007:** el dominio para alojar la página web, se deberá contratar añadiendo certificado SSL (Secure Sockets Layer), para cifrar los datos compartidos entre servidor y User|Employee|Admin, mediante el protocolos **SSL/TLS**, lo cual aportará seguridad a la transferencia de archivos y datos durante las interacciones con el sistema por parte de User|Employee|Admin.

**RNF-008:** el dominio deberá efectuar backups automáticos diarios y deberá poder facilitar una fácil restauración en caso de fallo del sistema.

### 5.2. BASE DE DATOS

Para la realización de la base de datos para la aplicación advisory LLC:

En primer lugar, he recopilado toda la información relativa a nuestro cliente para obtener los requerimientos. En otro tipo de proyecto, lo haría a través de entrevistas, pero en este proyecto lo

hago a través de la normativa del proyecto aportada en la plataforma, y a partir del tema elegido, presentado y autorizado.

### 5.2.1. ENTIDADES

Para la base de datos de este proyecto, se requieren las 4 entidades siguientes:

1. **USER** (USUARIO),
2. **APPOINTMENT** (CITA),
3. **SERVICE** (SERVICIO)
4. y **EMPLOYEE** (EMPLEADO).

De estas 4 entidades, necesitaré guardar los siguientes datos, que serán los atributos de estas entidades:

**user**: De cada user guardaré:

- su código único de usuario (que ha de ser universal, y que no debe poder repetirse, en este caso será su dni) **userDni**
- su nombre de usuario **userName**
- su apellido **userSurname**
- su dirección **userAddress**
  - su email de usuario **userEmail**
  - su password de acceso **userPassword**

**appointment** : De cada appointment, almacenaré:

- su código de cita (que será único e irrepetible) **appointmentId**
- su fecha y hora **appointmentDateTime**
- su tipología (online o presencial) **appointmentType**
- su estado (0 ó 1) **appointmentState**

**service**: De cada service guardaré:

- su código de servicio (único e irrepetible) **serviceId**
- su nombre de servicio) **serviceName**
- su descripción **serviceDescription**
- su precio **servicePrice**

**employee**: Y de cada employee guardaré:

- su código único e irrepetible (su dni en este caso) **employeeDni**
- su nombre **employeeName**
- su apellido **employeeSurname**
- su dirección **employeeAddress**

- su email **employeeEmail**
- su password **employeePassword**
- su perfil de empleado (que le dará acceso a unas u otras funcionalidades de la aplicación según su perfil departamental) **employeeProfile**

### 5.2.2. RELACIONES, CARDINALIDADES Y PARTICIPACIONES

Partiendo de las siguientes premisas, construiré las relaciones entre las entidades, buscando las cardinalidades y participaciones adecuadas en cada caso.

- Las relaciones las indicaremos con un rombo y un verbo en su interior.
- Las cardinalidades las indicaremos en uno de los vértices libres del rombo y formato X:X
- Las participaciones las indicaremos centro de un paréntesis con el formato (X, X)

Relación **USER** have **APPOINTMENT**:

- Cada usuario puede tener ninguna, una, o muchas citas ..... participación(0-n)
- Pero una cita solamente puede ser tenida por un sólo usuario ..... participación(1-1)

Relación **APPOINTMENT** provide **SERVICE**:

- Cada cita será para proveer uno y sólo un servicio..... participación (1-1)
- Y un servicio se proveerá en una cita ..... participación (1-1)

Relación **SERVICE** manage **EMPLOYEE**:

- Cada servicio estará gestionado por uno y sólo un empleado..... participación (1,1)
- Un empleado podrá gestionar de 0 a n servicios ..... participación (0,n)

En segundo lugar, una vez recopilada toda la información requerida para poder pasar a esta fase, realizaré el modelo Entidad-Relación, que es el modelo conceptual que mostrará las entidades, atributos, y relaciones definidos en los anteriores requerimientos, en un diagrama.

### 5.2.3. MODELO ENTIDAD - RELACIÓN

Una vez estructurada la información que queremos que tenga nuestra base de datos, construiremos el modelo Entidad-Relación, indicando las Entidades en mayúsculas y singular, los atributos en minúscula, singular, y dado que no puede haber espacios entre palabras, usaremos camelCase. Para evitar caracteres especiales, no utilizaré tildes ni tampoco la letra "ñ" para este modelo.

He diseñado el modelo E-R con el programa excel del paquete Office, dando como resultado el siguiente diagrama Entidad- Relación:

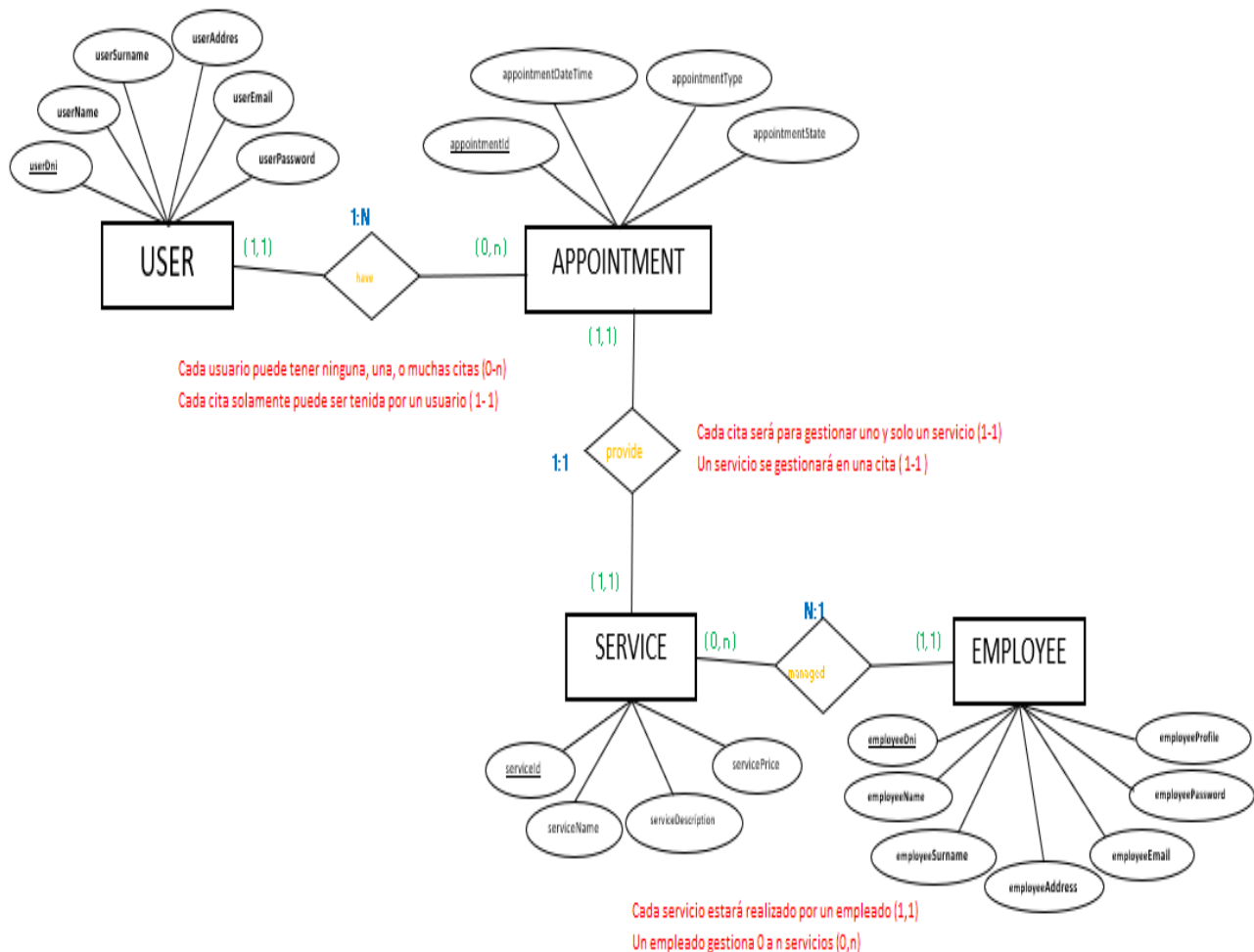


Figura 5: Entidad - Relación - Fuente: Elaboración propia.

En tercer lugar, he realizado el paso del modelo Entidad-Relación a modelo Relacional.

Las entidades pasan a ser tablas, y sus atributos pasarán a ser campos de las tablas. Según la cardinalidad existente en cada una de las relaciones entre las entidades, se construirán las tablas resultantes.

#### 5.2.4. MODELO RELACIONAL

Relación **managed** que se da entre **EMPLOYEE** y **SERVICE**:

- Cada servicio estará gestionado por uno y sólo un empleado..... participación (1,1)
- Un empleado podrá gestionar de 0 a n servicios ..... participación (0,n)

Esta relación tiene una Cardinalidad N:1 reflexiva, por lo que la entidad **EMPLOYEE**, pasará a ser una tabla, y la entidad **SERVICE**, pasará a ser una tabla donde, además de su clave primaria

serviceld y de sus atributos, también recibirá la clave primaria de la tabla employee como clave foránea, dando lugar a estas dos tablas.

**service** (serviceld, serviceName, serviceDescription, servicePrice, **employeeDni\_fk**)

**employee** (employeeDni, employeeName, employeeSurname, employeeAddress, employeeEmail, employeePassword, employeeProfile)

**Relación have que se da entre las entidades USER y APPOINTMENT:**

- Cada usuario puede tener ninguna, una, o muchas citas ..... participación(0-n)
- Pero una cita solamente puede ser tenida por un sólo usuario ..... participación(1-1)

Esta relación tiene una Cardinalidad N:1 reflexiva, por lo que la entidad USER, pasará a ser una tabla, y la entidad APPOINTMENT, pasará a ser una tabla donde, además de su clave primaria appointmentId y sus atributos, también recibirá la clave primaria de la tabla USER como clave foránea, dando lugar a las tablas:

**user**(userDni, userName, userSurname, userAddress, userEmail, userPassword)

**appointment** (appointmentId, appointmentDateTime, appointmentType, appointmentState, **userDni\_fk**)

(La tabla appointment no quedará finalmente así, cuando tengamos en cuenta la siguiente relación).

**y finalmente la Relación provide entre APPOINTMENT y SERVICE:**

- Cada cita será para proveer uno y sólo un servicio .....participación(1-1)
- Y un servicio se proveerá en una cita ... participación(1-1)

Esta relación tiene una Cardinalidad 1:1, que genera dos tablas donde una de ellas recibe como clave foránea la clave primaria de la otra tabla.

En este caso la tabla appointment recibirá la clave primaria de la tabla service como clave foránea, dando como resultado estas dos tablas:

- la tabla que ya teníamos y que no modificaremos:

**service** (serviceld, serviceName, serviceDescription, servicePrice, **employeeDni\_fk**)

- y la tabla appointment que también tenemos, pero que modificaremos, añadiéndole la clave primaria de la tabla service, que pasará como clave foránea, quedando finalmente la tabla appointment con los siguientes registros:

**appointment** (appointmentId, appointmentDateTime, appointmentType, appointmentState, **serviceld\_fk**, **userDni\_fk**)

**Resultado:** El resultado final del paso a Relacional, por lo tanto, serán las siguientes tablas y registros:

**user**(userDni, userName, userSurname, userAddress, userEmail, userPassword)

**appointment**(appointmentId, appointmentDateTime, appointmentType, appointmentState, **serviceld\_fk**, **userDni\_fk**)

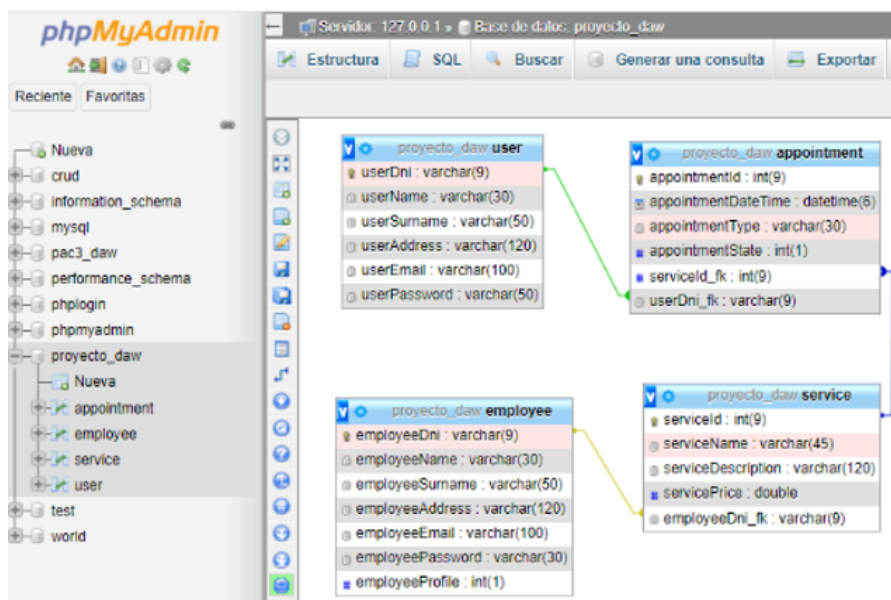
**service**(serviceld, serviceName, serviceDescription, servicePrice, **employeeDni\_fk**)

**employee**(employeeDni, employeeName, employeeSurname, employeeAddress, employeeEmail, employeePassword, employeeProfile)

Tras el modelo Relacional elegiré el **SGBD** (Sistema de Gestión de Bases de Datos) con el que trabajaré. Por las características del proyecto, tal y como se ha indicado en puntos anteriores, usaré el SGBD mySql.

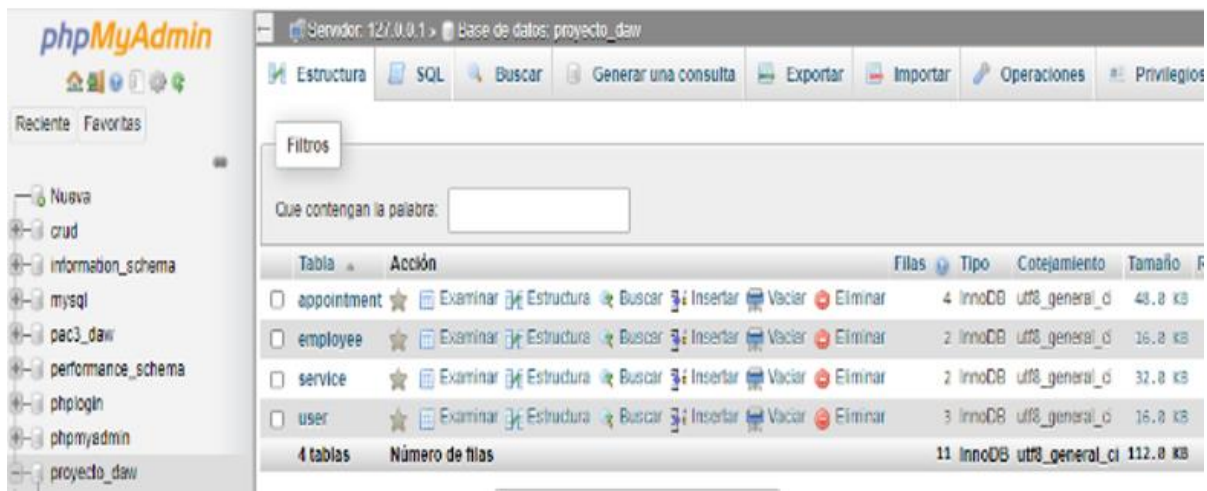
### 5.2.5. MODELO FÍSICO

Y por último, realizaré el modelo físico, esto es, implementaré el modelo Relacional en el software que gestionará la base de datos.



*Figuras 6: Captura de la base de datos proyecto\_daw - Fuente: Elaboración propia.*

En este caso, he creado la base de datos con mySql Workbench y después me ha resultado más intuitivo phpmyadmin y he continuado la creación de las tablas y registros iniciales mediante phpmyadmin. Para acceder a phpmyadmin simplemente se hace clic una vez abierta la aplicación xampp, iniciando Apache y MySQL, y haciendo clic en el botón "Admin" junto a MySQL en el panel de Xampp.



The screenshot shows the phpMyAdmin interface for a database named 'proyecto\_daw'. The left sidebar lists various databases and schemas, including 'crud', 'information\_schema', 'mysql', 'pac3\_daw', 'performance\_schema', 'phplogin', 'phpmyadmin', and 'proyecto\_daw'. The main panel displays the 'Estructura' (Structure) tab for the 'proyecto\_daw' database. It shows a list of tables: 'appointment', 'employee', 'service', and 'user'. Each table entry includes a star icon, a list of actions (Examinar, Estructura, Buscar, Insertar, Vaciar, Eliminar), and details about the table's engine (InnoDB), character set (utf8\_general\_ci), and size. A summary row at the bottom indicates there are 4 tables with a total of 11 rows.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño
<input type="checkbox"/> appointment	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8_general_ci	48.0 KB
<input type="checkbox"/> employee	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_ci	16.0 KB
<input type="checkbox"/> service	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_ci	32.0 KB
<input type="checkbox"/> user	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8_general_ci	16.0 KB
4 tablas	Número de filas	11	InnoDB	utf8_general_ci	112.0 KB

*Figuras 7: Captura de la estructura de la base de datos proyecto\_daw - Fuente: Elaboración propia.*

### 5.3. MODELADO DEL SISTEMA

Para la realización del análisis del sistema, utilizaré dos de los diagramas mas representativos del lenguaje de modelado UML (Unified Modeling Language), que me permitirán visualizar y documentar la aplicación:

- La vista del diagrama UML de casos de uso
- y la vista estática que aporta el diagrama de clases.

#### 5.3.1. CASOS DE USO

Expone cómo se mostrará a un usuario externo a éste, repartiendo el funcionamiento en transacciones importantes para el usuario del sistema.

- A los usuarios externos que interaccionan con el sistema los llamamos actores y los representamos con un "monigote" y su comunicación con el sistema con las líneas de comunicación.
- y a las transacciones funcionales en las que se dividirá, es a lo que llamamos casos de uso

El diagrama de casos de uso será la técnica que realizaré para modelar la vista de casos de uso, que mostrará los requisitos funcionales de advisoryLLC, donde tendremos los actores User, Employee y Admin.

Para la construcción del diagrama de casos, pasaremos por 3 fases:

##### 5.3.1.1. DIAGRAMA DE CONTEXTO

Inicialmente, el **Diagrama de contexto** que nos ayuda a identificar cual es el entorno del

sistema, es decir, los límites del sistema  
advisoryLLC que voy a desarrollar, con su  
nombre, e identificando los actores externos  
que interactuarán con el sistema. El **Diagrama inicial** refinará el Diagrama de contexto  
dado que además de incluir los actores ex-  
ternos y el entorno del sistema, también in-  
cluye los diferentes casos de uso.

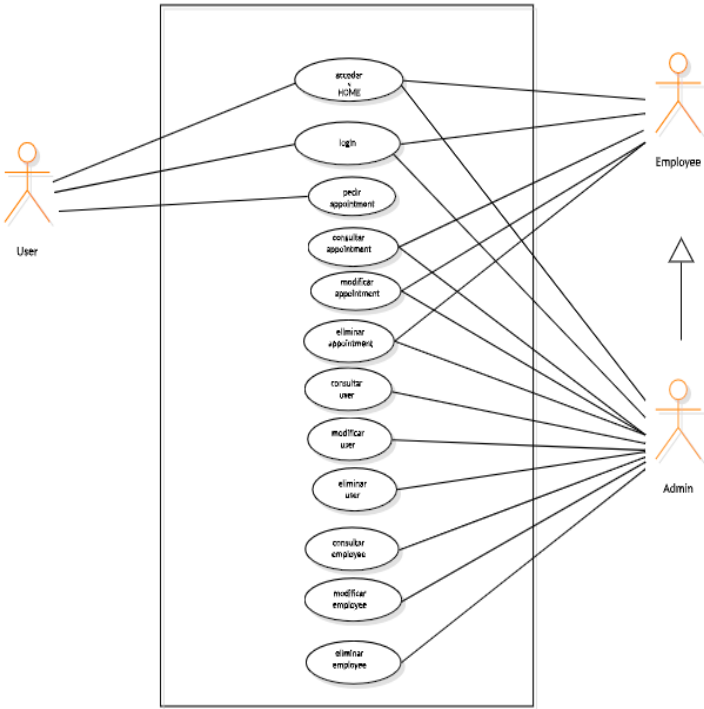


Figura 8: Diagrama de contexto de casos de uso - Fuente: Elaboración propia

5.3.1.2. PLANTILLAS DE DESCRIPCIÓN TEXTUAL

Plantillas con las que refinaré cada uno de los casos de uso detectados para el sistema:

Caso de Uso	Acceder al HOME
Actores	User, Employee, Admin
Resumen	El User tanto si está logueado como si no, y también el Employee y el Admin, pueden acceder a la vista de INICIO (HOME) y navegar por las vistas
Precondiciones	El User Employee Admin debe insertar la dirección URL en la barra de navegación
Postcondiciones	La página web muestra la vista principal HOME
Flujo de Eventos	
Actor	Sistema
1. El User Employee Admin accede a la HOME y navega por las vistas estáticas que no requieren de login para acceder a ellas	2. El sistema recibe la visita del User Employee Admin

Caso de Uso	user, Employee, no registrado
Actores	User, Employee
Resumen	El User Employee se intenta loguear pero al no estar registrado, el sistema lo redirige al formulario de registro
Precondiciones	El User Employee ha accedido a la vista Login e intenta loguearse
Postcondiciones	El User Employee se registra en el sistema mediante el formulario de registro
Flujo de Eventos	
Actor	Sistema
1. El User Employee en la vista Login introduce su email y contraseña y clicka el botón submit	2. El sistema redirige al User Employee al formulario de registro para que se registre
3. El User Employee introduce sus datos en el formulario de registro y clicka el botón submit	4. El sistema registra al User Employee y lo redirige al formulario Login para que, ahora sí, pueda loguearse



<b>Caso de Uso</b>	<b>Loguearse</b>
<b>Actores</b>	User, Employee, Admin
<b>Resumen</b>	El User   Employee   Admin se loguea
<b>Precondiciones</b>	El User   Employee   Admin clicó el enlace de Login en la barra de navegación, para loguearse
<b>Postcondiciones</b>	El User   Employee   Admin accede a la vista advisoryer home si está registrado, o bien accede al formulario de registro si no lo está
<b>Flujo de Eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. El User   Employee   Admin clicó el enlace de Login en la barra de navegación, para loguearse 3. El User   Employee   Admin se loguea 7. El User   Employee cumple el formulario de registro User   Employee con sus datos (una única vez) indicando si es usuario o empleado	2. El sistema redirige al User   Employee   Admin al formulario de login para que se logúee 4. El sistema comprueba si el User   Employee (Admin sí que lo estará) logueado está registrado 5. Si el User   Employee   Admin sí está registrado, lo redirige a la vista advisoryer home donde según su perfil tendrá unas funcionalidades u otras 6. Si el User   Employee no está registrado (Admin sí que lo estará), lo redirige al formulario de registro 8. El sistema almacena los datos del formulario de registro User   Employee en la base de datos del sistema, en la tabla User o en la tabla Employees 9. El sistema dirige al User   Employee   Admin registrado al formulario Login para que se logúee
<b>Caso de Uso</b>	<b>Pedir Cita</b>
<b>Actores</b>	User
<b>Resumen</b>	El User si está registrado, puede acceder a la vista donde puede solicitar una cita para una asesoría
<b>Precondiciones</b>	El User debe estar registrado El User debe haberse logueado
<b>Postcondiciones</b>	El User tiene un appointment.
<b>Flujo de Eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. El User dentro de la vista UserAdvisoryHome clicó la opción para pedir una cita para un servicio de asesoría 12. El User cumplimenta el formulario de solicitud de cita en una datetime concreta para un servicio de asesoría	2. La aplicación dirige al User al formulario de solicitud de cita para que se seleccione fecha y hora, tipo de cita (presencial o telemática) y el servicio 8. El sistema almacena los datos del formulario de solicitud de cita en la tabla appointment de la base de datos de la aplicación
<b>Caso de Uso</b>	<b>Consultar Cita</b>
<b>Actores</b>	Employee con perfil 1   Admin
<b>Resumen</b>	Employee   Admin si está registrado, puede acceder a la vista donde puede ver sus citas
<b>Precondiciones</b>	Employee (Admin sí que lo estará) debe estar registrado. Admin sí que lo estará Employee debe tener el perfil 1 en la vista EmployeeAdvisorHome   AdminAdvisoryHome se ha clicado en el botón para consultar citas
<b>Postcondiciones</b>	El User   Employee   Admin tiene acceso a la vista que muestra el listado de citas activas
<b>Flujo de Eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. Employee   Admin se encuentra en EmployeeAdvisoryHome   AdminAdvisoryHome y clicó el botón para visualizar las citas	2. La aplicación dirige al Employee   Admin a la vista que muestra el listado de citas
<b>Caso de Uso</b>	<b>modificar Cita</b>
<b>Actores</b>	Employee con perfil 1   Admin
<b>Resumen</b>	Employee   Admin si está registrado, puede acceder a la vista donde puede modificar una cita para una asesoría
<b>Precondiciones</b>	Employee debe estar registrado y debe tener perfil 1 para tener acceso a EmployeeAdvisoryHome Employee   Admin debe haberse logueado Employee debe tener el perfil 1 Employee   Admin debe haber sido redirigido a EmployeeAdvisoryHome   AdminAdvisoryHome Employee   Admin debe haber clicado el botón Listado Citas Employee   Admin debe haber clicado el botón Actualizar en el registro de la cita que quiere modificar
<b>Postcondiciones</b>	Employee   Admin ha modificado una cita
<b>Flujo de Eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. Employee   Admin se encuentra dentro de EmployeeAdvisoryHome   AdminAdvisoryHome y clicó el botón Listado citas 3. Employee   Admin clicó el botón Actualizar del registro (cita) que quiere modificar 5. Employee   Admin edita el/los campos que desean modificar y pulsa el botón submit	2. La aplicación lo dirige a la vista con el listado de citas donde cada registro tiene un botón para poder modificarlo 4. La aplicación lo redirige al formulario que contiene los datos del registro que quiere modificar, con los inputs editables y botón submit para ejecutar la modificación 6. El registro es modificado en la tabla appointment

<b>Caso de Uso</b>	<a href="#">eliminar Cita</a>
<b>Actores</b>	Employee con perfil 1   Admin
<b>Resumen</b>	Employee   Admin si está registrado, puede acceder a la vista donde puede eliminar una cita para una asesoría
<b>Precondiciones</b>	Employee debe estar registrado y debe tener perfil 1 para tener acceso a EmployeeAdvisoryHome Employee   Admin debe haberse logueado Employee   Admin debe haber sido redirigido a EmployeeAdvisoryHome   AdminAdvisoryHome Employee   Admin debe haber clicado el botón Listado Citas Employee   Admin debe haber clicado el botón Borrar en el registro de la cita que quiere modificar
<b>Postcondiciones</b>	Employee   Admin ha eliminado una cita
<b>Flujo de Eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. Employee   Admin se encuentra dentro de EmployeeAdvisoryHome   AdminAdvisoryHome y clic el botón Listado citas 3. Employee   Admin clic el botón Borrar del registro (cita) que quiere eliminar	2. La aplicación lo dirige a la vista con el listado de citas donde cada registro tiene un botón para poder eliminarlo 4. El registro es eliminado en la tabla appointment y la aplicación muestra un mensaje informativo

<b>Caso de Uso</b>	<a href="#">Consultar User</a>
<b>Actores</b>	Admin
<b>Resumen</b>	El Admin accede a la consulta de Users
<b>Precondiciones</b>	El Admin debe estar logueado El Admin se encuentra en la vista advisoryer home donde tiene (sólo el Admin) junto al resto de funcionalidades, la funcionalidad de consultar User/s
<b>Postcondiciones</b>	El Admin accede a los datos de User/s
<b>Flujo de Eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. El Admin clic la opción para consultar User/s en la vista Advisoryers	2. El sistema redirige al Admin a la vista que le permite visualizar User/s

<b>Caso de Uso</b>	<a href="#">Modificar User</a>
<b>Actores</b>	Admin
<b>Resumen</b>	El Admin puede modificar los datos de un User
<b>Precondiciones</b>	El Admin debe estar logueado El Admin debe haber clicado en el botón Lista usuarios la vista AdminAdvisoryHome El Admin se encuentra en la vista en la que se muestra la lista de usuarios
<b>Postcondiciones</b>	El Admin modifica los datos del usuario
<b>Flujo de Eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. Admin se encuentra dentro de AdminAdvisoryHome y clic el botón Listado usuarios 3. Admin clic el botón Actualizar del registro (usuario) que quiere modificar 5. Admin edita el/los campos que desea modificar y pulsa el botón submit	2. La aplicación lo dirige a la vista con el listado de usuarios donde cada registro tiene un botón para poder modificarlo 4. La aplicación lo redirige al formulario que contiene los datos del registro que quiere modificar, con los campos activos y botón submit para ejecutar la modificación 6. El registro es modificado en la tabla user

<b>Caso de Uso</b>	<a href="#">Eliminar User</a>
<b>Actores</b>	Admin
<b>Resumen</b>	El Admin puede eliminar los datos de un User
<b>Precondiciones</b>	El Admin debe estar logueado El Admin debe haber clicado en el botón Lista usuarios la vista AdminAdvisoryHome El Admin se encuentra en la vista en la que se muestra la lista de usuarios
<b>Postcondiciones</b>	El Admin modifica los datos del usuario
<b>Flujo de Eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. Admin se encuentra dentro de AdminAdvisoryHome y clic el botón Listado usuarios 3. Admin clic el botón Borrar del registro (usuario) que quiere eliminar	2. La aplicación lo dirige a la vista con el listado de usuarios donde cada registro tiene un botón para poder eliminarlo 4. El registro es eliminado de la tabla user

<b>Caso de Uso</b>	<b>Consultar Employee</b>
<b>Actores</b>	Admin
<b>Resumen</b>	El Admin accede a la consulta de Employee
<b>Precondiciones</b>	El Admin debe estar logueado El Admin se encuentra en la vista AdminAdvisoryHome donde tiene (sólo el Admin) junto al resto de funcionalidades, la funcionalidad de consultar empleados
<b>Postcondiciones</b>	El Admin accede al listado de empleados
<b>Flujo de Eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. El Admin clic el botón Listado empleados en la vista AdminAdvisoryHome que sólo ve él	2. La aplicación redirige al Admin a la vista que le permite visualizar la lista de empleados

<b>Caso de Uso</b>	<b>Modificar Employee</b>
<b>Actores</b>	Admin
<b>Resumen</b>	El Admin puede modificar los datos de un Employee
<b>Precondiciones</b>	El Admin debe estar logueado El Admin debe haber clicado en el botón Lista empleados la vista AdminAdvisoryHome El Admin se encuentra en la vista en la que se muestra la lista de empleados
<b>Postcondiciones</b>	El Admin modifica los datos del Employee
<b>Flujo de Eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. Admin se encuentra dentro de AdminAdvisoryHome y clic el botón Listado empleados 3. Admin clic el botón Actualizar del registro (employee) que quiere modificar 5. Admin edita el/los campos que desean modificar y pulsa el botón submit	2. La aplicación lo dirige a la vista con el listado de empleados donde cada registro tiene un botón para poder modificarlo 4. La aplicación lo redirige al formulario que contiene los datos del registro que quiere modificar, con los campos activos(incluido el de perfil, que el propio empleado tiene en modo hidden) y botón submit para ejecutar la modificación 6. El registro es modificado en la tabla employee

<b>Caso de Uso</b>	<b>Eliminar Employee</b>
<b>Actores</b>	Admin
<b>Resumen</b>	El Admin puede eliminar los datos de un Employee
<b>Precondiciones</b>	El Admin debe estar logueado El Admin debe haber clicado en la vista advisoryer home la opción de búsqueda de Employee El Admin se encuentra en la vista en la que accede al resultado de la consulta realizada, esto es, en la vista con los datos del Employee
<b>Postcondiciones</b>	El Admin elimina el registro de ese Employee
<b>Flujo de Eventos</b>	
<b>Actor</b>	<b>Sistema</b>
1. El Admin en la vista donde accede al Employee buscado, clic la opción para eliminar Employee 3. El Admin clic la opción para eliminar los datos del Employee	2. El sistema redirige al Admin a la vista que le muestra en un formulario, los datos del Employee con los campos inactivos y con la opción de eliminar el Employee 4. El sistema elimina el registro del Employee en la tabla Employee

### 5.3.1.3. MODELO DE CASOS DE USO O DIAGRAMA ESTRUCTURADO

Será la última de las 3 fases para la construcción del diagrama de casos de uso.

A partir de los diferentes detalles de los diferentes casos de uso implementados en las diferentes plantillas de la fase 2, refinará aún más los modelos anteriores, de manera que determinará:

- tanto los actores externos que interactúan con el sistema,
- como los diferentes casos de uso,

- como las relaciones que se dan tanto entre los diferentes casos de uso, como entre los diferentes actores.

He realizado el diagrama estructurado, o diagrama de casos de uso con el programa excel del paquete office.

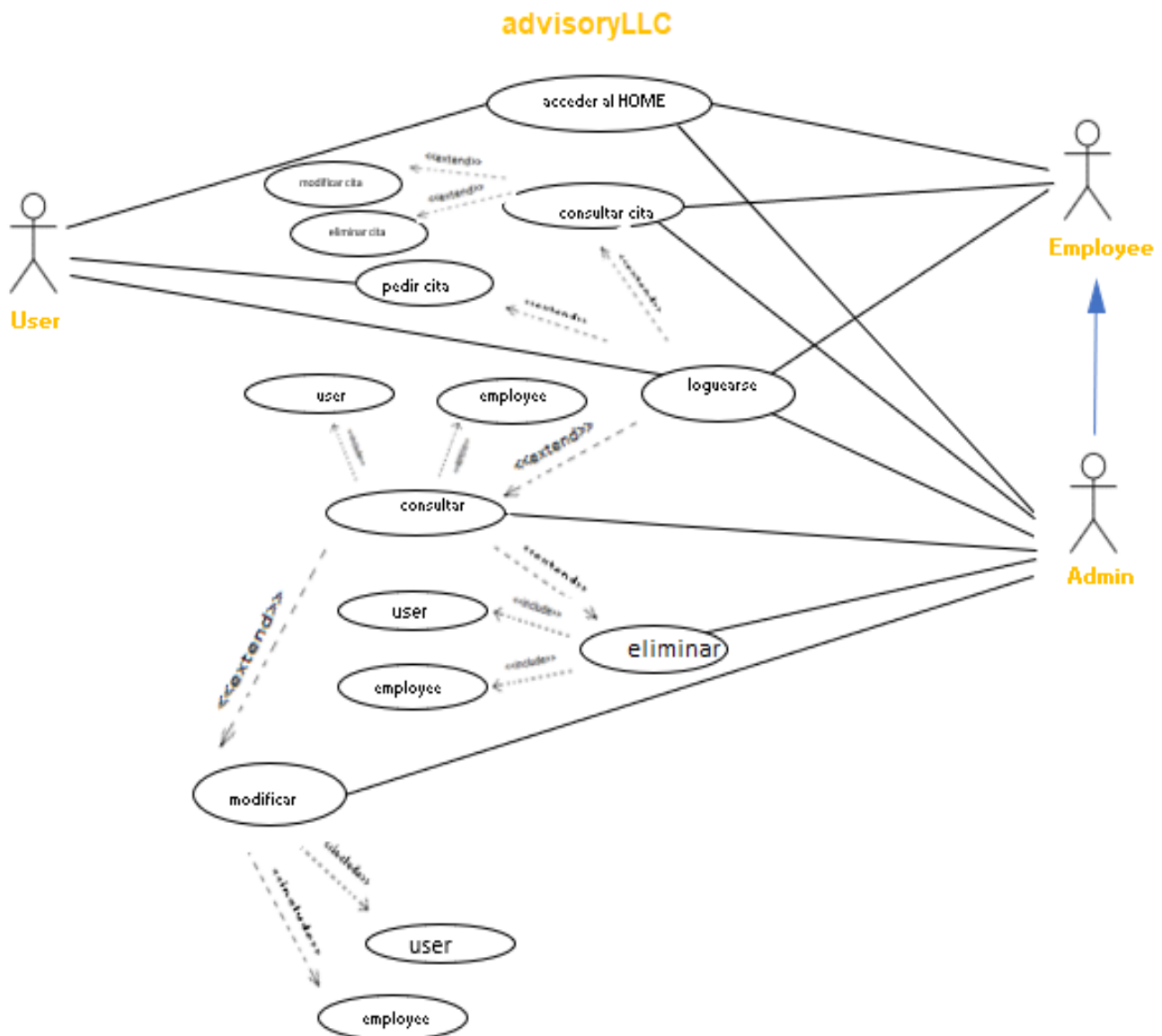


Figura 9: Modelo de Casos de Uso para el sistema advisoryLLC - Fuente: Elaboración propia

### 5.3.2 DIAGRAMA DE CLASES

El diagrama de clases representará la estructura lógica y funcional del sistema **advisoryLLC**.

Utilizaré la arquitectura MVC, por lo que en el diagrama UML de clases solamente incluiré las clases del modelo, relativas a la gestión de los datos del sistema:

- **User**

- **Appointment**
- **Service**
- **Employee**

Los controladores los consideraré como instancias de estas clases.

Las vistas las consideraré sólo como la representación de los datos.

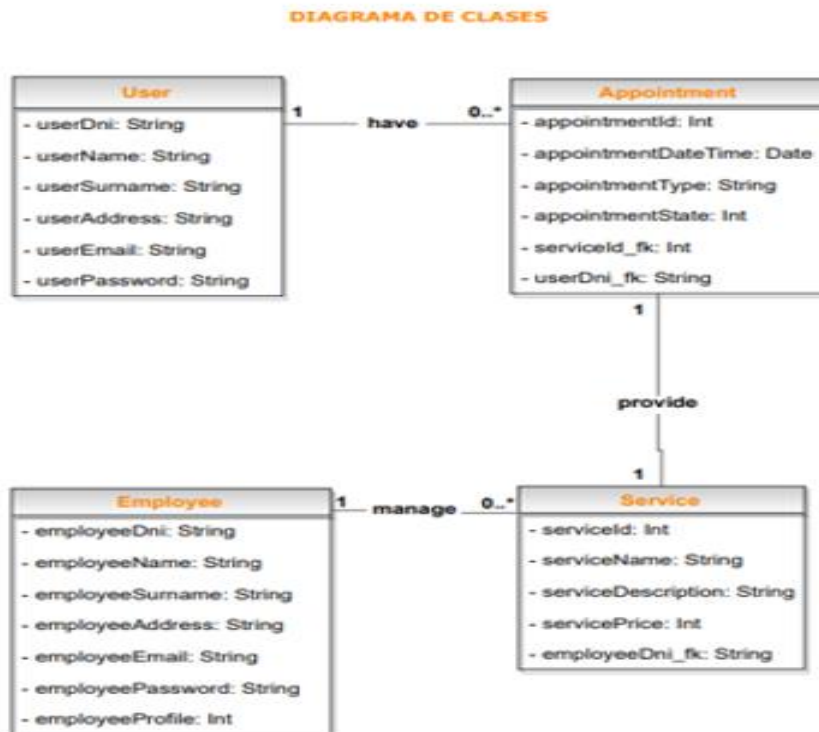


Figura 10: Diagrama de clases advisoryLLC - Fuente: Elaboración propia

## 6. DISEÑO DEL PROYECTO

He realizado un mapa de navegación con un diseño orientativo del sistema a modo de boceto, mediante el programa excel del paquete Office, que consideraré el diseño inicial del sistema, con las diferentes vistas y sus interacciones mediante el menú de navegación, los diferentes formularios y enlaces.

Este diseño realizado previo a la codificación, posteriormente ha sido modificado en algunos aspectos no principales una vez realizada la codificación, como por ejemplo, durante la codificación decidí dotar a empleado de mayor funcionalidad respecto de las citas para que, además de visualizarlas, pudiera modificarlas y anularlas, para una mayor gestión de su trabajo. Estas funcionalidades no estaban contempladas en el mapa de navegación / boceto inicial.

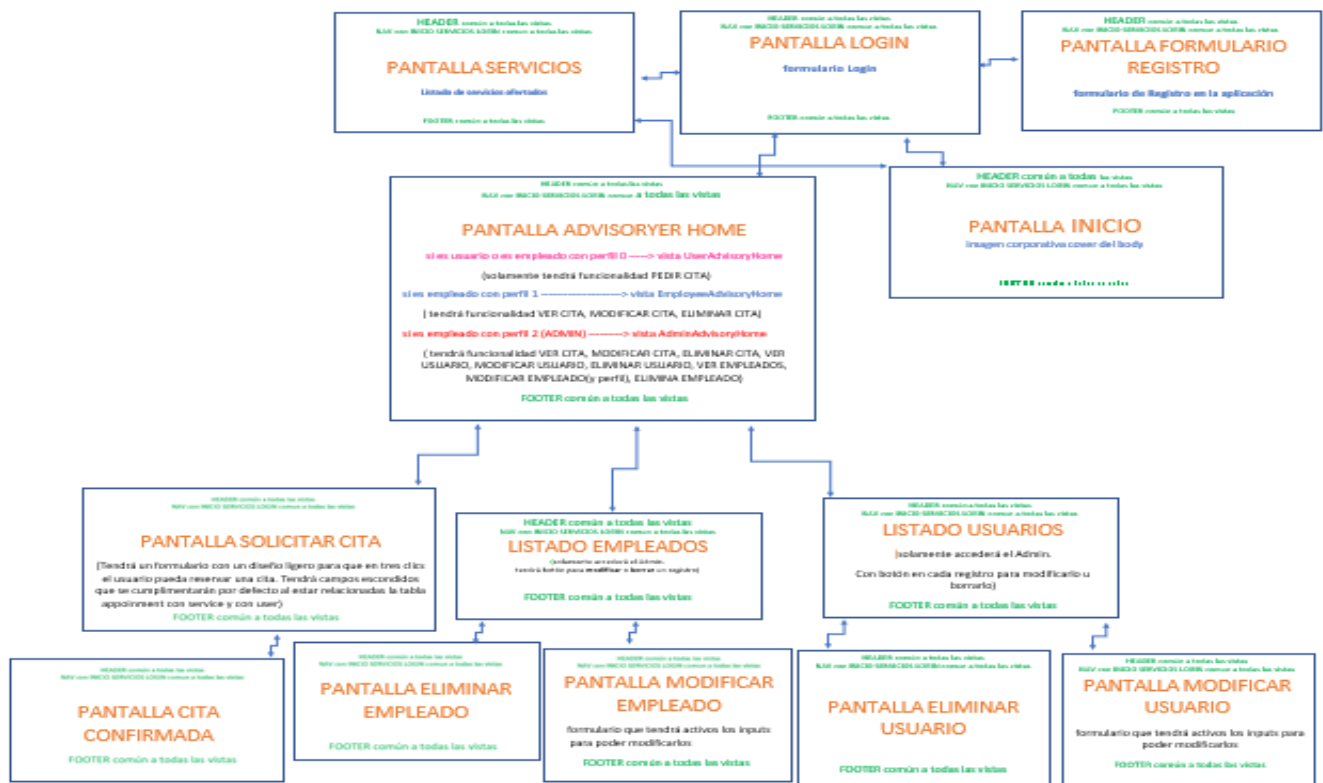


Figura 11: Diseño orientativo y mapa de navegación advisoryLLC - Fuente: Elaboración propia

## 6.1. DISEÑO DEL SITE ([index.php](#), [Services.php](#) y [Login.php](#))

Una vez avanzado el desarrollo del código, he perfilado el borrador de las vistas de INICIO ([index.php](#)), SERVICIOS, y LOGIN, por los que cualquier persona, aunque no se haya registrado, podrá navegar. Será la parte estática de la aplicación donde se podrá navegar a cualquiera de las otras tres vistas.

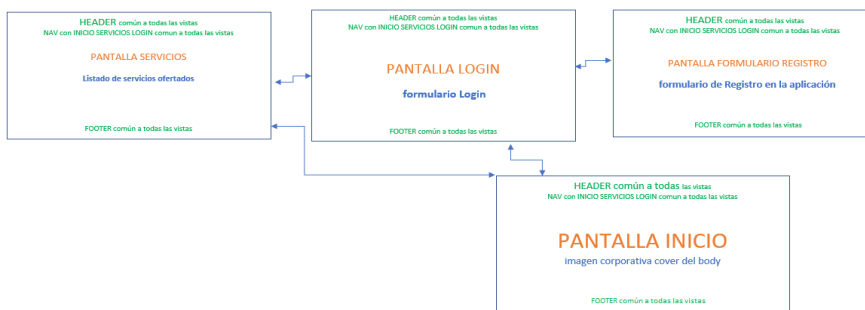


Figura 12: mapa de navegación y diseño previo de las vistas INICIO, SERVICIOS, LOGIN Y FORMULARIO DE REGISTRO de advisoryLLC y - Fuente: Elaboración propia

Como podemos observar, el header será común a las tres vistas (y veremos que también al formulario de registro al que se redirigirá la aplicación en caso de intentar loguearse sin estar registrado), así como al resto de vistas de la aplicación. Así conservaré una estética común de diseño del entorno de la aplicación.



Figura 13: captura de código y vista del header común en todas las vistas del aplicativo advisoryLLC - Fuente: Elaboración propia en ide VSC

La barra de navegación, que contendrá tres botones con los enlaces de navegación a las otras dos vistas también será común en todas las vistas de la aplicación, para que desde cualquier vista, se pueda regresar a INICIO, SERVICIOS o LOGIN.

En la zona vacía del nav, cuando la persona se logúee y sea redirigida a la vista de la aplicación según su perfil, aparecerá imprimido su login, para que perciba más personalizada la interacción con la aplicación.



Figuras 14 a, b y c: captura de código y vista del nav común en todas las vistas del aplicativo advisoryLLC - Fuente: Elaboración propia en ide VSC

Y finalmente el footer, también será común en las diferentes vistas. Así se tendrá acceso a los enlaces a las redes sociales desde cualquier vista. Las cuentas en redes no las incluí en los objetivos marcados y no las activaré para el proyecto, pero dejaré implementados los iconos y enlaces para simplemente cambiar su ruta de la actual (up en la misma vista) a la del perfil, una vez los active (objetivo que incluiré en vías futuras).



Figuras 15 a, b, y c: captura del diseño inicial y diseño final del footer común en todas las vistas del aplicativo advisoryLLC - Fuente: Elaboración propia en ide VSC

Respecto al css, comentar que he utilizado para todas las vistas dos tipos de fuentes, descargadas de fonts.google: Indie Flower, para elementos que he necesitado resaltar con una tipografía más artística, y Montserrat, para homogeneizar el resto. Y también de font-awesome para los iconos de acceso a las redes sociales del footer.

No he querido utilizar ninguna librería para los estilos de los formularios y demás elementos del DOM, ni para hacer responsive las vistas, para lo cual, me he decantado por un diseño web Mobile First, utilizando media queries, ya que hoy en día la mayoría de personas se conectan desde un teléfono móvil en la mayoría de casos.

```
/*diseño web Mobile first con media queries */
@media (min-width: 600px){
  .navQuery{
    display: grid;
    grid-template-columns: repeat(9, 1fr);
  }
}
@media (min-width: 600px){
  #logo{
    display: grid;
    grid-template-columns: repeat(5, 1fr);
  }
}
@media (min-width: 600px){
  .title{
    display: grid;
    grid-template-columns: 3 repeat(5, 1fr);
  }
}
@media (min-width: 600px){
  #bottom{
    display: grid;
    grid-template-columns: repeat(12, 3fr);
  }
}
```

Figura 16: captura del código css para diseño web Mobile First de los elementos footer, header y nav, comunes a todas las vistas del aplicativo advisoryLLC - Fuente: Elaboración propia en ide VSC

### 6.1.1. [Index.php](#)

Será el archivo de arranque de la aplicación cuando el usuario o empleado introduzca la URL en el navegador. Está compuesto por una barra de navegación con enlaces de navegación a las vistas "Servicios" y "Login", una imagen descriptiva del negocio (imagen corporativa) en su body, y un footer donde se incorporarán los enlaces a las redes sociales cuando se despliegue la aplicación.

Desde esta vista se podrá navegar a través de los enlaces de navegación del nav a SERVICIOS y a LOGIN.

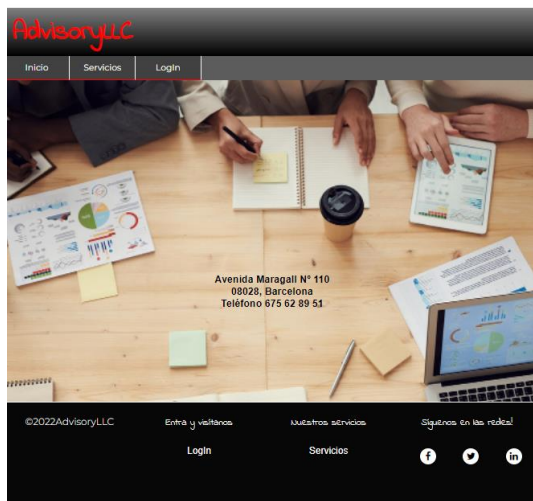


Figura 17: Diseño final pantalla INICIO advisoryLLC - Fuente: Elaboración propia

### 6.1.2. [Services.php](#)

A este archivo se podrá acceder clicando el enlace de navegación del nav, tanto desde index.php, como desde Login.php, como desde cualquier otra vista que contenga la barra de navegación común, como desde el footer de cualquier vista de la aplicación. Está compuesto por el mismo header, misma barra de navegación, para no romper la estética global, igualmente funcional. La imagen del body



está editada con la lista de servicios ofertados por la asesoría para que de un vistazo, se localice el servicio que se necesita solicitar.

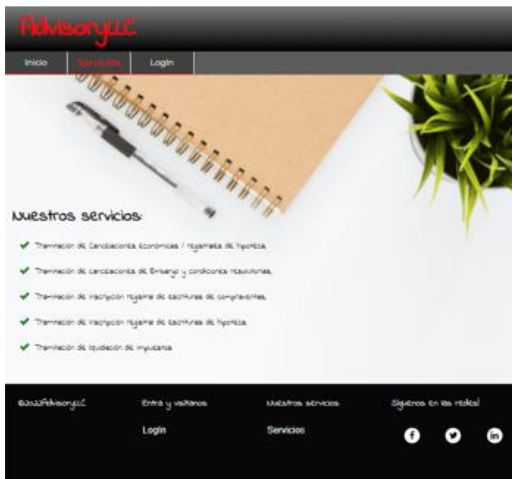


Figura 18: Diseño final pantalla SERVICIOS de advisoryLLC - Fuente: Elaboración propia

### 6.1.3. [Login.php](#)

Será la puerta de entrada de usuarios, empleados y administrador, a la aplicación AdvisoryLLC.

Se llegará a esta vista clicando el enlace de la barra del nav, tanto desde INICIO, como desde SERVICIOS, así como desde el footer, y como no, también será redirigido al Login desde el formulario de registro una vez cumplimente el registro del nuevo usuario/empleador, para que éste acceda a la plataforma.

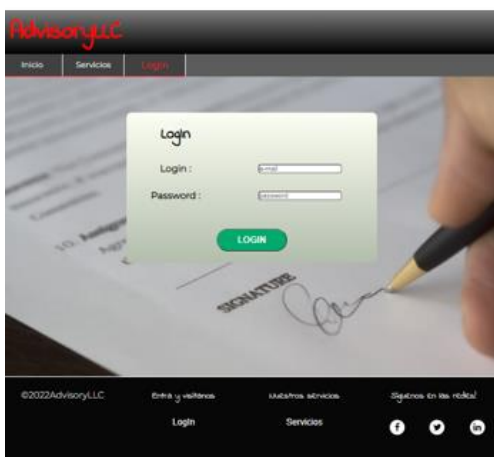


Figura 19: Diseño final pantalla LOGIN de advisoryLLC - Fuente: Elaboración propia

Con footer y barra de navegación común, contendrá en el body una imagen corporativa como background, y el formulario de Login donde se deberá introducir e-mail y contraseña, para autenticarse en el sistema contra la base de datos.

Los datos del formulario login se enviarán al archivo ValidateLogin.php donde serán autenticados contra la base de datos de la siguiente forma:

Preparando inicialmente las consultas preparadas con los marcadores :login y :password. Este uso de marcadores ayudará a la seguridad del sistema, ya que evitará el ataque por inyección sql.

```
$resultu = $connection->prepare("SELECT * FROM proyecto_daw.user WHERE user.userEmail= :login AND user.userPassword= :password");  
$resulte = $connection->prepare("SELECT * FROM proyecto_daw.employee WHERE employee.employeeEmail= :login AND employee.employeePassword= :password");  
$resultep = $connection->prepare("SELECT employeeProfile FROM proyecto_daw.employee WHERE employee.employeeEmail= :login AND employee.employeePassword= :password");
```

Figura 20: Captura consultas preparadas del archivo ValidateLogin.php para seleccionar todos los registros de la tabla user, todos los registros de la tabla employee, y el campo employeeProfile únicamente de la tabla employee, usando marcadores, para evitar inyección SQL - Fuente: Elaboración propia en VSC

Creando tras las consultas una variable donde almacenar los datos rescatados del formulario (\$login y \$password) y estableciendo que los marcadores serán estos datos rescatados como valores.

```
33 $resultu->bindValue(":login", $login);  
34 $resultu->bindValue(":password", $password);
```

Figura 21: Captura del establecimiento de los valores de los marcadores (bindeo) posterior a las consultas en archivo ValidateLogin.php- Fuente: Elaboración propia en VSC

Ejecutando inicialmente la primera consulta preparada que ataca la tabla user en una estructura condicional, donde, si devuelve alguna fila (registro en la tabla user donde login y password sea los introducidos en el formulario login), la aplicación redirigirá al actor user a la vista UserAdvisoryHome porque se tratará de un usuario.

Y si no devuelve ningún registro, entonces ejecutando la segunda consulta preparada que ataca la tabla employee y si devuelve registro (login y password introducidos en el formulario login pertenecen a un empleado de la tabla employee), entonces pasará a evaluar el valor del perfil de empleado tras ejecutar la tercera consulta preparada (extrae el valor del campo perfil del empleado con login y password introducidos en la tabla employee) mediante una estructura condicional switch:

- Si el valor del perfil es 0 será redirigido a la vista UserAdvisoryHome porque hasta que el admin no revise el registro del nuevo empleado y cambie su perfil a 1, no tendrá acceso a la vista de empleado,
- Si el valor del perfil es 1 será redirigido a la vista EmployeeAdvisoryHome,
- Si el valor del perfil es 2 será redirigido a la vista AdminAdvisoryHome

```
65 // obtención de la primera columna de la fila devuelta como resultado ---> https://www.php.net/manual/es/pdostatement.fetchcolumn.php  
66 $employeeProfile = $resultep->fetchColumn();  
67  
68 //si obtengo el perfil  
69 if(isset($employeeProfile)){  
70  
71     //Evaluó $employeeProfile directamente con un condicional switch  
72     switch ($employeeProfile) {  
73         case 0:  
74             //solo verá el UserAdvisoryHome hasta que el admin revise su ficha de empleado y le cambie el perfil a 1 para darle acceso a EmployeeAdvisoryHome  
75             header("location:UserAdvisoryHome.php");  
76             break;  
77         case 1:  
78             //lo redirige a EmployeeAdvisoryHome  
79             header("location:EmployeeAdvisoryHome.php");  
80             break;  
81         case 2:  
82             //Solamente el admin tendrá perfil 2. Será redirigido a AdminAdvisoryHome  
83             header("location:AdminAdvisoryHome.php");  
84             break;  
85     }  
86 }
```

Figura 22: Captura de la estructura condicional Switch en archivo ValidateLogin.php utilizada una vez recibido resultado a la consulta a la tabla empleado, y también recibido el resultado del perfil de usuario, para que en función del valor del perfil del empleado logueado, tenga acceso a la interfaz UserAdvisoryHome(perfil 0), EmployeeAdvisoryHome(perfil 1) o AdminAdvisoryHome (perfil 2) - Fuente: Elaboración propia en VSC

#### 6.1.4. RegistrationForm.php

Esta vista que contiene un formulario html para recoger los datos y registrarlos en la tabla correspondiente (user o employee) de la base de datos de la aplicación.



Figura 23 a y b: Diseño final RegistrationForm.php y captura del código en el formulario donde se da opción a marcar si se es o no empleado, y del campo oculto con el valor del perfil del mismo archivo en VSC en advisoryLLC - Fuente: Elaboración propia

Cabe destacar que utilizaré el mismo formulario para usuario y para empleado, de manera que, aparte de los campos habituales (dni, nombre, apellido, dirección, e-mail y password), que se corresponden tanto con los atributos de la entidad user, como con los atributos de la entidad employee, también tendrá para poder redireccionar a una u otra tabla la inserción:

- un desplegable (select) para que se indique si se es usuario u empleado,
- y también un input en modo oculto con un valor "0" para el perfil de usuario, que se asignará en el campo perfil de la tabla employee por defecto para la opción empleado, lo cual evitará que tenga acceso a la vista de empleados cualquier persona que seleccione que "es empleado".

El registro con los datos capturados del formulario será insertado o bien en la tabla user o bien en la tabla employee (pero con perfil de empleado "0").

Y a continuación, será dirigido a la vista Login para que se logúee y pueda entrar a la plataforma.

Tras loguearse, tanto si ha seleccionado que es usuario, como si ha seleccionado que se es empleado, sólo se tendrá acceso a la home UserAdvisoryHome.php, que es la vista de usuario.

Será el admin quien revise las nuevas incorporaciones de empleados a la empresa, verificando que sean correctas y cambiando el perfil de "0" a "1", para dar acceso al nuevo empleado a la plataforma dándole acceso a la vista EmployeeAdvisoryHome.

## 6.2. La conexión a la BBDD: archivo Connection.php

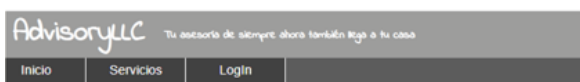
Es el archivo donde he creado la clase Connection y una variable donde la almaceno, que tendrá el modificador de acceso protected para que sea accesible desde las clases que hereden de la clase Connection. Y un constructor donde programo la conexión pasándole por parámetro host, nombre de la bbdd, usuario, contraseña (que dejo en blanco).

```
class Connection{
    //propiedad para poder almacenar la conexión que tendrá el modificador de acceso protected
    protected $connection_bd;
    // constructor
    public function Connection() {
        // dentro programo la conexión propiamente dicha donde el objeto this flecha variable creada es nueva extensión
        // mysql pasándole por parámetro host, nombre de la bbdd, usuario, contraseña (la dejo en blanco)
        //-----PDO USANDO PDO-----
        try{
            $this->connection_bd = new PDO('mysql:host=localhost; dbname=proyecto_daw','root','');
            $this->connection_bd->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
            $this->connection_bd->exec("SET CHARACTER SET utf8");
            return $this->connection_bd;
        } catch(Exception $e){
            echo "Hay un error de conexión en la línea: " . $e->getMessage();
        }
    }
}
```

Figura 24: Captura código clase Connection en archivo Connection.php - Fuente: Elaboración propia en VSC

## 6.3. Vistas de trabajo para usuarios, empleados y administrador

Una vez logueada la persona, siempre que esté registrado en el sistema, la aplicación lo redirigirá a la home específica a su rol, que tendrá unas funcionalidades más o menos restringidas, según su perfil.



PANTALLA ADVISORYER HOME

si es usuario o es empleado con perfil 0 ----> vista UserAdvisoryHome  
 si es empleado con perfil 1 -----> vista EmployeeAdvisoryHome  
 si es empleado con perfil 2 (ADMIN) -----> vista AdminAdvisoryHome

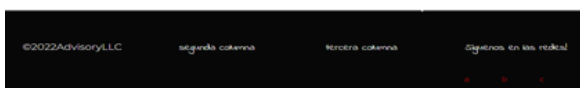


Figura 25: Diseño orientativo pantalla ADVISORYER en función del rol desempeñado - Fuente: Elaboración propia

Como medida de seguridad, en las tres vistas de trabajo, he creado un filtro para que sólo pueda acceder quien se haya logueado correctamente. Si no se almacenó nada en la variable de sesión (porque no accedió desde el login) lo redirijo al login para que se logúee. Así puedo evitar que se pueda acceder poniendo la url en la barra de navegación sin haber pasado por el formulario de Login.

```
// creo un filtro para que sólo pueda acceder quien se haya logueado correctamente
// si no se almacenó nada en la variable de sesión (porque no accedió desde el login) lo redirijo al login
// (para evitar que se pueda acceder poniendo la url en la barra de navegación sin haber pasado por el formulario de login)

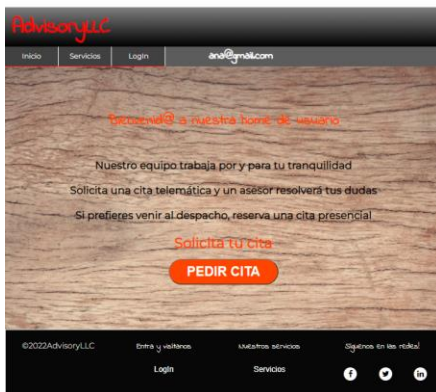
session_start();

if(!isset($_SESSION['login'])){
    header("location:Login.php");
}
```

Figura 26: Captura estructura condicional para redirigir al formulario Login en caso de no haber accedido desde éste - Fuente: Elaboración propia

### 6.3.1. [UserAdvisoryHome.php](#)

Si el usuario se loguea y está registrado en la tabla user, la aplicación lo redirigirá a la home específica para los usuarios, "userAdvisoryHome.php". Así mismo, si el empleado se loguea por primera vez una vez se ha registrado y antes de que el admin le modifique el perfil adquirido por defecto al registrarse, también será redirigido a esta pantalla. Esta vista contendrá, aparte de los elementos comunes ya mencionados (header, nav y footer) y la imagen del body, un botón que le permitirá reservar una cita para el servicio jurídico que necesite (únicamente la funcionalidad crud INSERT). Si se hace clic a éste, se accederá al formulario de solicitud de cita (FormAppointment.php)



Figuras 27: Diseño final pantalla UserAdvisoryhome - Fuente: Elaboración propia

### 6.3.2. [EmployeeAdvisoryHome.php](#)

Si se está registrado en la tabla employee y se tiene perfil "1" (perfil al que habrá sido previamente modificado por el administrador tras la revisión posterior al registro del nuevo empleado), la aplicación redirigirá a la home específica para los usuarios, "EmployeeAdvisoryHome.php" y sí que tendrá acceso a las funcionalidades del perfil de empleado, que serán las funcionalidades CRUD READ, UPDATE y DELETE respecto de las citas que fueron reservadas por los clientes, para tener un mayor control en la gestión de éstas para lo cual tendrá un botón. Inicialmente solamente había marcado como objetivo que las pudiera visualizar, pero es más funcional que pueda modificarlas o eliminarlas.

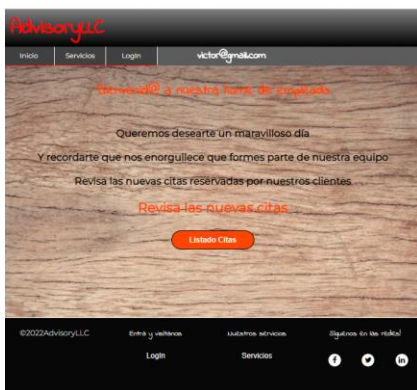


Figura 28: Diseño final pantalla EmployeeAdvisoryhome.php- Fuente: Elaboración propia

### 6.3.3. [AdminAdvisoryHome.php](#)

Si se está registrado en la tabla employee y se tiene perfil "2", la aplicación redirigirá a la home específica para el administrador de la aplicación, "AdminAdvisoryHome.php". Este entorno de trabajo tendrá todas las funcionalidades CRUD activas, tanto para citas, como para usuarios, como para empleados, por lo que será posible consultar, modificar o eliminar los registros de citas, usuarios y empleados, mediante los botones USUARIOS, EMPLEADOS y CITAS.

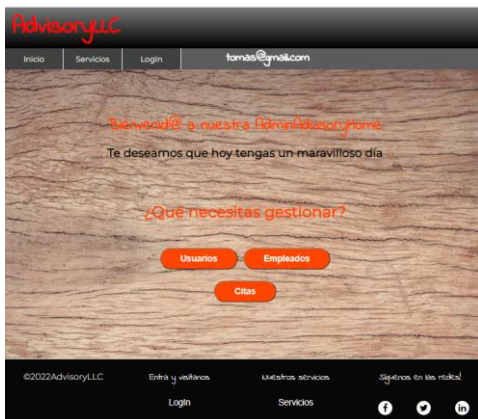


Figura 29: Diseño final pantalla AdminAdvisoryhome.php - Fuente: Elaboración propia

## 6.4. [Funcionalidad INSERT del CRUD](#)

### 6.4.1. [Formularios para las inserciones](#)

#### 6.4.1.1 [RegistrationForm.php](#)

El formulario de registro realiza la inserción en la tabla user o en la tabla employee de los datos cumplimentados del formulario de registro (archivo RegistrationForm.php). Ya comentado en el apartado previo de diseño del SITE (visualmente se tendrá acceso a él aun no habiéndose logueado y por este motivo lo comenté junto a INICIO, SERVICIOS y LOGIN)

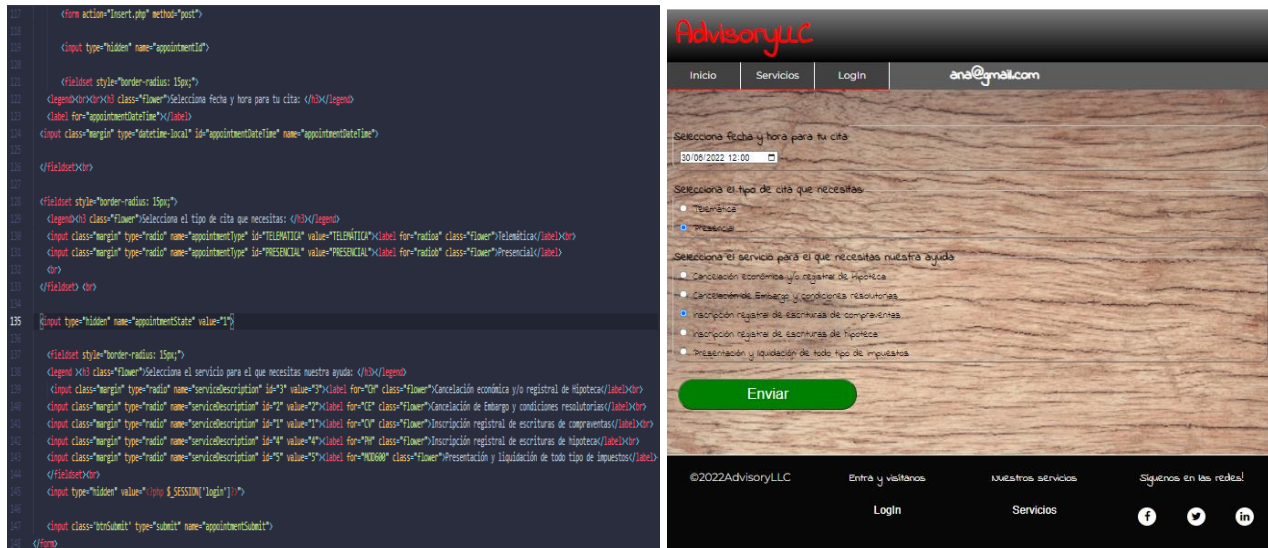
#### 6.4.1.2 [FormAppointment.php](#)

Vista a la que llegará el usuario al hacer clic en PEDIR CITA y que servirá para insertar en la tabla appointment los datos de la reserva del cliente.

He procurado que el diseño no pareciese un formulario al uso, denso y farragoso, para que el cliente tenga la percepción de que en tres clics puede hacer la reserva de su cita, para lo cual he creado:

- aparte del input de tipo datetime-local necesario para la reserva,
- dos inputs de tipo radio donde el usuario sólo deberá clicar, y he ocultado el resto de datos necesarios para la inserción en la tabla pero innecesarios para el conocimiento del usuario. Con esto consigo que el usuario perciba que en pocos segundos alcanzará su objetivo y la inserción en la tabla appointment se hace correctamente.





Figuras 30 y b: Captura del código y Diseño pantalla FormAppointment.php- Fuente: Elaboración propia

## 6.4.2 Inserciones en las tablas user, employee y appointment (Insert.php)

Al hacer clic tanto en el formulario de registro como en el formulario de solicitud de cita, los datos del formulario son enviados mediante el método POST al archivo Insert.php.

Para la inserción, en cada caso, creo una clase que hereda de la clase Connection.php, y creo la función insert en cada caso (insertU(), insertE() e insertA()) mediante consulta preparada, usando marcadores a los que posteriormente asigno los valores y ejecuto la función de inserción, para evitar la inyección SQL. Y la instancio para que ejecute el constructor de Connection conectándose a la BBDD, y hacer uso de la función de inserción creada.

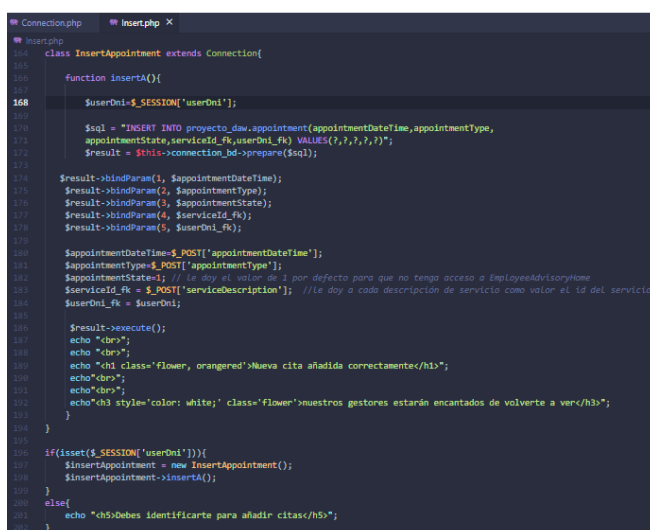


Figura 31: Captura de la clase InsertAppointment() en el archivo Insert.php- Fuente: Elaboración propia

## 6.5. Funcionalidad READ del CRUD

Tendrá esta funcionalidad por supuesto el administrador, de las tres tablas (todas), y el empleado, solamente de la tabla appointment.

Desde la vista de trabajo del administrador AdminAdvisoryHome.php, al hacer clic en uno los botones USUARIOS, EMPLEADOS o CITAS, o también desde la vista de trabajo del trabajador, al hacer clic en el botón CITAS, la aplicación lo redirige al listado de registros de esa tabla que almacena los registros con los que quiere trabajar: ShowEmployees.php, ShowAppointments.php o ShowUsers.php.

ID	FECHA	TIPO	ESTADO	ID SERVICIO	USUARIO
15	14/05/2022 11:00h	PRESENCIAL	1	5	11011011A
17	14/05/2022 12:00h	TELEMÁTICA	1	1	11011011A
19	24/05/2022 09:00h	PRESENCIAL	1	1	22222222B
21	14/05/2022 11:00h	TELEMÁTICA	1	5	22222222B
23	14/05/2022 11:00h	PRESENCIAL	1	2	22222222B
25	25/05/2022 11:30h	TELEMÁTICA	1	4	11011011A
27	06/07/2022 11:30h	TELEMÁTICA	1	1	11011011A
29	14/05/2022 11:00h	PRESENCIAL	1	1	11011011A
31	18/05/2022 13:30h	PRESENCIAL	1	1	11011011A

Figura 32: Captura vista ShowAppointments.php - Fuente: Elaboración propia

ShowAppointments.php (y en cada caso, ShowUsers.php y ShowEmployee.php) llama al archivo ReturnAppointment.php (o ReturnUsers.php o ReturnEmployees.php)

```

1  //?php
2  // requiero la conexión a la BBDD mediante una llamada al archivo Connection.php
3  require ("Connection.php");
4  // Crea la clase ReturnAppointment, que heredará de la clase Connection creada en el archivo Connection.php
5  // y pueda usar las variables y métodos de ésta
6
7  class ReturnAppointment extends Connection {
8
9  // No llamo al constructor de la clase padre Connection() porque la hereda implícitamente
10 // y va a ejecutar el constructor de Connection y va a conectar con la BBDD
11 // una vez he conectado con la BBDD voy a crear un segundo método que hará la consulta sql y me retornará un array de registros
12
13 //-----PDO usando la Librería PDO-----
14
15 function getAppointments(){
16
17     $sentence = $this->connection_bd->query(
18
19         "SELECT appointmentId, DATE_FORMAT(appointmentDateTime, '%d/%m/%Y %H:%i') AS appointmentDateTime, appointmentType,
20         appointmentState,serviceId_fk ,userDni_fk FROM proyecto_daw.appointment"
21     );
22
23     return $sentence->fetchAll(PDO::FETCH_ASSOC);
24
25 }
26
27
28
29

```

Figura 33: Captura archivo ReturnAppointment.php como ejemplo funcionalidad READ del CRUD- Fuente: Elaboración propia

Y tras llamarlo, crea una instancia de la clase ReturnAppointment para ejecutar el constructor de esta clase implícitamente, una variable para que almacene lo que devuelva el método getAppointments() del archivo ReturnAppointment.php, que es un return del array asociativo que devolverá la consulta a la tabla correspondiente (en el caso de appointment además con un formateado de la fecha en la



consulta SELECT); recorre con un bucle foreach el array a mostrar como (as) fila(row) y dentro del bucle foreach, construye una tabla con las filas almacenadas en el array asociativo.

[illegible]

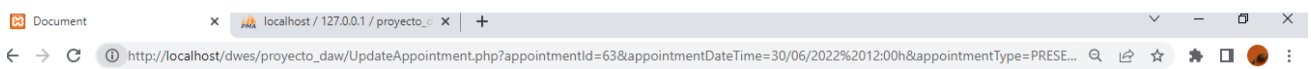
*Figura 34: Captura archivo ShowAppointment.php como ejemplo funcionalidad READ del CRUD- Fuente: Elaboración propia*

## 6.6. Funcionalidad UPDATE del CRUD

Esta funcionalidad la tendrá el empleado con perfil 1 solamente para las citas, y el administrador, tanto para modificar citas, como para gestionar la cartera de clientes, como para gestionar los empleados que forman el equipo.

Desde la vista ShowEmployees.php, ShowAppointments.php o ShowUsers.php existe un botón para actualizar, en cada uno de los registros de la tabla que muestra.

Al pulsar el botón ACTUALIZAR, será redirigido a la vista con el formulario de modificación en cada caso, pasando por URL todos los datos del registro que quiere modificar.



*Figura 35: Captura navegador donde se aprecia el paso por url de los datos - Fuente: Elaboración propia en vsc y vista en Chrome*

Este archivo UpdateAppointment.php (o en su caso UpdateUser.php o UpdateEmployee.php) tiene un formulario con los inputs nutridos con los datos del registro que ha seleccionado, y que he pasado por la url.

[illegible]

*Figura 36: Captura archivo UpdateAppointment.php como ejemplo funcionalidad UPDATE del CRUD- Fuente: Elaboración propia*

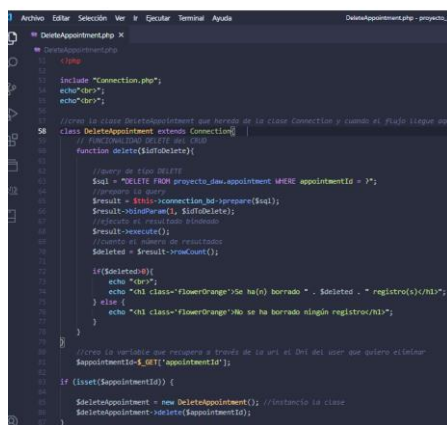
Y los campos de texto del formulario estarán activos para las modificaciones correspondientes, y una vez hecha la modificación, los datos del formulario de modificación se pasarán por el método post a la propia página para luego mostrar de nuevo la vista con el listado de registros con la modificación.

## 6.7. Funcionalidad DELETE del CRUD

Esta funcionalidad la tendrá el empleado con perfil 1 solamente para las citas, y el administrador, tanto para modificar citas, como para gestionar la cartera de clientes, como para gestionar los empleados que forman el equipo.

Desde la vista ShowEmployees.php, ShowAppointments.php o ShowUsers.php existe un botón para borrar, en cada uno de los registros de la tabla que muestra. Al pulsar el botón BORRAR, será redirigido al archivo DeleteAppointment.php (o DeleteUser.php o DeleteEmployee.php), pasando por URL el id del registro que quiere borrar.

DeleteAppointment.php llama al archivo Connection.php, crea la clase DeleteAppointment que hereda de la clase Connection y ejecuta el constructor de Connector, conectando con la bbdd, y crea la función delete(\$idToDelete) que recibe por parámetro el id del registro a eliminar, crea la consulta DELETE con un marcador, para evitar la inyección sql y que pudieran borrar todos los registros de la BBDD, la prepara, la bindea, y finalmente la ejecuta y, mediante una estructura condicional, si devuelve resultado o no devuelve (lo contamos con la función rowCount()) muestra un mensaje u otro por pantalla indicando el resultado de la ejecución.



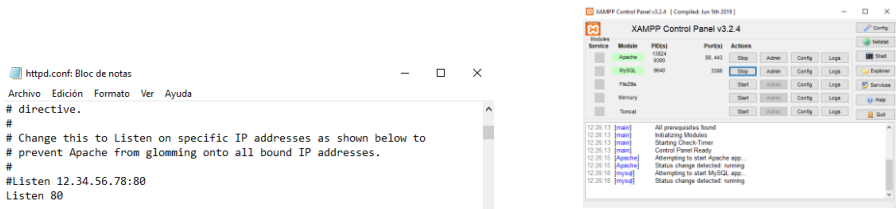
```
11 //...
12 //...
13 //...
14 //...
15 //...
16 //...
17 //...
18 //...
19 //...
20 //...
21 //...
22 //...
23 //...
24 //...
25 //...
26 //...
27 //...
28 //...
29 //...
30 //...
31 //...
32 //...
33 //...
34 //...
35 //...
36 //...
37 //...
38 //...
39 //...
40 //...
41 //...
42 //...
43 //...
44 //...
45 //...
46 //...
47 //...
48 //...
49 //...
50 //...
51 //...
52 //...
53 //...
54 //...
55 //...
56 //...
57 //...
58 //...
59 //...
60 //...
61 //...
62 //...
63 //...
64 //...
65 //...
66 //...
67 //...
68 //...
69 //...
70 //...
71 //...
72 //...
73 //...
74 //...
75 //...
76 //...
77 //...
78 //...
79 //...
80 //...
81 //...
82 //...
83 //...
84 //...
85 //...
86 //...
87 //...
88 //...
89 //...
90 //...
91 //...
92 //...
93 //...
94 //...
95 //...
96 //...
97 //...
98 //...
99 //...
100 //...
101 //...
102 //...
103 //...
104 //...
105 //...
106 //...
107 //...
108 //...
109 //...
110 //...
111 //...
112 //...
113 //...
114 //...
115 //...
116 //...
117 //...
118 //...
119 //...
120 //...
121 //...
122 //...
123 //...
124 //...
125 //...
126 //...
127 //...
128 //...
129 //...
130 //...
131 //...
132 //...
133 //...
134 //...
135 //...
136 //...
137 //...
138 //...
139 //...
140 //...
141 //...
142 //...
143 //...
144 //...
145 //...
146 //...
147 //...
148 //...
149 //...
150 //...
151 //...
152 //...
153 //...
154 //...
155 //...
156 //...
157 //...
158 //...
159 //...
160 //...
161 //...
162 //...
163 //...
164 //...
165 //...
166 //...
167 //...
168 //...
169 //...
170 //...
171 //...
172 //...
173 //...
174 //...
175 //...
176 //...
177 //...
178 //...
179 //...
180 //...
181 //...
182 //...
183 //...
184 //...
185 //...
186 //...
187 //...
188 //...
189 //...
190 //...
191 //...
192 //...
193 //...
194 //...
195 //...
196 //...
197 //...
198 //...
199 //...
200 //...
201 //...
202 //...
203 //...
204 //...
205 //...
206 //...
207 //...
208 //...
209 //...
210 //...
211 //...
212 //...
213 //...
214 //...
215 //...
216 //...
217 //...
218 //...
219 //...
220 //...
221 //...
222 //...
223 //...
224 //...
225 //...
226 //...
227 //...
228 //...
229 //...
230 //...
231 //...
232 //...
233 //...
234 //...
235 //...
236 //...
237 //...
238 //...
239 //...
240 //...
241 //...
242 //...
243 //...
244 //...
245 //...
246 //...
247 //...
248 //...
249 //...
250 //...
251 //...
252 //...
253 //...
254 //...
255 //...
256 //...
257 //...
258 //...
259 //...
260 //...
261 //...
262 //...
263 //...
264 //...
265 //...
266 //...
267 //...
268 //...
269 //...
270 //...
271 //...
272 //...
273 //...
274 //...
275 //...
276 //...
277 //...
278 //...
279 //...
280 //...
281 //...
282 //...
283 //...
284 //...
285 //...
286 //...
287 //...
288 //...
289 //...
290 //...
291 //...
292 //...
293 //...
294 //...
295 //...
296 //...
297 //...
298 //...
299 //...
300 //...
301 //...
302 //...
303 //...
304 //...
305 //...
306 //...
307 //...
308 //...
309 //...
310 //...
311 //...
312 //...
313 //...
314 //...
315 //...
316 //...
317 //...
318 //...
319 //...
320 //...
321 //...
322 //...
323 //...
324 //...
325 //...
326 //...
327 //...
328 //...
329 //...
330 //...
331 //...
332 //...
333 //...
334 //...
335 //...
336 //...
337 //...
338 //...
339 //...
340 //...
341 //...
342 //...
343 //...
344 //...
345 //...
346 //...
347 //...
348 //...
349 //...
350 //...
351 //...
352 //...
353 //...
354 //...
355 //...
356 //...
357 //...
358 //...
359 //...
360 //...
361 //...
362 //...
363 //...
364 //...
365 //...
366 //...
367 //...
368 //...
369 //...
370 //...
371 //...
372 //...
373 //...
374 //...
375 //...
376 //...
377 //...
378 //...
379 //...
380 //...
381 //...
382 //...
383 //...
384 //...
385 //...
386 //...
387 //...
388 //...
389 //...
390 //...
391 //...
392 //...
393 //...
394 //...
395 //...
396 //...
397 //...
398 //...
399 //...
400 //...
401 //...
402 //...
403 //...
404 //...
405 //...
406 //...
407 //...
408 //...
409 //...
410 //...
411 //...
412 //...
413 //...
414 //...
415 //...
416 //...
417 //...
418 //...
419 //...
420 //...
421 //...
422 //...
423 //...
424 //...
425 //...
426 //...
427 //...
428 //...
429 //...
430 //...
431 //...
432 //...
433 //...
434 //...
435 //...
436 //...
437 //...
438 //...
439 //...
440 //...
441 //...
442 //...
443 //...
444 //...
445 //...
446 //...
447 //...
448 //...
449 //...
450 //...
451 //...
452 //...
453 //...
454 //...
455 //...
456 //...
457 //...
458 //...
459 //...
460 //...
461 //...
462 //...
463 //...
464 //...
465 //...
466 //...
467 //...
468 //...
469 //...
470 //...
471 //...
472 //...
473 //...
474 //...
475 //...
476 //...
477 //...
478 //...
479 //...
480 //...
481 //...
482 //...
483 //...
484 //...
485 //...
486 //...
487 //...
488 //...
489 //...
490 //...
491 //...
492 //...
493 //...
494 //...
495 //...
496 //...
497 //...
498 //...
499 //...
500 //...
501 //...
502 //...
503 //...
504 //...
505 //...
506 //...
507 //...
508 //...
509 //...
510 //...
511 //...
512 //...
513 //...
514 //...
515 //...
516 //...
517 //...
518 //...
519 //...
520 //...
521 //...
522 //...
523 //...
524 //...
525 //...
526 //...
527 //...
528 //...
529 //...
530 //...
531 //...
532 //...
533 //...
534 //...
535 //...
536 //...
537 //...
538 //...
539 //...
540 //...
541 //...
542 //...
543 //...
544 //...
545 //...
546 //...
547 //...
548 //...
549 //...
550 //...
551 //...
552 //...
553 //...
554 //...
555 //...
556 //...
557 //...
558 //...
559 //...
560 //...
561 //...
562 //...
563 //...
564 //...
565 //...
566 //...
567 //...
568 //...
569 //...
570 //...
571 //...
572 //...
573 //...
574 //...
575 //...
576 //...
577 //...
578 //...
579 //...
580 //...
581 //...
582 //...
583 //...
584 //...
585 //...
586 //...
587 //...
588 //...
589 //...
590 //...
591 //...
592 //...
593 //...
594 //...
595 //...
596 //...
597 //...
598 //...
599 //...
600 //...
601 //...
602 //...
603 //...
604 //...
605 //...
606 //...
607 //...
608 //...
609 //...
610 //...
611 //...
612 //...
613 //...
614 //...
615 //...
616 //...
617 //...
618 //...
619 //...
620 //...
621 //...
622 //...
623 //...
624 //...
625 //...
626 //...
627 //...
628 //...
629 //...
630 //...
631 //...
632 //...
633 //...
634 //...
635 //...
636 //...
637 //...
638 //...
639 //...
640 //...
641 //...
642 //...
643 //...
644 //...
645 //...
646 //...
647 //...
648 //...
649 //...
650 //...
651 //...
652 //...
653 //...
654 //...
655 //...
656 //...
657 //...
658 //...
659 //...
660 //...
661 //...
662 //...
663 //...
664 //...
665 //...
666 //...
667 //...
668 //...
669 //...
670 //...
671 //...
672 //...
673 //...
674 //...
675 //...
676 //...
677 //...
678 //...
679 //...
680 //...
681 //...
682 //...
683 //...
684 //...
685 //...
686 //...
687 //...
688 //...
689 //...
690 //...
691 //...
692 //...
693 //...
694 //...
695 //...
696 //...
697 //...
698 //...
699 //...
700 //...
701 //...
702 //...
703 //...
704 //...
705 //...
706 //...
707 //...
708 //...
709 //...
710 //...
711 //...
712 //...
713 //...
714 //...
715 //...
716 //...
717 //...
718 //...
719 //...
720 //...
721 //...
722 //...
723 //...
724 //...
725 //...
726 //...
727 //...
728 //...
729 //...
730 //...
731 //...
732 //...
733 //...
734 //...
735 //...
736 //...
737 //...
738 //...
739 //...
740 //...
741 //...
742 //...
743 //...
744 //...
745 //...
746 //...
747 //...
748 //...
749 //...
750 //...
751 //...
752 //...
753 //...
754 //...
755 //...
756 //...
757 //...
758 //...
759 //...
760 //...
761 //...
762 //...
763 //...
764 //...
765 //...
766 //...
767 //...
768 //...
769 //...
770 //...
771 //...
772 //...
773 //...
774 //...
775 //...
776 //...
777 //...
778 //...
779 //...
780 //...
781 //...
782 //...
783 //...
784 //...
785 //...
786 //...
787 //...
788 //...
789 //...
790 //...
791 //...
792 //...
793 //...
794 //...
795 //...
796 //...
797 //...
798 //...
799 //...
800 //...
801 //...
802 //...
803 //...
804 //...
805 //...
806 //...
807 //...
808 //...
809 //...
810 //...
811 //...
812 //...
813 //...
814 //...
815 //...
816 //...
817 //...
818 //...
819 //...
820 //...
821 //...
822 //...
823 //...
824 //...
825 //...
826 //...
827 //...
828 //...
829 //...
830 //...
831 //...
832 //...
833 //...
834 //...
835 //...
836 //...
837 //...
838 //...
839 //...
840 //...
841 //...
842 //...
843 //...
844 //...
845 //...
846 //...
847 //...
848 //...
849 //...
850 //...
851 //...
852 //...
853 //...
854 //...
855 //...
856 //...
857 //...
858 //...
859 //...
860 //...
861 //...
862 //...
863 //...
864 //...
865 //...
866 //...
867 //...
868 //...
869 //...
870 //...
871 //...
872 //...
873 //...
874 //...
875 //...
876 //...
877 //...
878 //...
879 //...
880 //...
881 //...
882 //...
883 //...
884 //...
885 //...
886 //...
887 //...
888 //...
889 //...
890 //...
891 //...
892 //...
893 //...
894 //...
895 //...
896 //...
897 //...
898 //...
899 //...
900 //...
901 //...
902 //...
903 //...
904 //...
905 //...
906 //...
907 //...
908 //...
909 //...
910 //...
911 //...
912 //...
913 //...
914 //...
915 //...
916 //...
917 //...
918 //...
919 //...
920 //...
921 //...
922 //...
923 //...
924 //...
925 //...
926 //...
927 //...
928 //...
929 //...
930 //...
931 //...
932 //...
933 //...
934 //...
935 //...
936 //...
937 //...
938 //...
939 //...
940 //...
941 //...
942 //...
943 //...
944 //...
945 //...
946 //...
947 //...
948 //...
949 //...
950 //...
951 //...
952 //...
953 //...
954 //...
955 //...
956 //...
957 //...
958 //...
959 //...
960 //...
961 //...
962 //...
963 //...
964 //...
965 //...
966 //...
967 //...
968 //...
969 //...
970 //...
971 //...
972 //...
973 //...
974 //...
975 //...
976 //...
977 //...
978 //...
979 //...
980 //...
981 //...
982 //...
983 //...
984 //...
985 //...
986 //...
987 //...
988 //...
989 //...
990 //...
991 //...
992 //...
993 //...
994 //...
995 //...
996 //...
997 //...
998 //...
999 //...
1000 //...
```

Figura 37: Captura archivo DeleteAppointment.php como ejemplo funcionalidad DELETE del CRUD- Fuente: Elaboración propia

## 7. DESPLIEGUE Y PRUEBAS

### 7.1. DESPLIEGUE

El despliegue de la aplicación lo he realizado en local, mediante el paquete XAMPP, que contiene servidor Apache, que escucha en el puerto 80.



Figuras 38 a y b: Captura archivo httpd.conf y vista de panel de xampp - Fuente: Elaboración propia

La carpeta con el código fuente se encuentra ubicada en C:\xampp\htdocs\

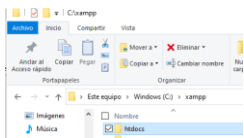


Figura 39: Captura directorio htdocs en C:\xampp\htdocs\ - Fuente: Elaboración propia

y en navegador indico `http://localhost/proyecto_daw/` (en mi caso creé una carpeta intermedia DWES que he querido conservar)

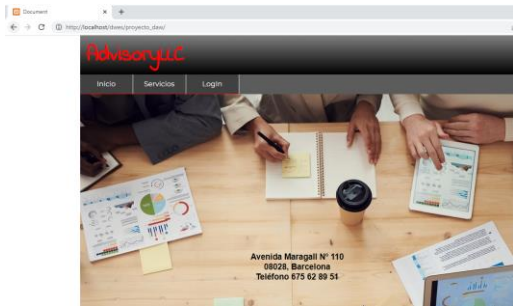


Figura 40: Captura vista index.php al indicar `http://localhost/dwes/proyecto_daw/` en navegador Chrome - Fuente: Elaboración propia

## 7.2. PRUEBAS

He realizado las siguientes pruebas funcionales en caja negra, basándome en los diferentes casos de uso detectados para el sistema AdvisoryLLC:

N.º	ESPECIFICACIÓN DE PRUEBAS
1	<p><b>Objetivo probado:</b> Acceder al HOME (INICIO) sin estar logueado como usuario, y también como empleado</p> <p>quiero acceder a la página de inicio</p> <p>para así navegar por las vistas HOME, SERVICIOS y LOGIN</p> <p>Requisitos probados: RF-001, RF-032, RNF-001, RNF-004</p> <p>Pruebas que realizar: Se introduce la URL en la barra de navegación del navegador. Acceso permitido.</p>

2	<p><b>Objetivo probado: Registro de usuario no registrado</b>  <b>como</b> usuario  <b>quiero</b> registrarme en la plataforma  <b>para así</b> poder utilizarla</p> <p>Requisitos probados: RF-003, RF-032, RF-034, RNF-002, RNF-004</p> <p>Pruebas que realizar: Se realiza el registro de un usuario introduciendo su dni, nombre, apellidos, dirección, e-mail y contraseña. Registro permitido y almacenado en la tabla usuario.</p>
3	<p><b>Objetivo probado: Registro de empleado no registrado</b>  <b>como</b> empleado  <b>quiero</b> registrarme en la plataforma  <b>para así</b> poder utilizarla</p> <p>Requisitos probados: RF-004, RF-032, RF-034, RNF-002, RNF-004</p> <p>Pruebas que realizar: Se realiza el registro de un empleado introduciendo su dni, nombre, apellidos, dirección, e-mail y contraseña. Registro permitido y almacenados en la tabla empleado</p>
4	<p><b>Objetivo probado: Autenticado de un usuario - empleado - administrador</b>  <b>como</b> usuario o empleado  <b>quiero</b> acceder a la aplicación  <b>para así</b> poder utilizarla</p> <p>Requisitos probados: RF-005, RF-017, RF-009, RF-032, RF-033, RNF-002, RNF-004</p> <p>Pruebas que realizar: Se introduce el e-mail y la contraseña en el formulario Log In. Autenticación permitida</p>
5	<p><b>Objetivo probado: Autenticado de un usuario - empleado - administrador</b>  <b>como</b> usuario o empleado  <b>quiero</b> acceder a la aplicación  <b>para así</b> poder utilizarla</p> <p>Requisitos probados: RF-005, RF-009, RF-017, RF-032, RF-033, RNF-004</p> <p>Pruebas que realizar: Se hace clic en el botón submit sin haber introducido el e-mail y/o la contraseña. Autenticación no permitida</p>
6	<p><b>Objetivo probado: Pedir cita</b>  <b>como</b> usuario  <b>quiero</b> pedir una cita  <b>para así</b> poder ser atendido para un servicio de asesoría jurídica</p> <p>Requisitos probados: RF-006, RF-008, RF-007, RNF-002, RNF-004</p> <p>Pruebas que realizar: Se accede tras loguearse a UserAdvisoryHome y se hace clic en la opción pedir cita y se accede a la vista de formulario de solicitud de cita, donde se cumplimenta fecha y hora, servicio y tipo de cita. Solicitud permitida.</p>
7	<p><b>Objetivo probado: Consultar citas</b>  <b>como</b> empleado y como administrador  <b>quiero</b> consultar el listado de citas  <b>para así</b> poder organizar el volumen de trabajo y horarios presenciales en el despacho</p> <p>Requisitos probados: RF-011, RF-012, RF-018, RF-019, RF-031, RF-032, RNF-002, RNF-004</p> <p>Pruebas que realizar: Se hace clic en la opción de listar las citas. Acceso permitido.</p>

8	<p><b>Objetivo probado: Modificar cita</b>  <b>como</b> empleado y como administrador  <b>quiero</b> modificar una cita  <b>para así</b> ajustarla a la realidad actual</p> <p>Requisitos probados: RF-013, RF-018, RF-020, RF-032, RNF-002, RNF-004  Pruebas que realizar: Dentro del listado de citas, se hace clic en el botón "actualizar" junto al registro a modificar. La aplicación redirige a vista con los datos del registro en el interior de los cuadros de texto y editables. Se modifica el campo a modificar y se pulsa el submit. Modificación permitida.</p>
9	<p><b>Objetivo probado: Eliminar cita</b>  <b>como</b> empleado y como administrador  <b>quiero</b> eliminar una cita  <b>para así</b> eliminar el registro de la cita</p> <p>Requisitos probados: RF-021, RF-022, RNF-002, RNF-004  Pruebas que realizar: Dentro del listado de citas, se hace clic en el botón "Borrar" junto al registro a eliminar. La aplicación elimina el registro de la cita en la base de datos appointment. Eliminación permitida.</p>
10	<p><b>Objetivo probado: Consultar usuarios</b>  <b>como</b> administrador  <b>quiero</b> consultar el listado de usuarios  <b>para así</b> poder controlar y gestionar la cartera de clientes de la asesoría.</p> <p>Requisitos probados: RF-023, RF-026, RF-031, RF-032, RNF-002, RNF-004</p> <p>Pruebas que realizar: Se hace clic en la opción de listar los usuarios. Acceso permitido.</p>
11	<p><b>Objetivo probado: Modificar usuarios</b>  <b>como</b> administrador  <b>quiero</b> modificar los datos de un usuario  <b>para así</b> disponer de unos datos de los usuarios actualizados</p> <p>Requisitos probados: RF-024, RNF-002, RNF-004</p> <p>Pruebas que realizar: Dentro del listado de usuarios, se hace clic en el botón "actualizar" junto al registro a modificar. La aplicación redirige a vista formulario de modificación con los cuadros de texto activos para la edición. Se modifica el/los campos a modificar y se hace clic en el botón submit. Modificación permitida.</p>
12	<p><b>Objetivo probado: Eliminar usuario</b>  <b>como</b> administrador  <b>quiero</b> eliminar un usuario  <b>para así</b> eliminar el registro de un usuario</p> <p>Requisitos probados: RF-025, RF-032, RNF-002, RNF-004  Pruebas que realizar: Dentro del listado de usuarios, se hace clic en el botón "Borrar" junto al registro a eliminar. La aplicación elimina el registro del usuario en la base de datos user. Eliminación permitida.</p>

13	<p><b>Objetivo probado: Consultar empleados</b> <b>como</b> administrador <b>quiero</b> consultar el listado de empleados <b>para así</b> poder controlar y gestionar el equipo de empleados que forman parte de la asesoría.</p> <p>Requisitos probados: RF-027, RF-030, RF-031, RF-032, RNF-002, RNF-004</p> <p>Pruebas que realizar: Se hace clic en la opción de listar los empleados. Acceso permitido.</p>
14	<p><b>Objetivo probado: Modificar empleado</b> <b>como</b> administrador <b>quiero</b> modificar los datos de un empleado <b>para así</b> disponer de unos datos de los empleados actualizados</p> <p>Requisitos probados: RF-028, RF-032, RNF-002, RNF-004</p> <p>Pruebas que realizar: Dentro del listado de empleados, se hace clic en el botón "actualizar" junto al registro a modificar. La aplicación redirige a vista formulario de modificación con los cuadros de texto activos para la edición. Se modifica el/los campos a modificar y se hace clic en el botón submit. Modificación permitida.</p>
15	<p><b>Objetivo probado: Eliminar empleado</b> <b>como</b> administrador <b>quiero</b> eliminar un empleado <b>para así</b> eliminar el registro de un empleado</p> <p>Requisitos probados: RF-029, RF-032, RNF-002, RNF-004</p> <p>Pruebas que realizar: Dentro del listado de empleados, se hace clic en el botón "Borrar" junto al registro a eliminar. La aplicación elimina el registro del empleado en la base de datos employee. Eliminación permitida.</p>

## 8. CONCLUSIONES

Durante la fase de preanálisis, me propuse una serie de objetivos a alcanzar, que finalmente he alcanzado casi en su totalidad, y que detallo en el siguiente punto.

### 8.1 OBJETIVOS ALCANZADOS

Objetivos Generales alcanzados (todos):

- visibilizar AdvisoryLLC .....**OBJETIVO ALCANZADO**
- desarrollar un sistema que facilite a AdvisoryLLC la gestión de sus citas, usuarios, y empleados..... **OBJETIVO ALCANZADO**

Objetivos Específicos alcanzados. (todos) Que la aplicación desarrollada permita:

- que los usuarios y los empleados puedan crear una cuenta introduciendo sus datos en el sistema..... **OBJETIVO ALCANZADO.**
- que tenga una vista de escritorio, a la que usuario, empleado y administrador (una vez registrados) puedan conectarse a través de un navegador web..... **OBJETIVO ALCANZADO.**
- que los usuarios puedan reservar citas tanto telemáticas como presenciales para los servicios jurídicos que necesiten..... **OBJETIVO ALCANZADO**

- que los empleados y el administrador puedan visibilizar las citas reservadas por los clientes, modificarlas o eliminarlas..... **OBJETIVO ALCANZADO.**
- que el administrador pueda visibilizar a los clientes de la empresa, modificar sus datos personales, buscar por dni a uno de ellos, o en última instancia, pueda eliminar un cliente..... **OBJETIVO ALCANZADO.**
- que el administrador pueda visibilizar a los empleados de la empresa, así como modificar sus datos y también su acceso a la interfaz de empleado, hacer búsqueda por dni de uno de ellos, o en última instancia, eliminarlo..... **OBJETIVO ALCANZADO.**

La falta de tiempo ha impedido que alcanzase un objetivo que, aunque no lo incluí por previsión a que finalmente no pudiera alcanzarlo, sí que me lo había propuesto si la disponibilidad de tiempo me lo permitía, y que detallo:

Personalmente me marqué como objetivo que la aplicación web fuese totalmente responsive, utilizando media queries para el diseño mobile first, de forma que se pudiera utilizar también desde dispositivos móviles sin problema, pero en la fase final, he comprobado que aunque tanto el header como el nav, y el body que son totalmente responsives, no acaba de quedar como yo quisiera el footer en el simulador de chrome de pantalla mobile, de forma que no puedo estar satisfecha con ese objetivo que aunque no fue incluido como tal. Lo incluiré en vías futuras para resolverlo cuando finalice DAW y disponga de tiempo para modificarlo adecuadamente.

## 8.2 CONCLUSIONES DEL PROYECTO REALIZADO

Este proyecto ha supuesto un gran reto personal para mí, dado que nunca había realizado una aplicación completa de tal envergadura, marcando desde inicio los objetivos a alcanzar, analizando requisitos funcionales y no funcionales, los diferentes casos de uso, realizando los diagramas E-R y paso a relacional y creación de la base de datos completa con tablas relacionadas, y documentándolo todo. Inicialmente tuve mis dudas sobre si podría o no alcanzar la mayoría al menos de los objetivos, pero estoy muy satisfecha de que paso a paso, haya podido alcanzar casi la totalidad de ellos.

Inicialmente me resultaba extraño no comenzar a codificar directamente, teniendo que realizar la fase de preanálisis y análisis antes de comenzar a implementar el código, pero después he comprendido el por qué, y he agradecido todo el trabajo previo durante la fase de implementación, dado que me ha evitado muchos errores y modificaciones.

También ha supuesto un reto para mí codificar en lenguaje PHP utilizando POO con la librería PDO y he tenido que documentarme.

Y valorando también las dificultades externas al proyecto, he tenido que salvar muchas dificultades a nivel personal también. El hecho de trabajar a tiempo completo y de ser madre, me ha obligado a

optimizar mi tiempo al máximo durante las noches, y algunos días, incluso restándome horas de sueño para poder llegar a tiempo a las entregas del proyecto.

Pero pese a todas las dificultades, la realización de este proyecto ha sido muy gratificante para mí por el muchas razones. El hecho de trabajar a nivel global un proyecto completo ha hecho que se fortaleciesen los conocimientos teóricos que tenía de diferentes módulos al tener que implementarlos, y también he adquirido otros conocimientos que no tenía de diferentes áreas, maquetación, POO, diagramas de casos de uso, las pruebas de caja negra, etc. Y lo más importante para mí, considero que he ampliado durante el desarrollo de este proyecto mi capacidad de búsqueda de información en internet.

## 9. VÍAS FUTURAS

A nivel comercial, esta aplicación podría ayudar en la gestión de cualquier empresa de servicios que tenga que gestionar citas, usuarios y empleados. La funcionalidad crud es básica, pero teniendo en cuenta el tiempo que he tenido para su realización, la verdad es que me sorprende haber podido lograr todos los objetivos marcados al inicio del proyecto.

Mejoraré la vista mobile para que sea totalmente responsive incluso el footer cuando finalice DAW y disponga de tiempo para modificarlo adecuadamente porque no ha quedado como hubiera querido.

Una de la posibles vías futuras consistiría en intentar formar una comunidad online, creando una zona de posts donde la asesoría pueda ir informando de las diferentes campañas anuales de liquidación de impuestos, o de temas que puedan ser de interés de sus clientes en la página de inicio.

También se podría añadir, dentro de la vista UserAdvisoryHome, una funcionalidad para que el usuario pudiera realizar una consulta jurídica a AdvisoryLLC, implementándose un formulario con su identificación y la consulta a realizar, para que desde la asesoría se le pudiera resolver u orientar. Esta consulta podría ser gratuita o por sistema de ticketing en función de la dificultad a resolver.

Por otro lado, las cuentas en redes sociales no las incluí en los objetivos marcados y no las he activado para el proyecto, pero he dejado implementados los iconos y enlaces para simplemente cambiar su ruta de la actual (up en la misma vista) a la del perfil, una vez los active.

El Despliegue de la aplicación lo he realizado en local, con el paquete de software XAMPP, que contiene el servidor web HTTP Apache2 que escucha en el puerto 80 nuestra conexión HTTP como he indicado, pero cuando contratemos hosting (y añadiremos el SSL en esta contratación), lo modificaremos para que redirija ésta al puerto 443.

Algunas pruebas se han de realizar con el cliente, por lo que las incluyo en vías futuras:

**RNF-004:** El User|Employee|Admin deberá tener acceso a internet.



**RNF-005:** La página web estará disponible 24/7 a través de internet, por lo que se alojará en un hosting, que deberá ser 24/7 y ofrecernos un mantenimiento también de 24/7 por si se diera algún problema.

**RNF-006:** El usuario requiere tener acceso a internet desde cualquier dispositivo.

**RNF-007:** el dominio para alojar la página web, se contratará añadiendo certificado SSL (Secure Sockets Layer), para cifrar los datos compartidos entre servidor y User|Employee|Admin, mediante el protocolo **SSL/TLS**, lo cual aportará seguridad a la transferencia de archivos y datos durante las interacciones con el sistema por parte de User|Employee|Admin.

**RNF-008:** el dominio deberá efectuar backups automáticos diarios y deberá poder facilitar una fácil restauración en caso de fallo del sistema.

## 10. GLOSARIO

**Cancelación económica de hipoteca:** Cancelación del préstamo hipotecario mediante el pago adelantado del mismo. El banco expedirá el certificado de saldo cero a tal efecto.

**Certificado de saldo cero:** Documento que acredita que un préstamo hipotecario ha sido cancelado económicamente.

**Carta de Pago:** Escritura que otorga una entidad bancaria una vez dispone de certificado de saldo cero, imprescindible para la cancelación registral de una hipoteca que financió y que se encuentra cancelada económicamente pero no registralmente.

**Cancelación registral de hipoteca:** Cancelación en el registro de la propiedad, de la anotación de una hipoteca en una finca registral.

**Modelo 600:** Impuesto a liquidar durante la tramitación registral, tras la retirada de notaría y previo a la inscripción registral.

**Cancelación de embargo:** Cancelación de nota preventiva de embargo que grava una finca.

**Cancelación de Condición resolutoria:** Cancelación de la garantía de precio aplazado mediante letras de cambio.

**UserAdvisoryHome:** Zona de la aplicación a la que accederán los usuarios registrados y logueados, así como los empleados recién registrados que tienen por defecto perfil 0.

**EmployeeAdvisoryHome:** Zona de la aplicación a la que accederán los empleados registrados y logueados, una vez ya han sido revisados por el admin del sistema y modificado su perfil a 1.

**AdminAdvisoryHome:** Zona de la aplicación a la que accederá solamente el empleado con perfil 2, es decir, el administrador.

**Pantalla Advisoryer Home:** Zona de la aplicación a la que la aplicación dirige tras el login, y que será AdminAdvisoryHome | EmployeeAdvisoryHome | UserAdvisoryHome en función del rol, y tendrá unas determinadas funcionalidades según este rol.

## 11. BIBLIOGRAFÍA / WEBGRAFÍA

He necesitado consultar algunas fuentes para distintas cuestiones a lo largo de la realización de este proyecto, como por ejemplo, para algunos puntos de la realización de la memoria:

Fases del desarrollo, Ciclo de vida, Diagrama de Casos:

*Material didáctico Entornos de desarrollo. (2021). Ilerna Online.*

[https://campus.ilerna.es/courses/2961/pages/material-didactico-en-formato-interactivo?module\\_item\\_id=237143](https://campus.ilerna.es/courses/2961/pages/material-didactico-en-formato-interactivo?module_item_id=237143)

Creación de la BBDD y realización modelo Entidad Relación y paso a Relacional:

*Material didáctico Bases de datos A. (2021). Ilerna Online.*

<https://campus.ilerna.es/courses/1578>

metodología Lean:

*Metodología Lean. (2021). kanbantool.com.*

<https://kanbantool.com/es/guia-kanban/metodologia-lean>

Diagramas UML:

*Diagramas UML. (2020, mayo). wikipedia.org/wiki.*

[https://es.wikipedia.org/wiki/Lenguaje\\_unificado\\_de\\_modelado](https://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado)

Herramienta online para la realización del Diagrama de casos de uso y Diagrama de clases:

*Programa para hacer diagramas online. (2022). Lucidchart.*

<https://www.lucidchart.com/>

También he usado varias fuentes para el maquetado de la aplicación:

*Colorzilla.com(2021)*

<https://www.colorzilla.com/gradient-editor>

*Google Fonts. (2021). https://fonts.google.com.*

<https://fonts.google.com>

*Stock de imágenes Imágenes libres. (2021). Pexels.*

<https://www.pexels.com/es-es/>

Y por último, he tenido que consultar diversas fuentes para formarme en algunos aspectos de la codificación, como por ejemplo:

*Documentación de PHP*

<https://www.php.net/>

J.D. (2015). *Programación PHP orientada a objetos. Píldoras informáticas.*

<https://www.youtube.com/watch?v=l75CUdSJfw&list=PLU8oAIHdN5BkinrODGXTToK9oPAInJxmW>

J.D. (2015). *Consultas preparadas para evitar la Inyección SQL. Píldoras informáticas.*

<https://www.youtube.com/watch?v=l75CUdSJfw&list=PLU8oAIHdN5BkinrODGXTToK9oPAInJxmW>

## 12. ANEXOS

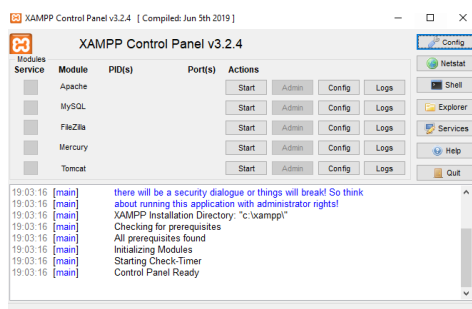
### 12.1 ANEXO I: MANUAL DE INSTALACIÓN

Detallo el manual en local, dado que no he contratado ningún hosting para el proyecto, y lo he desplegado de forma local.

Requisitos: Se ha de disponer del paquete XAMPP (o equivalente) instalado, que contiene como he indicado del servidor web Apache, el intérprete del lenguaje PHP, un SGBD MySQL y también phpMyAdmin. Es posible instalarlos por separado, pero es más sencillo usar el paquete XAMPP que los engloba y tiene un panel de control muy intuitivo.

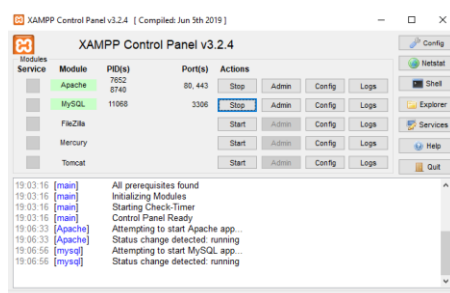
Pasos:

#### 1. Abrir XAMPP



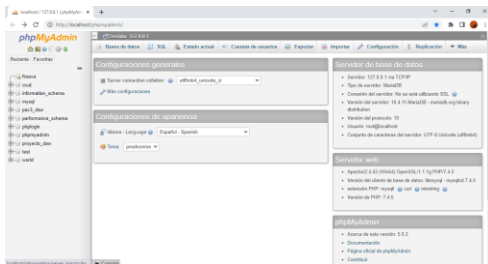
Anexo I-a: Panel de control del paquete XAMPP antes de iniciar servicios.- Fuente: Elaboración propia.

#### 2. Iniciar los servicios Apache y MySQL haciendo clic en "Start"



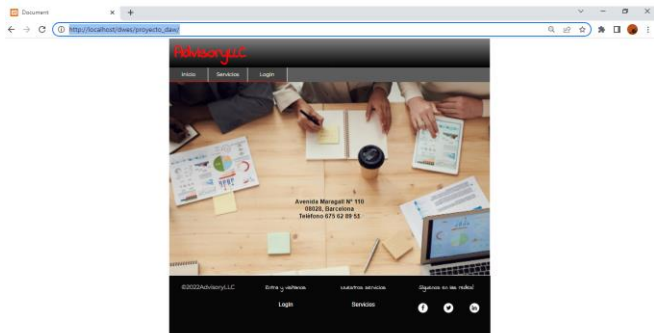
Anexo I-b: Panel de control del paquete XAMPP tras iniciar servicios Apache y MySQL.- Fuente: Elaboración propia.

3. Descargar la carpeta .zip con la entrega realizada del proyecto,
4. Descomprimirla,
5. Arrastrar o copiar la carpeta del código fuente completa, y pegarla en C:\xampp\htdocs\
6. Hacer clic en "Admin" de MySQL para que abra PhpMyAdmin,



*Anexo I-c: Captura de la interface de phpMyAdmin donde haremos clic en Importar para la importación del archivo .sql con la base de datos del proyecto.- Fuente: Elaboración propia.*

7. Hacer clic en "Importar"
8. Seleccionar el archivo .sql con la BBDD creada
9. Escribir en el navegador `http://localhost/nombre_de_la_carpeta`
10. Y tendremos acceso a la aplicación en el navegador



*Anexo I-d: Captura de la vista del archivo index.php en local (localhost).- Fuente: Elaboración propia.*

## 12.2 ANEXO II: MANUAL DE USUARIO y CREDENCIALES ADMIN

Se puede navegar por las vistas INICIO, SERVICIOS, LOGIN y FORMULARIO DE REGISTRO sin estar autenticado.

Pero para usar las funcionalidades de la aplicación es necesario autenticarse introduciendo **e-mail** y **password** en el formulario **Login**, y se ha de estar registrado.

**Si no** se está registrado, se ha de **cumplimentar el formulario de registro** indicando si se es usuario, o empleado. Tras esto, la aplicación redirigirá al Login para poder autenticarse.

### CREDENCIALES PARA PRUEBAS

Para trabajar con el perfil de admin, es necesario loguearse con usuario "tomas@gmail.com" y contraseña "tomas1234"

Las credenciales de los diferentes actores son:

**ADMIN:** Está registrado con login "tomas@gmail.com" y password "tomas1234"

**EMPLEADO:** Está registrado con login "victor@gmail.com" y password "victor1234"

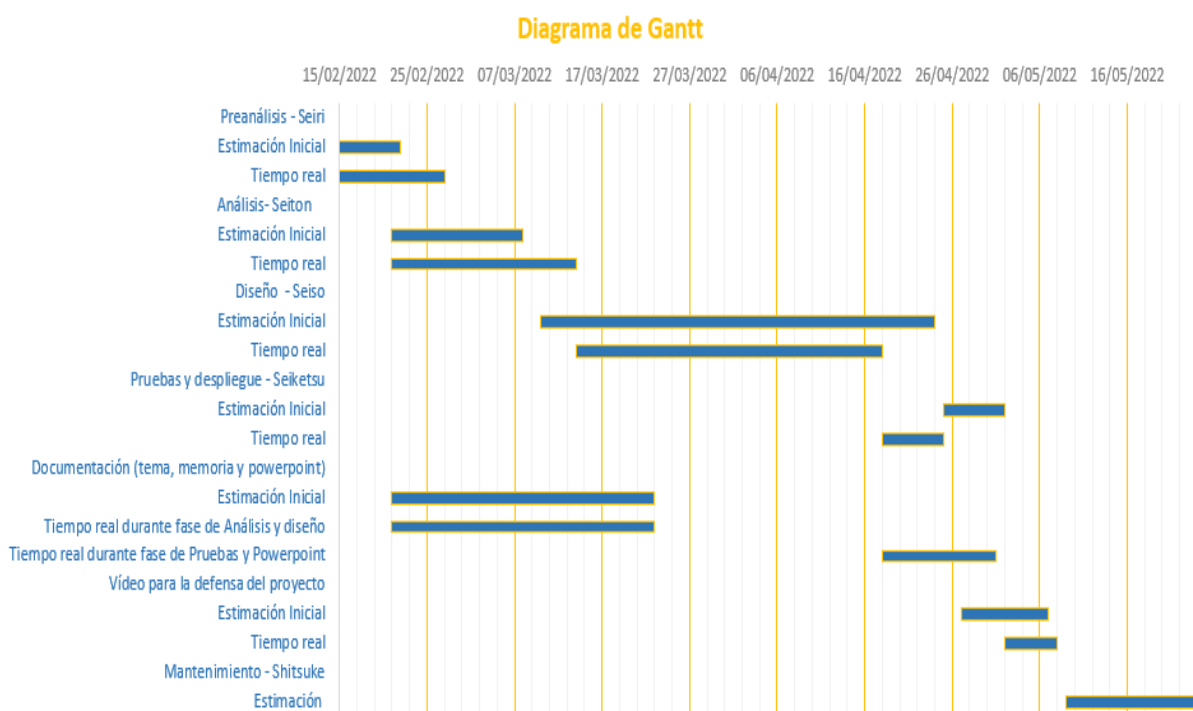
**USUARIO:** Está registrado con login "ana@gmail.com" y password "ana1234"

### 12.3 ANEXO III: Tabla previa al Diagrama de Gantt

Item Proyecto Final	Fecha inicio	Duración	Fecha finalización
<b>Preanálisis - Seiri</b>			
Estimación Inicial	15/02/2022	7	22/02/2022
Tiempo real	15/02/2022	12	27/02/2022
<b>Análisis- Seiton</b>			
Estimación Inicial	21/02/2022	15	08/03/2022
Tiempo real	21/02/2022	21	14/03/2022
<b>Diseño - Seiso</b>			
Estimación Inicial	10/03/2022	45	24/04/2022
Tiempo real	14/03/2022	35	18/04/2022
<b>Pruebas y despliegue - Seiketsu</b>			
Estimación Inicial	25/04/2022	7	02/05/2022
Tiempo real	18/04/2022	7	25/04/2022
<b>Documentación (tema, memoria y powerpoint)</b>			
Estimación Inicial	21/02/2022	30	23/03/2022
Tiempo real durante fase de Análisis y diseño	21/02/2022	30	23/03/2022
Tiempo real durante fase de Pruebas y Powerpoint	18/04/2022	13	01/05/2022
<b>Vídeo para la defensa del proyecto</b>			
Estimación Inicial	27/04/2022	10	07/05/2022
Tiempo real	02/05/2022	6	08/05/2022
<b>Mantenimiento - Shitsuke</b>			
Estimación	09/05/2022	15	24/05/2022

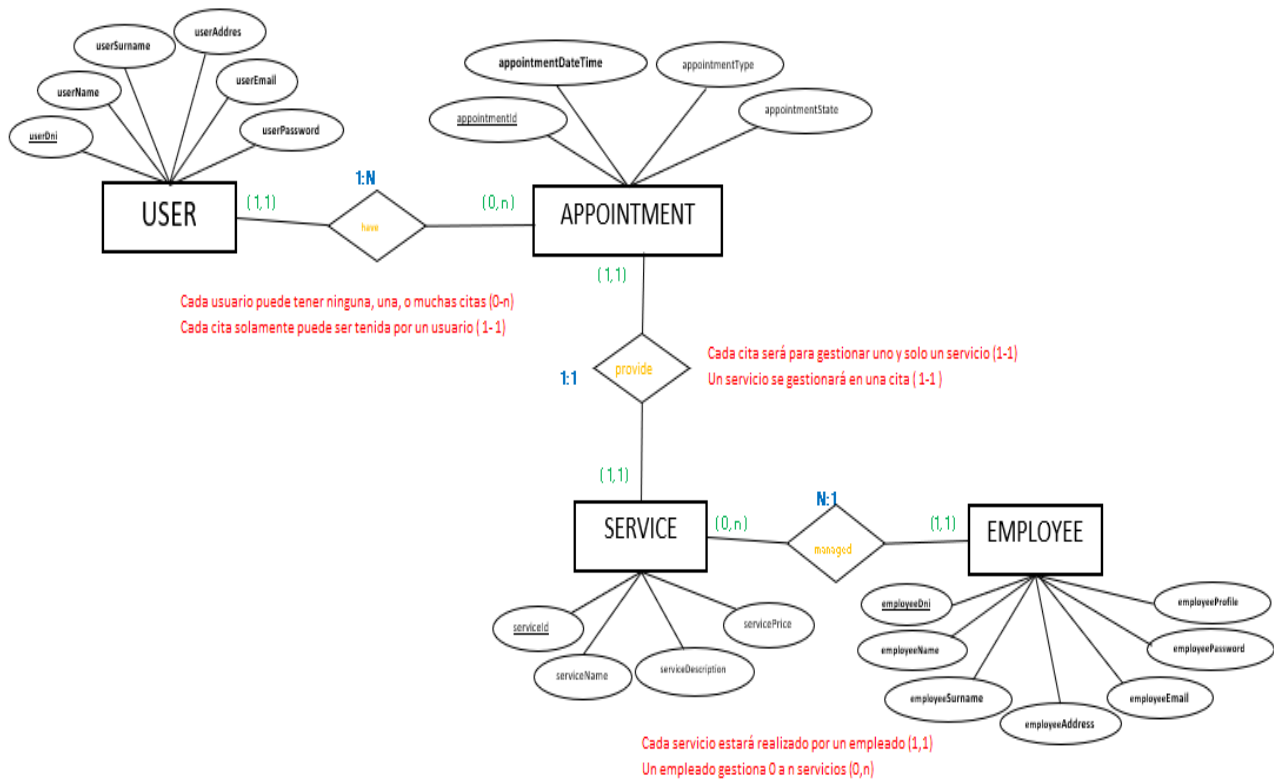
Anexo III: Excel previo a la elaboración del Diagrama de Gantt.- Fuente: Elaboración propia.

### 12.4. ANEXO IV: Diagrama de Gantt



Anexo IV: Diagrama de Gantt.- Fuente: Elaboración propia.

## 12.5 ANEXO V: Modelo Entidad-Relación



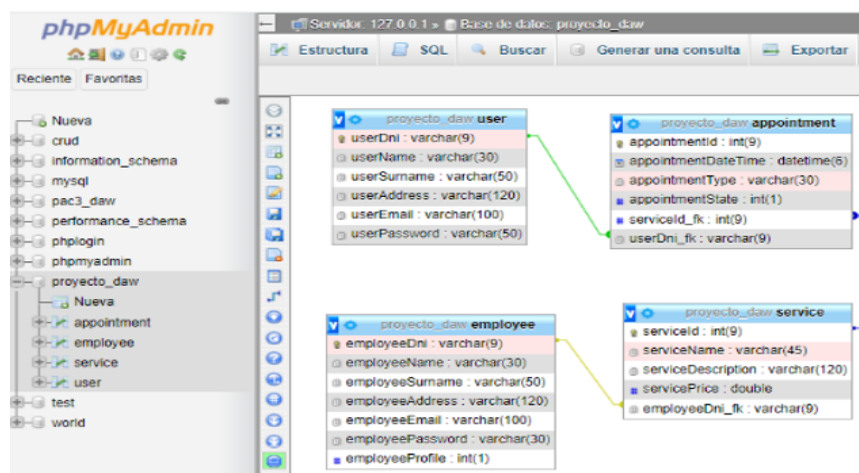
Anexo V: Modelo Entidad - Relación para la creación de la base de datos proyecto\_daw- Fuente: Elaboración propia.

## 12.6 ANEXO VI: MODELO RELACIONAL

user(**userDni**, userName, userSurname, userAddress, userEmail, userPassword)  
 appointment(**appointmentId**, appointmentDateTime, appointmentType, appointmentState, **serviceId\_fk**, **userDni\_fk**)  
 service(**serviceId**, serviceName, serviceDescription, servicePrice, **employeeDni\_fk**)  
 employee(**employeeDni**, employeeName, employeeSurname, employeeAddress, employeeEmail, employeePassword, employeeProfile)

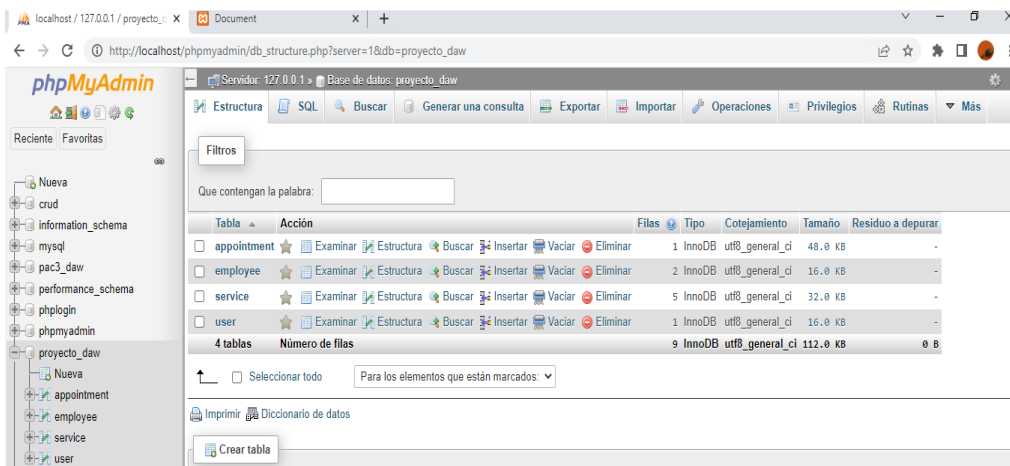
Anexo VI: Modelo Relacional para la creación de la base de datos proyecto\_daw- Fuente: Elaboración propia.

## 12.7. ANEXO VII: MODELO FÍSICO



Anexo VII: Modelo Físico base de datos proyecto\_daw- Fuente: Elaboración propia.

## 12.8. ANEXO VIII: ESTRUCTURA BASE DE DATOS proyecto\_daw

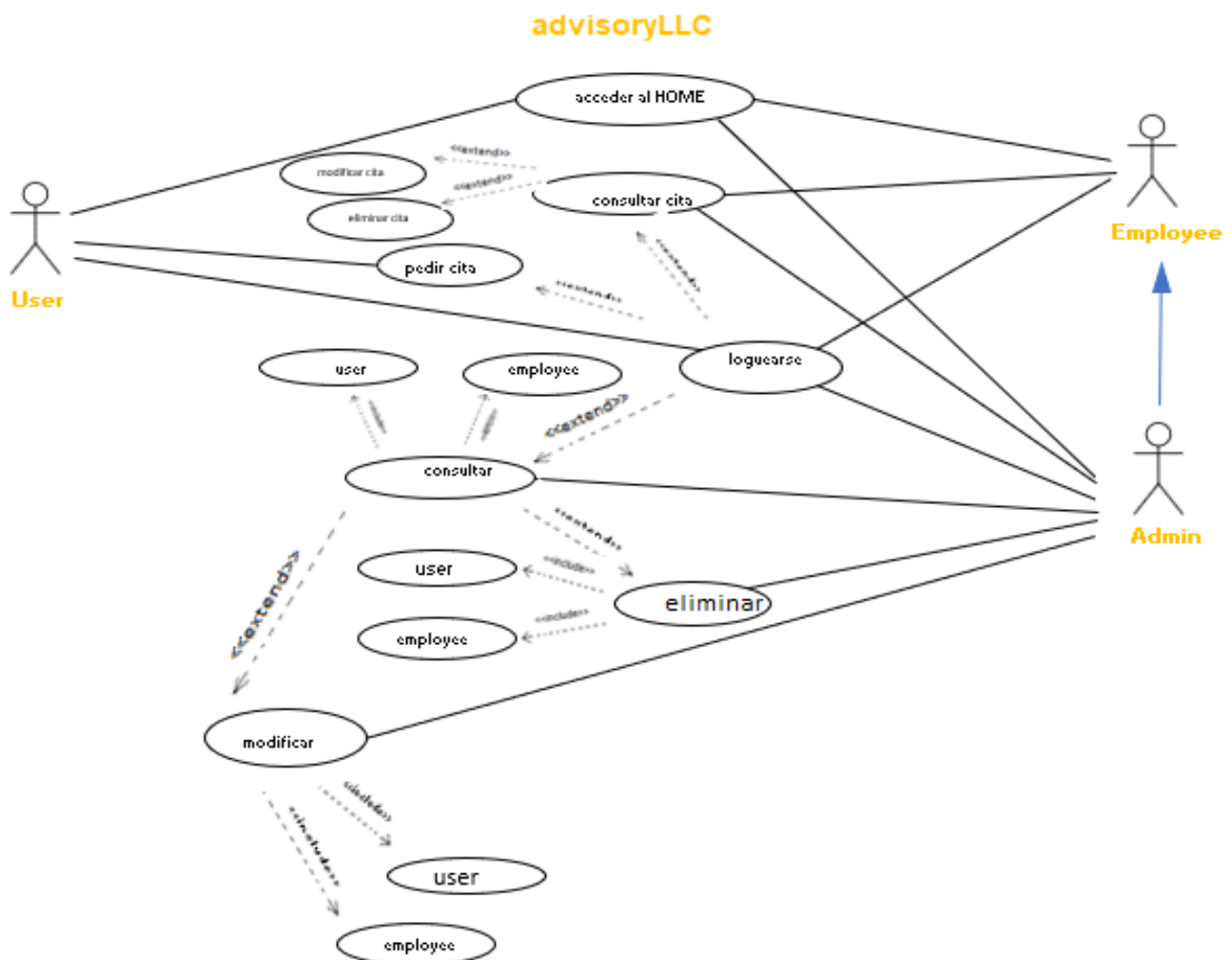


The screenshot shows the phpMyAdmin interface for the 'proyecto\_daw' database. The left sidebar lists various databases and tables. The main area displays the structure of the 'proyecto\_daw' database, showing four tables: 'appointment', 'employee', 'service', and 'user'. Each table entry includes icons for actions like 'Examinar', 'Estructura', 'Buscar', 'Insertar', 'Vaciar', and 'Eliminar'. Below the table list, there is a summary row indicating 4 tables and a total of 9 rows.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
appointment	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	48.0 KB	-
employee	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_ci	16.0 KB	-
service	Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8_general_ci	32.0 KB	-
user	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	16.0 KB	-
<b>4 tablas</b>	<b>Número de filas</b>	<b>9</b>	<b>InnoDB</b>	<b>utf8_general_ci</b>	<b>112.0 KB</b>	<b>0 B</b>

Anexo VIII: Captura de la estructura de la base de datos proyecto\_daw - Fuente: Elaboración propia.

## 12.9. ANEXO IX: DIAGRAMA DE CASOS DE USO



Anexo IX: Modelo de Casos de Uso para el sistema advisoryLLC - Fuente: Elaboración propia

## 12.10. ANEXO X: MAPA DE NAVEGACIÓN DE LA APLICACIÓN Y DISEÑO ESBOZO INICIAL

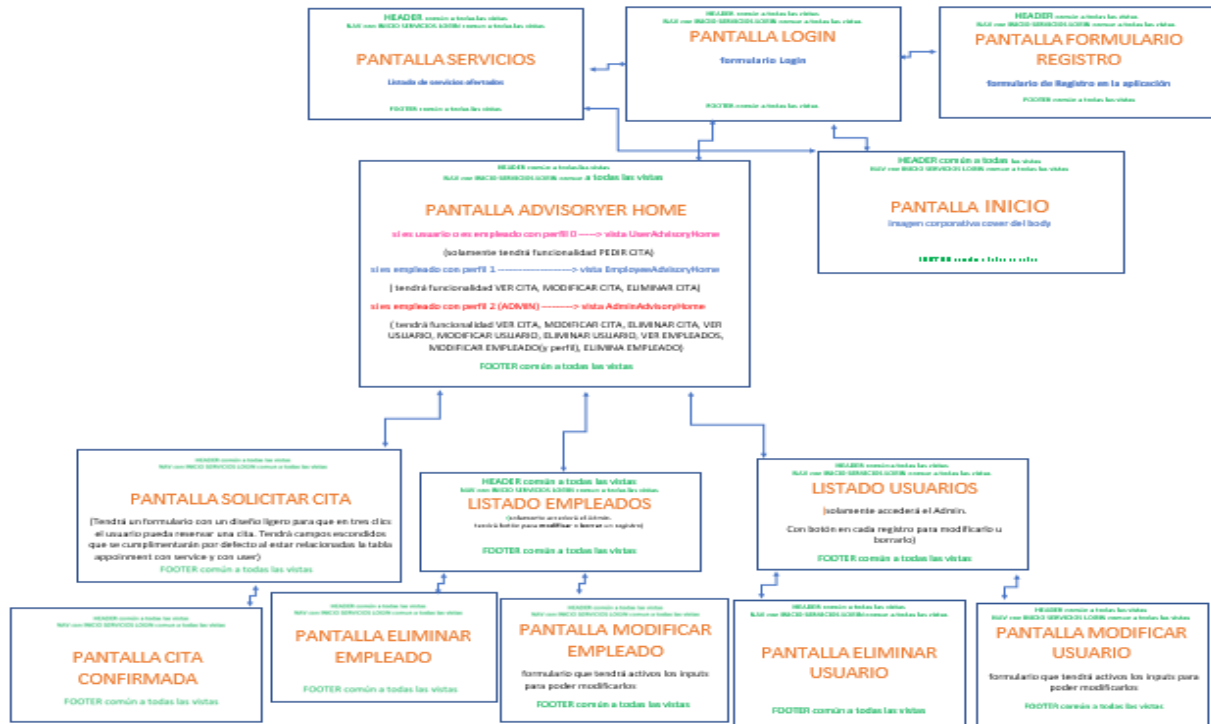
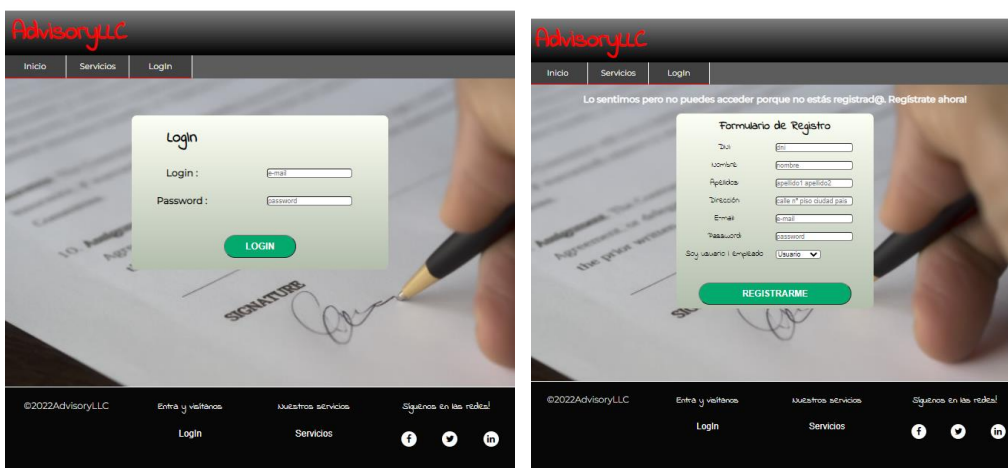


Figura X: Diseño orientativo y mapa de navegación advisoryLLC - Fuente: Elaboración propia

## 12.11. ANEXO XI: VISTAS OBTENIDAS EN PRUEBAS REALIZADAS

### Prueba 1. La persona se intenta loguear con campo login y/o campo password vacíos

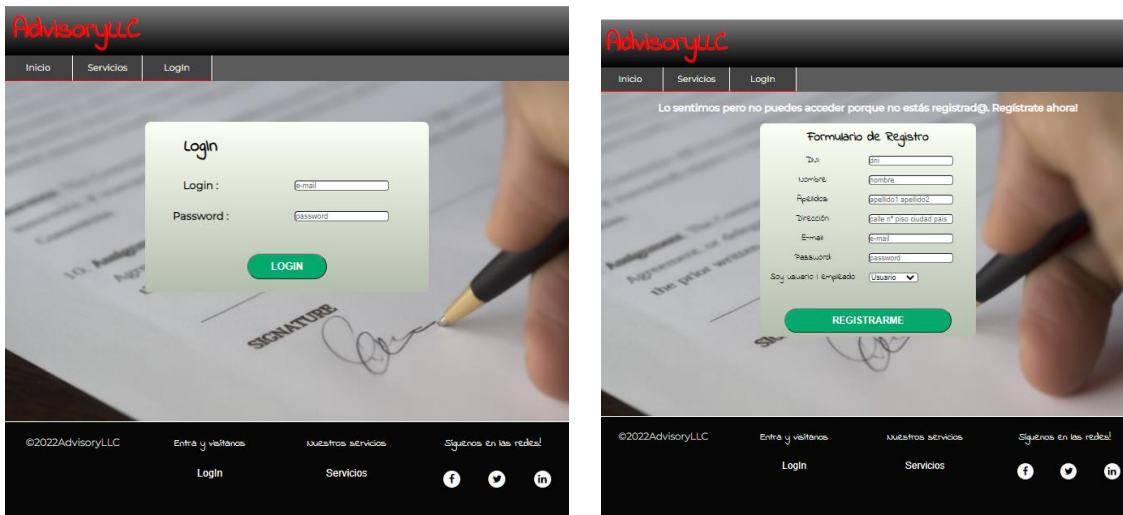
La persona hace clic sin haber introducido e-mail y/o contraseña. La aplicación la redirige al formulario de Registro porque entiende que no está registrada (desde el nav se puede volver al Login si fué un error al clicar sin introducir los datos)



Figuras XI prueba 1: La persona se intenta loguear con campo login y/o campo password vacíos - Fuente: Elaboración propia



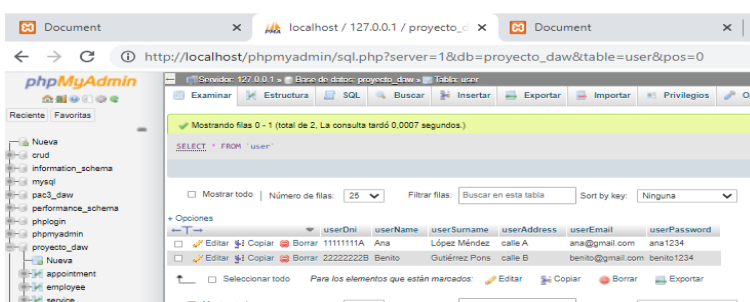
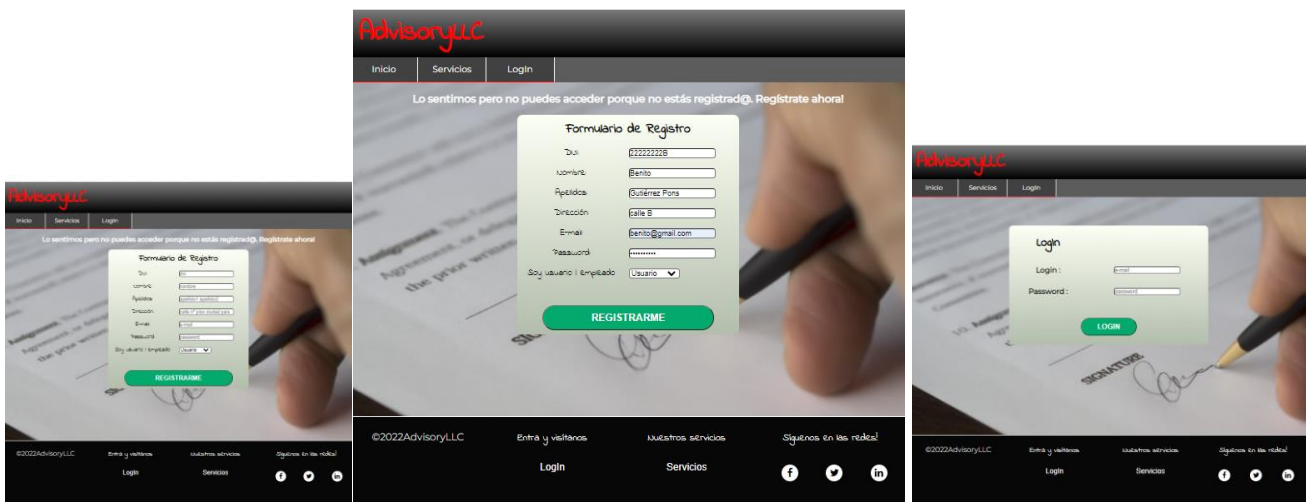
## Prueba 1bis. La persona se intenta loguear y no estaba registrada (mismo resultado)



Figuras XI prueba 1bis: La persona se intenta loguear y no estaba registrada (mismo resultado) - Fuente: Elaboración propia

## Prueba 2 La persona se registra en la plataforma indicando que es usuario (INSERT)

Cumplimenta el formulario de registro y una vez clique el botón REGISTRARME, se insertará el registro en la tabla user, dado que ha indicado en el desplegable que es usuario y no empleado, y será redirigido al formulario de Login para que pueda acceder a la plataforma.

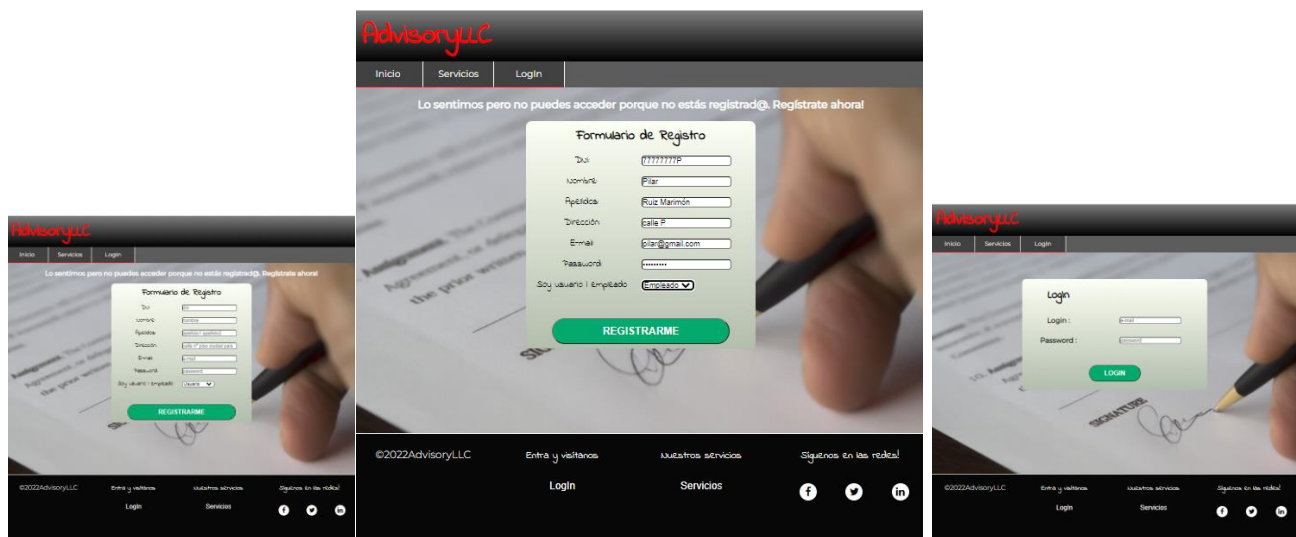


Figuras XI prueba 2: La persona se registra en la plataforma indicando que es usuario (INSERT) - Fuente: Elaboración propia

### Prueba 3. La persona se registra en la plataforma indicando que es empleado (INSERT)

Cumplimenta el formulario de registro y una vez clique el botón REGISTRARME, se insertará el registro en la tabla employee, dado que ha indicado en el desplegable que es empleado y no usuario, y será redirigido al formulario de Login para que pueda acceder a la plataforma.

Para evitar que cualquiera seleccione que es empleado y acceda a la plataforma de empleados, el formulario tiene el campo con el valor 0 para el atributo perfil, y, al autenticarse durante el Login, al tener perfil 0 será redirigido a la home de usuario, y no a la de empleado (provisionalmente hasta que el administrador revise es correcto el registro del nuevo empleado y le cambie el perfil al asignado al rol de empleado, el 1).



Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Usuarios roles

Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0008 segundos)

```
SELECT * FROM "employee"
```

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Sort by key: Ninguna

Opciones

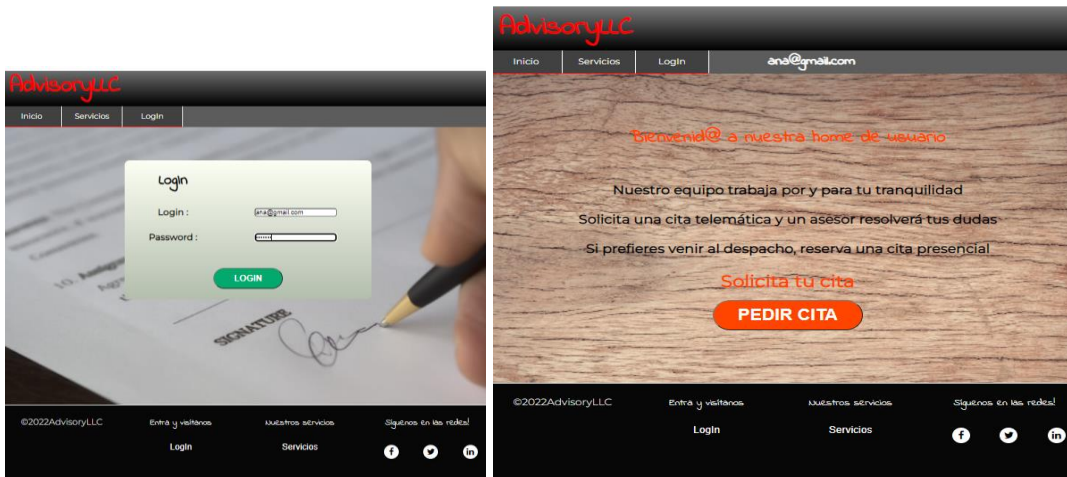
	employeeDni	employeeName	employeeSurname	employeeAddress	employeeEmail	employeePassword	employeeProfile
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	55555555T	Tomas	Lozano Mas	calle T	tomas@gmail.com	tomas1234	2
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	66666666V	Victor	Fernández Díaz	calle V	victor@gmail.com	victor1234	1
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	77777777P	Pilar	Ruiz Marimón	calle P	pilar@gmail.com	pilar1234	0

☐ Seleccionar todo | Para los elementos que están marcados: ☐ Editar ☐ Copiar ☐ Borrar ☐ Exportar

Figuras XI prueba 3: La persona se registra en la plataforma indicando que es empleado (INSERT) - Fuente: Elaboración propia

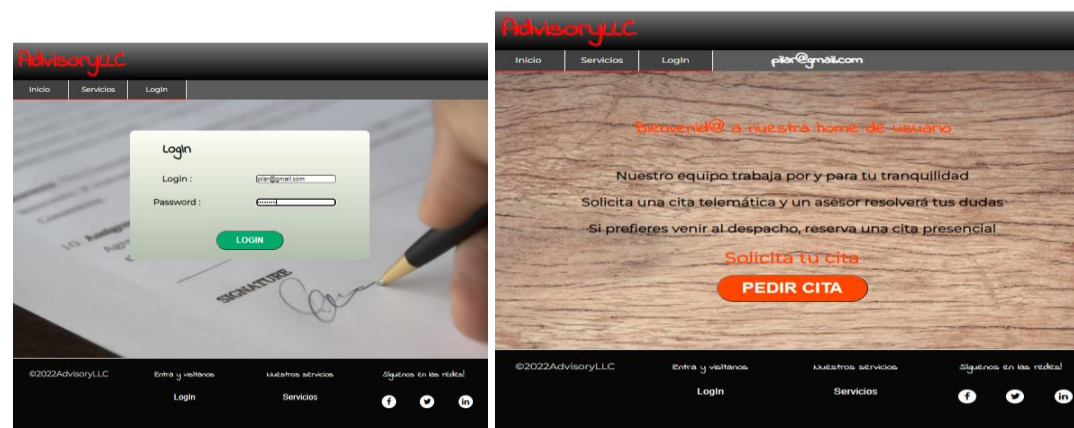
### Prueba 4a. La persona se loguea y es un usuario

La aplicación valida mediante una consulta que está registrada en la plataforma, en la tabla user, por lo que la redirige a la home de usuarios. Además, almacena en una variable de sesión el e-mail y lo imprime, para darle un toque más personalizado. El usuario solamente tiene la funcionalidad PEDIR CITA (funcionalidad INSERT del CRUD).



Figuras XI prueba 4a - usuario: La persona se loguea y es un usuario- Fuente: Elaboración propia

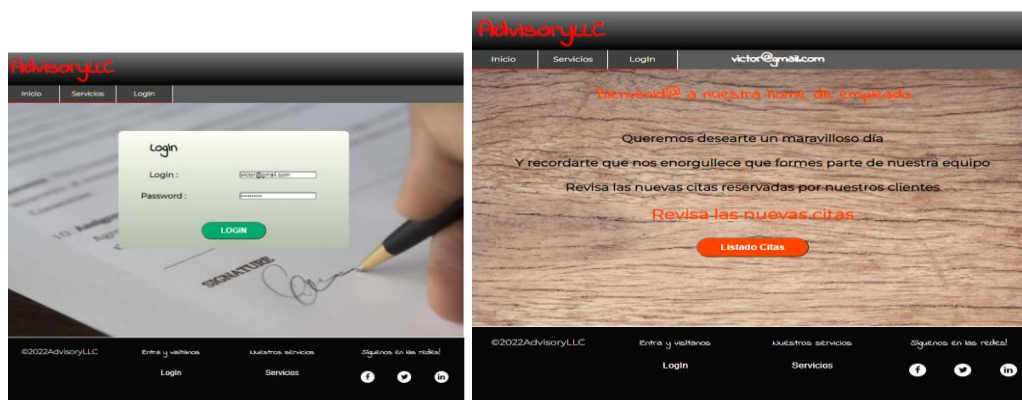
Prueba 4b. La persona se loguea y es un empleado recién registrado (con perfil 0) mismo resultado que prueba 4 para usuario



Figuras XI prueba 4b - empleado perfil 0: La persona se loguea y es un empleado recién registrado (con perfil 0) - Fuente: Elaboración propia

Como todavía tiene perfil 0, es redirigida a la home de usuario. Ha de esperar a que el administrador revise su registro y le habilite el perfil para disponer de acceso a la home de empleado.

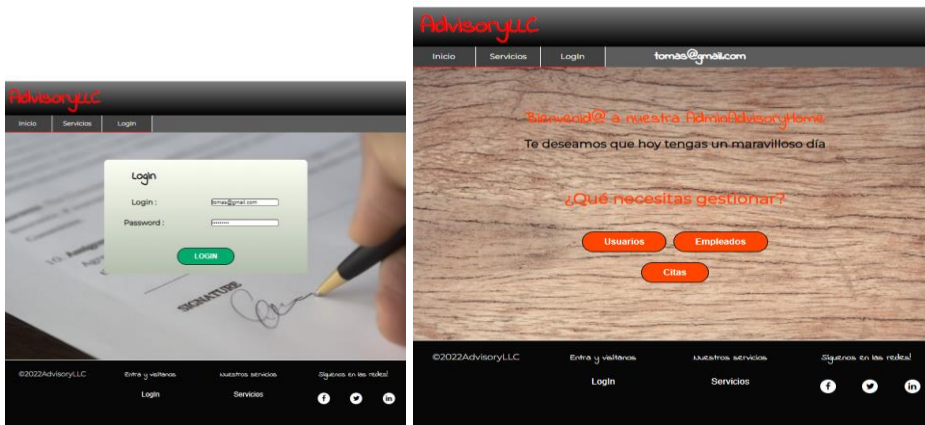
Prueba 4c La persona se loguea y es un empleado con perfil 1



Figuras XI prueba 4c - empleado perfil 1: La persona se loguea y es un empleado ya verificado por admin (con perfil 1) - Fuente: Elaboración propia



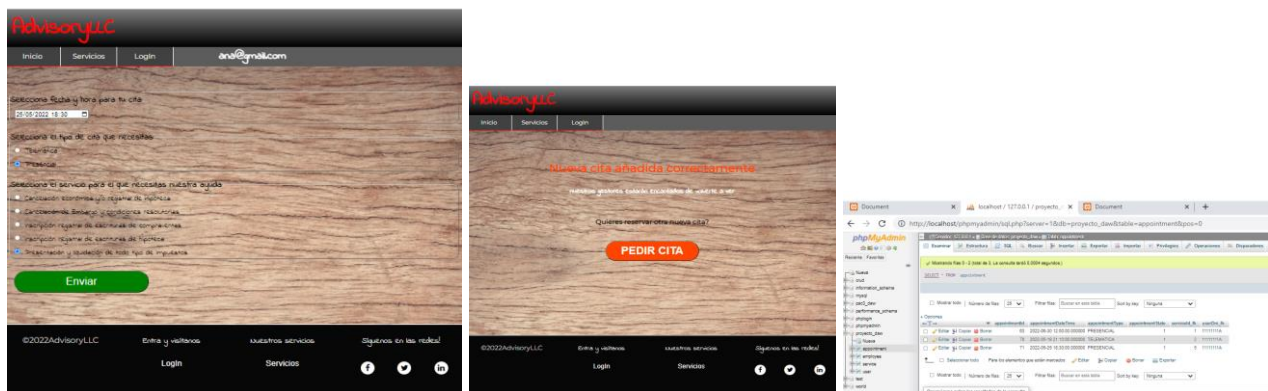
## Prueba 4d. La persona se loguea y es un empleado con perfil 2 (ADMINISTRADOR)



Figuras XI prueba 4d - empleado perfil 2: La persona se loguea y es un empleado es el admin - Fuente: Elaboración propia

## Prueba 6. El usuario solicita una cita (INSERT)

El usuario hace clic en el botón PEDIR CITA y es redirigido a la vista del formulario de solicitud de cita, que he intentado que no tuviese apariencia densa de formulario, y que en tres clics (he elegido inputs de tipo radio que sólo requieren de un clic) pudiera pedir una cita rápidamente.



Y el registro se ha insertado en la tabla appointment

Figuras XI prueba 6: El usuario solicita una cita (INSERT) - Fuente: Elaboración propia

Pruebas 7, 10 y 13 ADMINISTRADOR o EMPLEADO quieren ver las citas, y mismo resultado si ADMINISTRADOR quiere ver los usuarios o empleados (READ)

ID CITA	FECHA	TIPO	ESTADO	ID SERVICIO	DEL USUARIO	Acciones
75	22/05/2022 11:00h	PRESENCIAL	1	5	11111111A	<a href="#">Actualizar</a> <a href="#">Borrar</a>
77	23/05/2022 10:00h	TELEMÁTICA	1	1	11111111A	<a href="#">Actualizar</a> <a href="#">Borrar</a>
79	24/05/2022 10:00h	PRESENCIAL	1	4	22222222B	<a href="#">Actualizar</a> <a href="#">Borrar</a>
80	24/05/2022 10:15h	TELEMÁTICA	1	5	22222222B	<a href="#">Actualizar</a> <a href="#">Borrar</a>
85	16/05/2022 16:15h	PRESENCIAL	1	2	22222222B	<a href="#">Actualizar</a> <a href="#">Borrar</a>
87	25/05/2022 16:30h	TELEMÁTICA	1	4	11111111A	<a href="#">Actualizar</a> <a href="#">Borrar</a>
88	06/07/2022 16:30h	TELEMÁTICA	1	1	11111111A	<a href="#">Actualizar</a> <a href="#">Borrar</a>
89	16/05/2022 11:00h	PRESENCIAL	1	1	11111111A	<a href="#">Actualizar</a> <a href="#">Borrar</a>
92	10/05/2022 13:30h	PRESENCIAL	1	1	11111111A	<a href="#">Actualizar</a> <a href="#">Borrar</a>

Figuras XI prueba 7, 10 y 13 - ADMINISTRADOR o EMPLEADO quieren ver las citas, y mismo resultado si ADMINISTRADOR quiere ver los usuarios o empleados (READ) - Fuente: Elaboración propia

Pruebas 8, 11 y 14 ADMINISTRADOR o EMPLEADO quieren modificar una cita, y mismo resultado si ADMINISTRADOR quiere modificar un usuario o empleado (UPDATE)

ID CITA	FECHA	TIPO	ESTADO	ID SERVICIO	DNI USUARIO	Actualizar	Borrar
75	23/05/2022 11:00h	PRESENCIAL	1	5	11111111A	Actualizar	Borrar
77	24/05/2022 10:00h	TELEMÁTICA	1	1	11111111A	Actualizar	Borrar
79	24/05/2022 09:00h	PRESENCIAL	1	4	22222222B	Actualizar	Borrar
80	14/05/2022 16:15h	TELEMÁTICA	1	5	22222222B	Actualizar	Borrar
85	14/05/2022 16:15h	PRESENCIAL	1	2	22222222B	Actualizar	Borrar
87	25/05/2022 16:30h	TELEMÁTICA	1	4	11111111A	Actualizar	Borrar
88	06/07/2022 16:30h	TELEMÁTICA	1	1	11111111A	Actualizar	Borrar
89	16/05/2022 11:20h	PRESENCIAL	1	1	11111111A	Actualizar	Borrar
92	10/05/2022 13:30h	PRESENCIAL	1	1	11111111A	Actualizar	Borrar

Figuras XI pruebas 8, 11 y 14: ADMINISTRADOR o EMPLEADO quieren modificar una cita, y mismo resultado si ADMINISTRADOR quiere modificar un usuario o empleado (UPDATE) - Fuente: Elaboración propia

Pruebas 9, 12 y 15. ADMINISTRADOR o EMPLEADO quieren eliminar una cita, y mismo resultado si ADMINISTRADOR quiere eliminar un usuario o empleado (DELETE)

ID CITA	FECHA	TIPO	ESTADO	ID SERVICIO	DNI USUARIO	Actualizar	Borrar
77	24/05/2022 10:00h	TELEMÁTICA	1	1	11111111A	Actualizar	Borrar
79	24/05/2022 09:00h	PRESENCIAL	1	4	22222222B	Actualizar	Borrar
80	14/05/2022 16:15h	TELEMÁTICA	1	5	22222222B	Actualizar	Borrar
85	14/05/2022 16:15h	PRESENCIAL	1	2	22222222B	Actualizar	Borrar
87	25/05/2022 16:30h	TELEMÁTICA	1	4	11111111A	Actualizar	Borrar
88	06/07/2022 16:30h	TELEMÁTICA	1	1	11111111A	Actualizar	Borrar
89	16/05/2022 11:20h	PRESENCIAL	1	1	11111111A	Actualizar	Borrar

Figuras XI pruebas 9, 12 y 15. ADMINISTRADOR o EMPLEADO quieren eliminar una cita, y mismo resultado si ADMINISTRADOR quiere eliminar un usuario o empleado (DELETE) - Fuente: Elaboración propia

Muchas gracias.