

Trabajo Fin de Ciclo

# Raúl Salazar Roldán

Desarrollo de Aplicaciones Web



## UNIMED

# 2021



## **ÍNDICE**

1. Motivación.
2. Definición General, Introducción.
3. Especificaciones. Requisitos, alcance, y limitaciones.
4. Herramientas y Recursos.
5. Desarrollo, metodología y estilos.
6. Procedimientos de instalación y demo.
7. Modelo de datos.
8. Mejoras futuras.

# UNIMED

## Trabajo de Fin de Ciclo

### **1. Motivación.**

La motivación del proyecto Unimed es ampliar los conocimientos ya adquiridos con Angular y PHP, así como facilitar la gestión de un lugar de trabajo muy solicitado sobre todo durante tiempos de pandemia y permitir gestionar en un futuro más de un centro médico a la vez, usando una única página web como plataforma, y unificándolos..

### **2. Definición general o Introducción. Respondiendo a las preguntas ¿qué? ¿para qué? y ¿para quién?**

Unimed se trata de una plataforma orientada a facilitar la gestión de citas de uno, o varios centros médicos en el futuro situados en múltiples sitios del país. Así como ofrecer la oportunidad a sus usuarios de solicitar cita con sus médicos de confianza en fechas que les sean convenientes.

El objetivo del proyecto es ayudar con la organización en los centros médicos. Se orienta a todo tipo de público ya que es una plataforma de uso sencillo y generalizado, a la cual podrán acceder y utilizar cualquier tipo de personal. El usuario medio será un particular que desee acudir al médico de X especialidad cualquier día que le sea posible.

### **3. Especificaciones. Requisitos, alcance y limitaciones del proyecto.**

El proyecto dispone de una página principal en la cual se muestran las especialidades dadas en los centros médicos, y páginas de carácter secundario como una lista de los médicos pertenecientes a los centros así como su especialidad, y otra página con la ubicación del centro médico principal, además de varias localizaciones alternativas donde otros centros médicos se encuentran.

La plataforma posee un login y un registro para que las personas ingresen sus datos principales, así como su número de la Seguridad Social , teléfono... Para facilitar la consulta y el contacto.

También se permite modificar tu perfil una vez has entrado, así como pedir cita con X especialidad la cual pertenece a un médico concreto, así como borrar una cita ya existente en caso de querer cancelarla.

El alcance se limita a los centros médicos almacenados en la base de datos, ya que no se puede pedir cita fuera de estos.

Entre otras limitaciones no permite entrar con roles distintos, esta plataforma solo permite entrar como un paciente que busca encontrar una consulta, necesitaría de una plataforma secundaria orientada al personal sanitario si se desea aún más personalización.

#### 4. Herramientas y recursos.

Lenguaje	Framework	End
TypeScript	Angular	Front-End
PHP	Symfony	Back-End

Las herramientas utilizadas para este proyecto son:

- Angular como framework para el Front-End
- TypeScript como lenguaje usado en Angular para el Front-End
- Symfony como framework para el Back-End
- Php como lenguaje usado en Symfony para el Back-End
- PhpMyAdmin Para la base de datos, usando MySQL (MariaDB) a través de XAMPP (Local)

## 5. Desarrollo, metodología y estilo.

El desarrollo se ha llevado de la siguiente forma:

Primero se ha comenzado el desarrollo del modelo de datos necesario para la aplicación y ahorrarnos el mayor numero de cambios posibles.

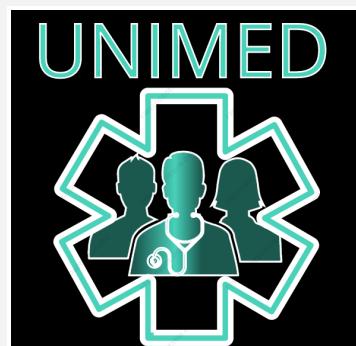
En segundo lugar se ha llevado a cabo del desarrollo del backend desde symfony junto con cada una de sus rutas.

En tercer lugar se han testeado (incluido el token cuando era necesario) cada una de las rutas de la aplicación de cara a empezar con el front-end.

En cuarto lugar se ha testeado el login desde el front-end y su comunicación con el back, y se ha procedido con el resto cuando esto ha funcionado.

En quinto lugar se han aplicado estilos, validaciones... y se ha comenzado la documentación de forma simultánea.

Para el logotipo se ha optado por uno sencillo y familiar:



El estilo se ha desarrollado a partir de Bootstrap 4, aplicando algunas reglas CSS en caso de necesitar pequeños ajustes a la plantilla, reglas de @Media para hacer el diseño responsivo, y otras técnicas de estilos. Todos los componentes heredan el body de un index.html principal.

A continuación explicamos los estilos más importantes:

Estilo base:

A partir de esta hoja de estilos heredan el resto de componentes, por lo que situamos una raíz y un body de un 100% para que luego cada contenedor utilice el espacio necesario dependiendo del componente.

```
/* You can add global styles to this file, and also import other style files */
body {
    background:
        url("assets/public/inicio.jpg");
}

html, body { height: 100%; }

app-root{
    width: 100%;
    height: 100%;
}
```

## Navegación:

En el estilo de este componente tenemos un nav sencillo con varios enlaces a las rutas de la página, y otro oculto en caso de no estar logueado (Citas). Además vemos bajo el nav un botón que nos permitirá hacer login a la plataforma, o editar nuestro perfil en caso de estar logueados. En caso de estar en un dispositivo móvil, bootstrap colapsará la navegación y obtendremos a través de un botón similar al del login las rutas de la página.

```
@media only screen and (max-width: 975px) {  
    .navbar-toggler{  
        margin-left: 5%;  
    }  
  
    .fa-sign-in{  
        font-size: 2em;  
    }  
    .navbar-toggler{  
        color: white;  
        border-radius: 5%;  
        border-style: double;  
        border-color: #rgb(1, 107, 110);  
    }  
  
    .navbar[]  
  
    font-family: 'Cormorant Unicase', serif;  
    font-size: 1.3em;  
  
}  
  
a{  
    font-family: 'Cormorant Unicase', serif;  
    font-size: 1.3em;  
}  
  
.centro{  
    font-size: 1.7em;  
}  
  
.btn{  
    font-family: 'Acme', sans-serif;  
    font-size: 1.7em;  
}
```

## Home:

Es el componente de inicio, en el cual aparecen las especialidades del centro, y se forma con un card creado con bootstrap en columnas, de ese modo situamos 3 columnas de cards para aprovechar espacio, en dispositivos más pequeños se reduce la fuente y los márgenes hasta que para móvil se opta por un estilo en lista.

```
1  p{
2      font-family: 'Oswald', sans-serif;
3  }
4
5  .card{
6      font-size: 22px;
7  }
8
9  h1{
10     font-family: 'Acme', sans-serif;
11 }
12
13 .jumbotron{
14     margin-right: 20%;
15     margin-left: 20%;
16 }
17
18 @media only screen and (max-width: 800px) {
19     .jumbotron{
20         width: 100%;
21         margin: 0;
22     }
23 }
24
25 @media only screen and (max-width: 1603px) {
26     p{
27         font-size: 80%;
28     }
29 }
```

## Médicos:

Es el componente encargado de mostrar los médicos actuales del centro, se muestra como un card, organizando cada uno en dos columnas, una para el nombre y otro para el apellido. Los dispositivos móviles ven esto de forma muy reducida pero obtienen la ventaja de que la lista se transforma en un acordeón en el cual pueden ver la especialidad de cada médico.

## Cuando estamos en modo móvil:

```
@media only screen and (max-width: 1050px) {
  .estiloDefecto {
    display: none;
    visibility: hidden;
  }
  .estiloMovil {
    display: inline-block;
    visibility: visible;
    width: 100%;
    padding: 0;
  }
  .a{
    width: 60%;
    margin: auto;
  }
  .btn{
    width: 100%;
    margin: 0.2vh;
  }
  h1{
    font-size: 2.2vh;
  }
  .card{
    border-style: double;
    border-color: #rgba(0, 132, 172, 0.459);
    margin-top: 2vh;
  }
}
```

```
op > componentes > medicos > medicos.component.html > div.container.estiloMovil
<div class="container estiloMovil">
  <div class="acordeon" *ngFor='let medico of medicos'>
    <a class="btn btn-primary text-primary bg-white" data-toggle="collapse" href="#{{medico.id}}" role="button" aria-expanded="false" aria-controls="collapseExample">
      {{medico.nombre}} {{medico.apellidos}}
    </a>
    <div class="collapse" id="{{medico.id}}">
      <div class="card card-body card-accordion mb-2">
        <h1>{{medico.especialidad}}</h1>
      </div>
    </div>
  </div>
</div>
```

Cuando estamos en modo escritorio:

```
@media only screen and (min-width: 1051px) {

    .estiloMovil {
        display:none;
        visibility: hidden;
    }

    .contenedor {
        margin: 15%;
        margin-top: 2%;
        margin-bottom: 5%;
        width: 65%;
        padding-bottom: 10%;
    }

    .card {
        font-size: 2rem;
        font-family: "JetBrains Mono", monospace;
        width: 100%;
    }

    .header-title {
        font-size: 2rem;
        font-family: "JetBrains Mono", monospace;
        width: 100%;
    }

    h2 {
        font-family: "Acme", sans-serif;
        width: 100%;
    }
}
```

```
<div class="contenedor estiloDefecto">
    <h2 class="mt-5">Médicos</h2>
    <hr>
    <div class="row">
        <div id="nombreLabel" class="col-sm header-title">
            Nombre
        </div>
        <div id="apeLabel" class=" col-sm header-title">
            Apellidos
        </div>
    </div>

    <hr>
    <div class="card py-4 bg-white border-primary text" *ngFor='let medico of medicos' >
        <div class="row">
            <div class="ml-3 col-sm text-primary" id="nombre">
                {{ medico.nombre }}
            </div>
            <div class="ml-3 col-sm text-primary" id="apellidos">
                {{ medico.apellidos }}
            </div>
        </div>
    </div>
</div>
```

## Mapa:

En esta página se organizan dos columnas, una que posee el mapa donde se encuentra el centro, y otra mediante la cual podemos navegar a otros centros si pinchamos en ellos.

```
<div class="row">
  <div class="col" id="col1">
    <h1>CÓMO ENCONTRARNOS</h1>
    <div id='posicionMapa' style='height: 70vh'></div>
  </div>
  <div class="col" id="col2">

    <h1>OTROS CENTROS (Click para explorar)</h1>
    <div class="card" >

      <ul *ngFor='let otro of otros' class="list-group list-group-flush" >
        <button (click)="cambiar(otro)" class="list-group-item border-secondary" id="zona">
          Nombre: {{otro.nombre}} <br>
          Localidad: {{otro.localidad}} <br>
          Direccion: {{otro.direccion}} <br>
          Telefono: {{otro.telefono}}
        </button>
      </ul>
    </div>
  </div>
</div>
```

```
@import "https://unpkg.com/leaflet@1.7.1/dist/leaflet.css";
```

```
@media only screen and (max-width: 1000px) {
```

```
}
```

```
#zona:hover {
```

```
  cursor: pointer;
```

```
}
```

```
ul{
```

```
  border: 0.3vh;
  border-style: double;
  border-color: #rgb(8, 111, 228);
```

```
}
```

```
.row{
```

```
  margin-top: auto;
  margin: 3%;
```

```
}
```

```
h1{
```

```
  font-family: 'Acme', sans-serif;
```

```
}
```

```
/* Clear floats after the columns */
```

```
.row:after {
  content: "";
  display: table;
  clear: both;
}
```

## Perfil/Registro/Login:

Esta parte contiene un solo formulario que nos servirá para editar nuestro perfil, el registro y login tienen prácticamente de la misma estructura. de igual modo tenemos un botón borrar que nos permitirá borrar el mismo. Ya que la mitad del trabajo con formularios la hace bootstrap, no es necesario demasiado CSS.

```
<div class="cont">
<form [formGroup]="formEditar" (submit)="edit()">

  <div class="form-group">
    <label for="exampleInputPassword1">Email</label>
    <input type="email" class="form-control" value="{{currentUser.email}}" formControlName="email">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Contraseña</label>
    <input type="password" class="form-control" value="{{currentUser.password}}" formControlName="password">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Nombre</label>
    <input type="text" class="form-control" value="{{currentUser.nombre}}" formControlName="nombre">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Apellidos</label>
    <input type="text" class="form-control" value="{{currentUser.apellidos}}" formControlName="apellidos">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Telefono</label>
    <input type="text" class="form-control" value="{{currentUser.telefono}}" formControlName="telefono">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Seguridad Social</label>
    <input type="text" class="form-control" value="{{currentUser.s_social}}" formControlName="social">
  </div>
  <input type="submit" class="btn btn-primary" value="Actualizar" [disabled]="#formEditar.invalid">

</form>
<button class="btn btn-primary" (click)="borrar()">Borrar</button>
</div>
```

```
.btn{
  width: 100%;
  font-size: 1.8vh;
  padding: 2px;
  margin: 0.3em;
  text-align: center;
}

form{
  width: 100%;
  margin: auto;
}

.alert{
  width: 60%;
  margin: auto;
  font-size: 2em;
}

.cont{
  width: 70%;
  margin: auto;
}
```

## Citas:

La parte de citas se organiza en una tabla en la cual ponemos las citas que el usuario tiene pendientes, y en caso de tener la llenamos con cards que tienen su propio botón para cancelarlas.

Justo debajo tenemos un formulario que nos ayudará a concertar una cita nueva. En caso de dispositivos móviles, las citas tomarán forma de acordeón permitiéndonos mostrar más datos solo si pinchamos sobre ellas.

```
#estiloMovil {  
    display: none;  
    visibility: hidden;  
}  
.alert{  
    width: 60%;  
    margin: auto;  
    font-size: 2em;  
}  
@media only screen and (max-width: 1200px) {  
    #estiloDefecto {  
        display: none;  
        visibility: hidden;  
    }  
}  
@media only screen and (max-width: 1200px) {  
    #estiloMovil {  
        display: contents;  
        visibility: visible;  
  
        text-align: center;  
        padding: 2px;  
  
    }  
    .acordeon{  
        width: 100%;  
        padding: 1%;  
  
    }  
    .card-accordeon{  
        border-style: dashed;  
        border-color: #rgb(0, 106, 177);  
    }  
}
```

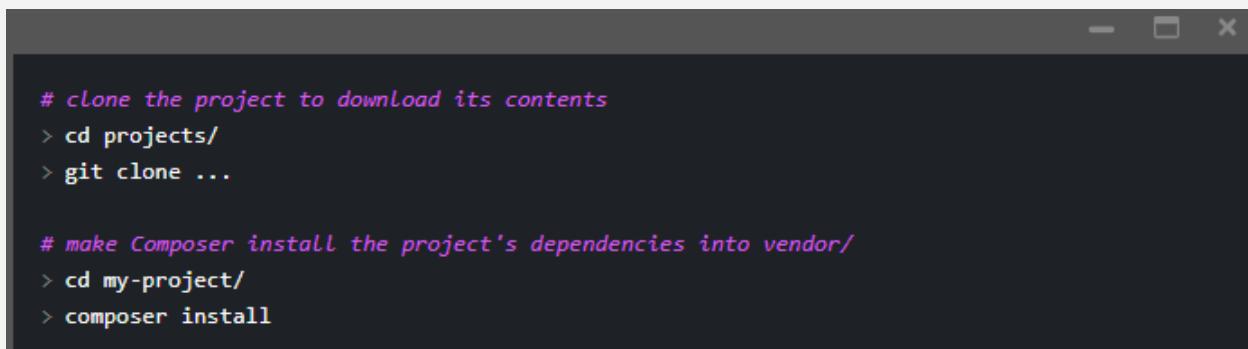
## 6. Procedimientos de instalación y demo.

### Base de datos

Tras instalar XAMPP ejecutaremos phpmyadmin e importaremos el archivo .sql que incluimos en el proyecto. Cuando el script finalice usted podrá continuar configurando el back y el front.

### Back

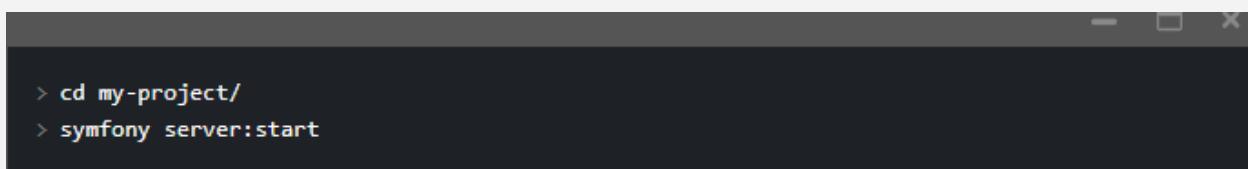
En primer lugar clonaremos el back e instalaremos el framework



```
# clone the project to download its contents
> cd projects/
> git clone .....

# make Composer install the project's dependencies into vendor/
> cd my-project/
> composer install
```

En segundo lugar iremos a la carpeta contenedora y ejecutaremos el comando mostrado para poner el back en funcionamiento



```
> cd my-project/
> symfony server:start
```

## Front

En el caso de front clonaremos de la misma forma el proyecto en la carpeta que deseemos, lo abriremos con un IDE (Preferiblemente Visual Code) y con Ctrl + ñ abriremos un terminal con el cual debemos navegar a la carpeta si aun no estamos en ella (Usando el comando CD) y ejecutaremos npm install para que nos haga la instalacion de todos los paquetes que el proyecto utiliza. Tras ello solo necesitaremos ejecutar ‘ng serve -o’

## Demo

Inicio (Escritorio):

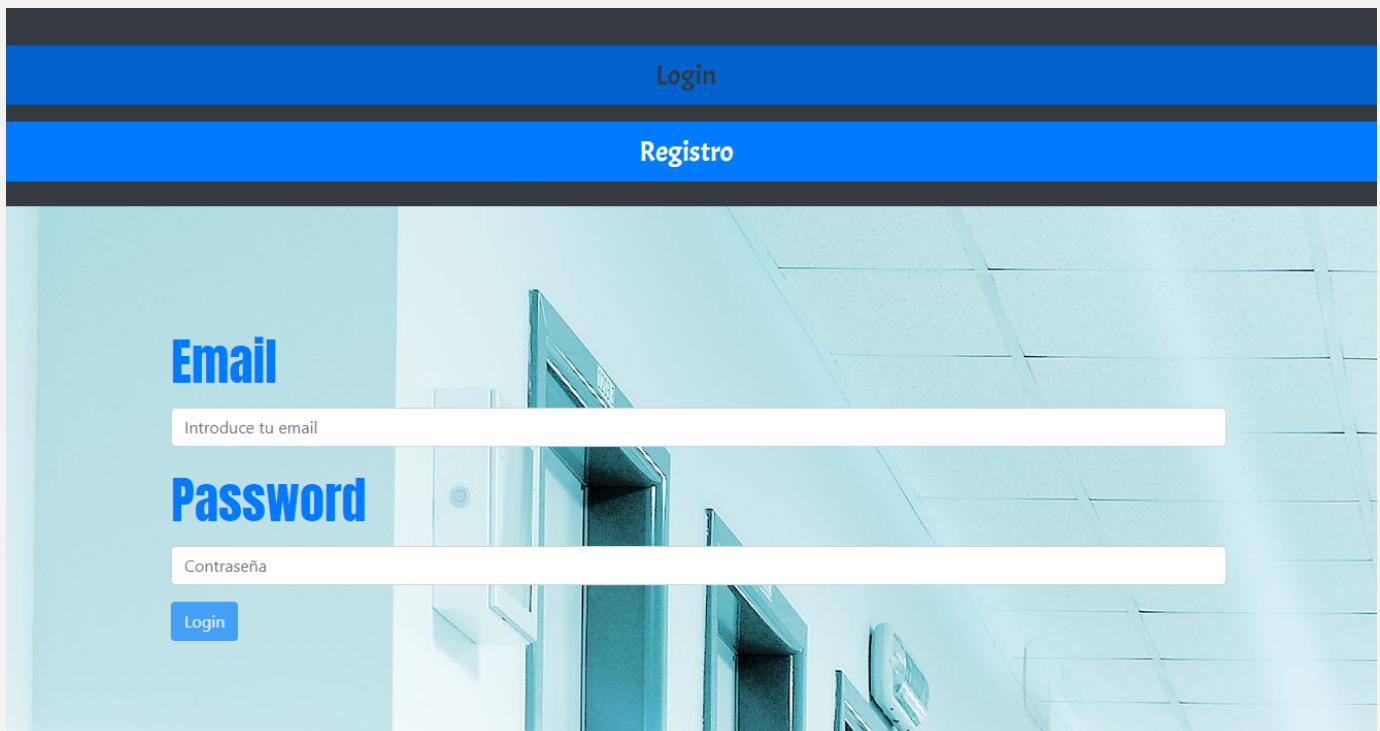


Inicio (Móviles):

Si hacemos click en el icono superior derecho desplegamos el menú.



## Login y registro:



### NUEVO USUARIO

**Email**  
Introduce tu email

**Password (Longitud Mínima: 4)**  
Password

**Nombre**  
Nombre

**Apellidos**  
Apellidos

**Telefono**  
Telefono

**Seguridad Social (12 Dígitos)**  
Seguridad Social

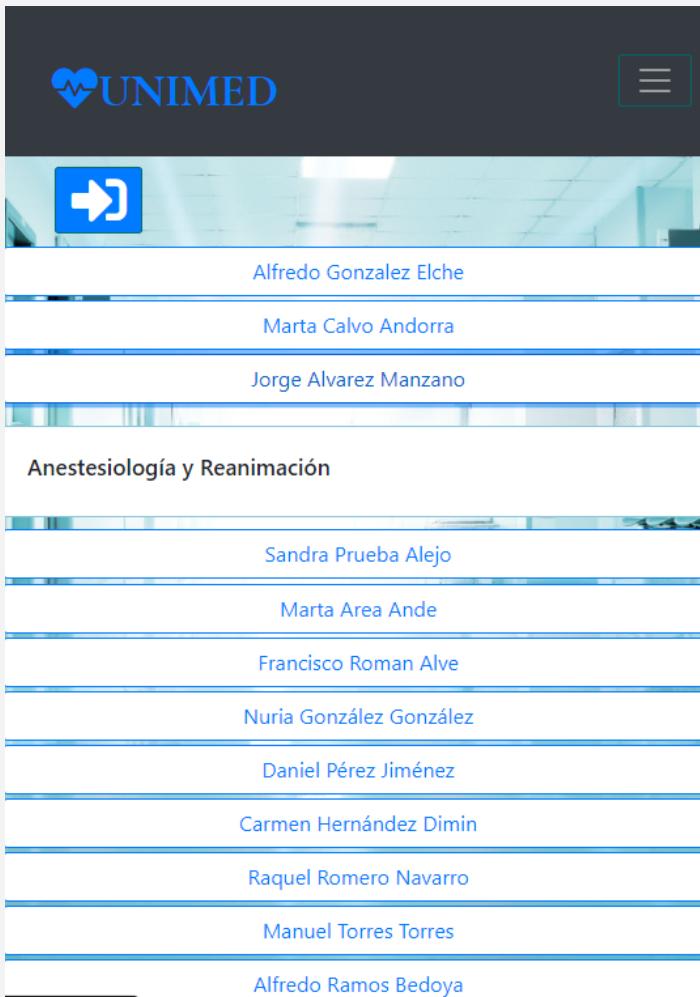
**Registrarse**

Raúl Salazar Roldán

## Vista Médicos (Escritorio):

Médicos	
Nombre	Apellidos
Alfredo	Gonzalez Elche
Marta	Calvo Andorra
Jorge	Alvarez Manzano
Sandra	Prueba Alejo
Marta	Area Ande
Francisco	Roman Alve
Nuria	González González
Daniel	Pérez Jiménez

## Vista Médicos (Movil):



	
	
Alfredo Gonzalez Elche	
Marta Calvo Andorra	
Jorge Alvarez Manzano	
<b>Anestesiología y Reanimación</b>	
Sandra Prueba Alejo	
Marta Area Ande	
Francisco Roman Alve	
Nuria González González	
Daniel Pérez Jiménez	
Carmen Hernández Dímin	
Raquel Romero Navarro	
Manuel Torres Torres	
Alfredo Ramos Bedoya	

## Vista Ubicación:

Aquí el usuario podrá consultar el teléfono y la localización de los centros médicos, además clicando en cada uno de ellos el mapa les dará una ayuda sobre donde se encuentran dichos centros.



## Sobre Mí:

En esta vista, el usuario podrá visualizar tanto el CV como la Carta de presentación del desarrollador al pinchar en 'ver'.

**VER CARTA**

Carta de Presentación [Alcázar De San Juan 24/05/2021]

[Nombre de la empresa]  
[Nombre y apellido de la persona a quien va dirigida la carta]  
[Dirección de la empresa]

Asunto: **Candidatura al puesto de Programador/Desarrollador**  
Estimado(a) [Nombre y apellido de la persona de RRHH].

Al departamento de RRHH de la empresa (nombre):

Se adjuntan mi Cv y Carta de Presentación de Grado Superior de Desarrollo de Aplicaciones Web que terminaré en Junio de este año y además, soy titulado en Técnico Superior en Desarrollo de Aplicaciones Multiplatformas. He decidido enviarles mi CV porque estoy interesado en hacer prácticas profesionales en su empresa. Considero que trabajar en su empresa será una gran oportunidad para el desarrollo de mi carrera profesional.

Durante mi etapa como estudiante, he tenido ocasión de ampliar mis conocimientos y el grado me ha ofrecido la oportunidad de profundizar en diferentes áreas, que considero de especial valor para poder trabajar en su empresa. Además, hablo inglés fluidamente y poseo el nivel C1 (Cambridge).

Considero que las prácticas en su empresa podrían resultar muy provechosas para mi desarrollo y especialización tanto personal como profesional. Además, creo que yo podría aportarles la pasión y el entusiasmo por el trabajo bien hecho, responsabilidad, habilidades y conocimientos.

Me gustaría poder ampliar mi perfil como candidato en una entrevista personal. Por este motivo, estaré disponible al teléfono 648229147 o por correo electrónico a salazar.devel@outlook.com.

Espero que mi perfil pueda ser útil para poder desempeñar el cargo propuesto y tener la oportunidad de discutir más acerca de la empresa y sus objetivos.

Un cordial saludo,

Raúl Salazar Roldán

**VER CV**

CURRICULUM VITAE

**1. DATOS PERSONALES**

Apellido: Salazar Roldán  
Nombre: Raúl  
Teléfono: 648229147  
Fecha de nacimiento: 01-07-1997  
Domicilio: Avenida Juan Carlos I nº99  
Localidad: La Puebla de Almoradiel (Toledo)  
Email: salazar.devel@outlook.com

**2. FORMACIÓN**

**Académica**

- 2010-2011 Titulación Superior en Desarrollo de Aplicaciones Web en IES Juan Bocan de Almoradiel de San Juan (Toledo)
- 2018-2020 Titulación Superior en Desarrollo de Aplicaciones Web en IES Juan Bocan de Almoradiel de San Juan (Toledo)
- 2018-2018 Doble Bachillerato Técnico en ETS Alfonso X el Sabio de Almoradiel (Toledo)

**Complementaria**

Conocimientos informáticos 2 o más medios/teléfonos móviles como Office y Open Office, sistema de navegación de documentos como Microsoft Word, Excel e Internet Explorer.

Nivel de inglés: Alto - Certificado en Cambridge C1

**3. EXPERIENCIA PROFESIONAL**

2020 Asistente de Oficina en Caja de Pensiones, Ciudad Real con una duración de 300 horas

**4. OTROS DATOS DE INTERÉS**

Carret de conducir  
Verbaje profundo y流uididad en inglés.  
Ingeniería mecánica.

EN PUERTA DE ALMORADIEL A 23-02-2021

**Carta de Presentación**

[Alcázar De San Juan 24/05/2021]

[Nombre de la empresa]  
[Nombre y apellido de la persona a quien va dirigida la carta]  
[Dirección de la empresa]

Asunto: **Candidatura al puesto de Programador/Desarrollador**  
Estimada(o) [Nombre y apellido de la persona de RRHH].

Al departamento de RRHH de la empresa (nombre):

Soy estudiante del Ciclo Formativo de Grado Superior de Desarrollo de Aplicaciones Web que terminaré en Junio de este año y además, soy titulado en Técnico Superior en Desarrollo de Aplicaciones Multiplatformas. He decidido enviarles mi CV porque estoy interesado en hacer prácticas profesionales en su empresa. Considero que trabajar en su empresa sería una gran oportunidad para el desarrollo de mi carrera profesional.

Durante mi etapa como estudiante, he tenido ocasión de ampliar mis conocimientos y el grado me ha ofrecido la oportunidad de profundizar en diferentes áreas, que considero de especial valor para poder trabajar en su empresa. Además, hablo inglés fluidamente y poseo el nivel C1 (Cambridge).

Considero que las prácticas en su empresa podrían resultar muy provechosas para mi desarrollo y especialización tanto personal como profesional. Además, creo que yo podría aportarles la pasión y el entusiasmo por el trabajo bien hecho, responsabilidad, habilidades y conocimientos.

Me gustaría poder ampliar mi perfil como candidato en una entrevista personal. Por este motivo, estaré disponible al teléfono 648229147 o por correo electrónico a salazar.devel@outlook.com.

Espero que mi perfil pueda ser útil para poder desempeñar el cargo propuesto y tener la oportunidad de discutir más acerca de la empresa y sus objetivos.

## Otros aspectos:

No es necesario mencionar que la aplicación posee varias alertas que permitirán al usuario saber qué error ha habido a la hora de llenar formularios, además todos los formularios son reactivos por lo que no será posible introducir información incorrecta en la mayoría de los casos.

Por ejemplo, si el email de registro ya existe:

The screenshot shows a mobile application interface for 'Nuevo Usuario'. The form fields are as follows:

- Email: testing@gmail.com
- Password (Longitud Mínima: 4): ..... (redacted)
- Nombre: Raul
- Apellidos: Salazar
- Teléfono: 655555444
- Seguridad Social (12 Dígitos): 545454545455

An error message 'El email ya existe' (The email already exists) is displayed in a red box above the 'Registrarse' button. The 'Registrarse' button is blue.

## Citas:

En el caso de las citas, esta es la vista:

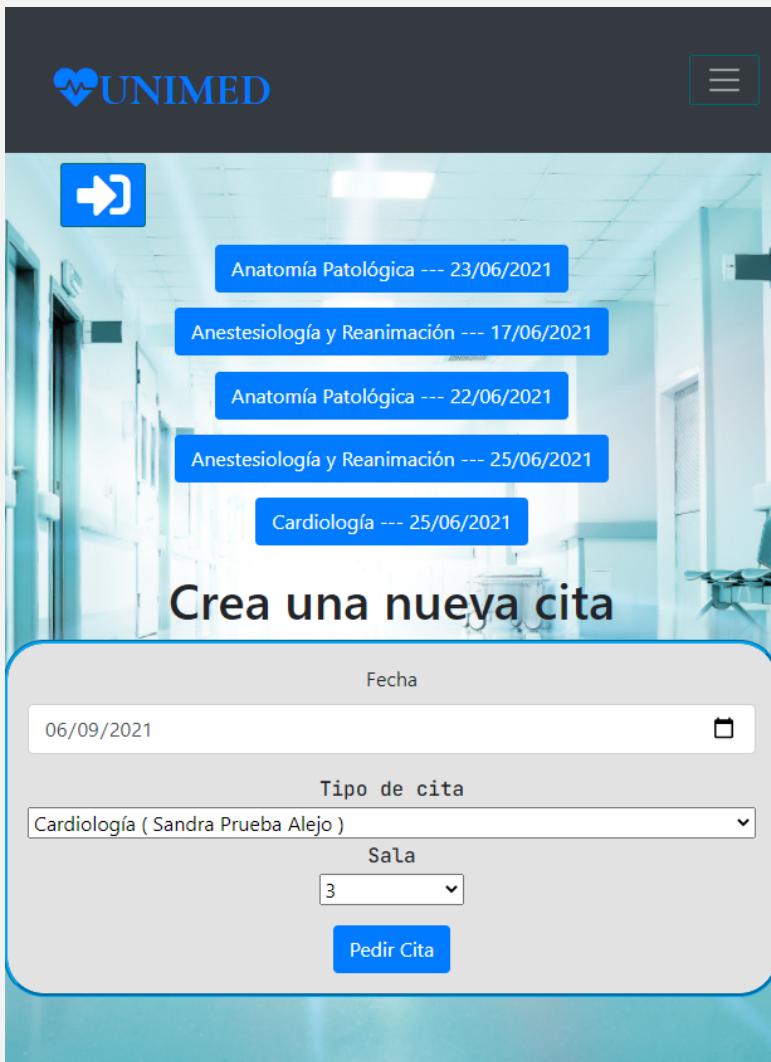
The screenshot shows a user interface for managing appointments (Citas). At the top, there are five columns: Médico (Doctor), Fecha (Date), Estado (Status), Sala (Room), and Operación (Operation). Below this, a table lists three existing appointments:

Médico	Fecha	Estado	Sala	
Marta Calvo Andorra (Anatomía Patológica)	23/06/2021	Activa	4	<a href="#">Pincha para borrar la cita</a>
Jorge Alvarez Manzano (Anestesiología y Reanimación )	17/06/2021	Activa	3	<a href="#">Pincha para borrar la cita</a>
Marta Calvo Andorra (Anatomía Patológica)	22/06/2021	Activa	4	<a href="#">Pincha para borrar la cita</a>

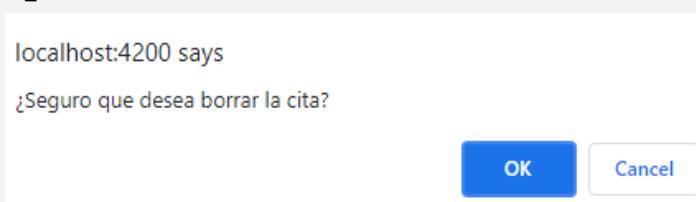
Below the table, there is a date picker for June 2021, showing the 10th selected. A message 'Puedes elegir una cita' (You can choose an appointment) is displayed above a text input field for 'mm/dd/yyyy'. At the bottom, there are dropdown menus for 'Tipo de cita:' and 'Sala:', and a blue button labeled 'Pedir Cita'.

El sistema solo te permitirá pedir cita sin urgencia (es decir no se permite pedir una cita en el día actual), ya que al no exigir una hora concreta es necesario un tiempo mínimo de preparación para el equipo médico por lo que mínimo deberá ser para el día siguiente. Se especificará qué sala se desea (de las que se encuentran activas en la base de datos) y que especialidad se desea. El hecho de especificar la sala viene debido a que se quiere ayudar a las personas con algún tipo de discapacidad o problemas de movimiento, por lo que se podrá elegir una sala más cerca de la entrada, en la primera planta, etc...

Esta es la vista para dispositivos móviles:



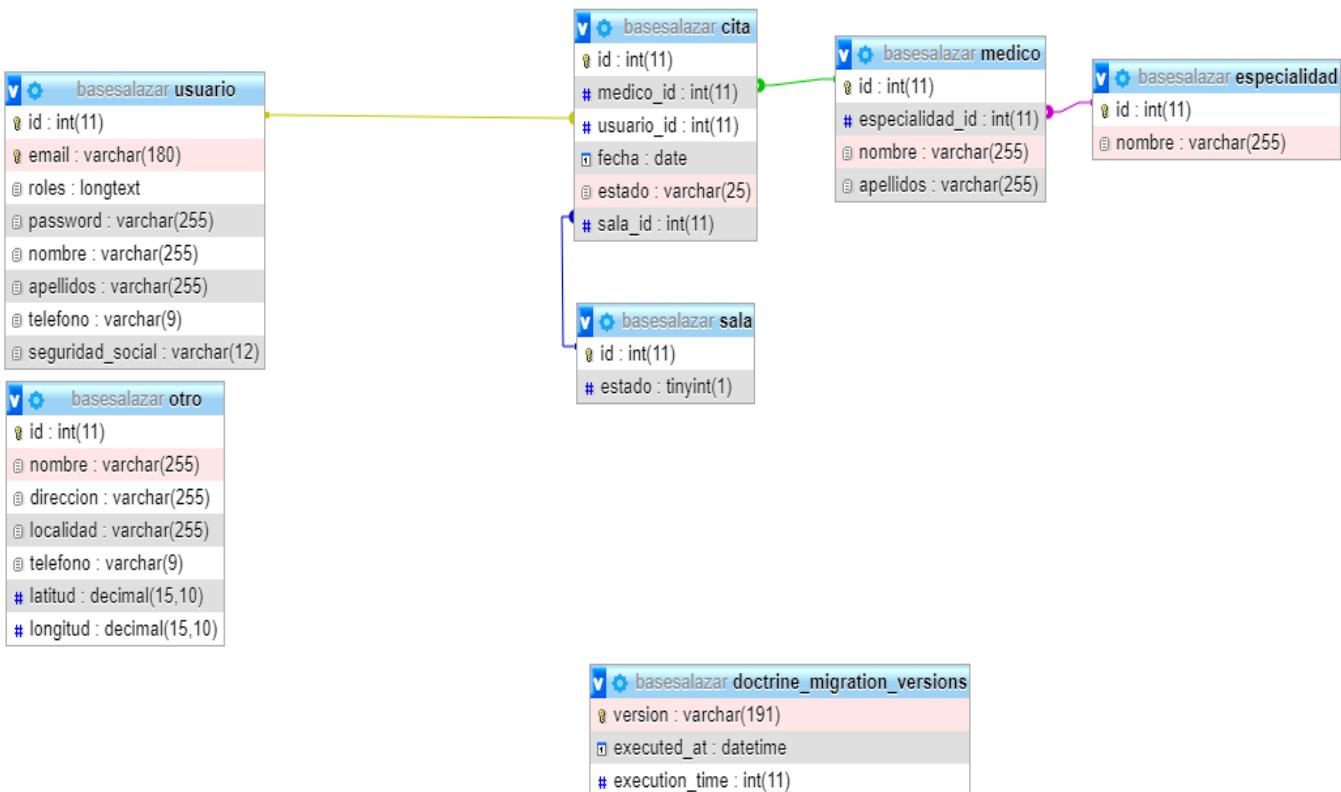
Si deseamos borrar la cita solo tenemos que pinchar en el botón azul y nos saldrá un propio dialog desde el navegador que usemos para confirmar la operación.



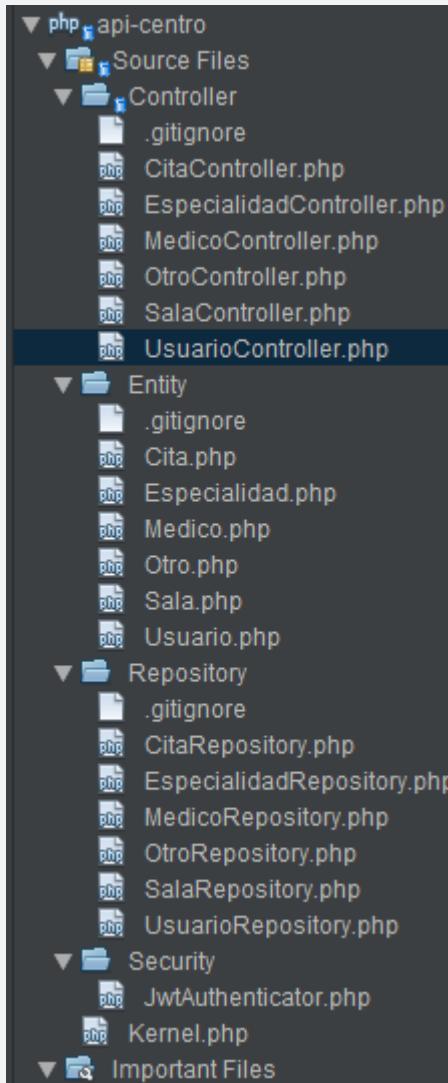
## 7. Modelo de datos y arquitectura del sistema.

**La arquitectura del sistema es la siguiente:**

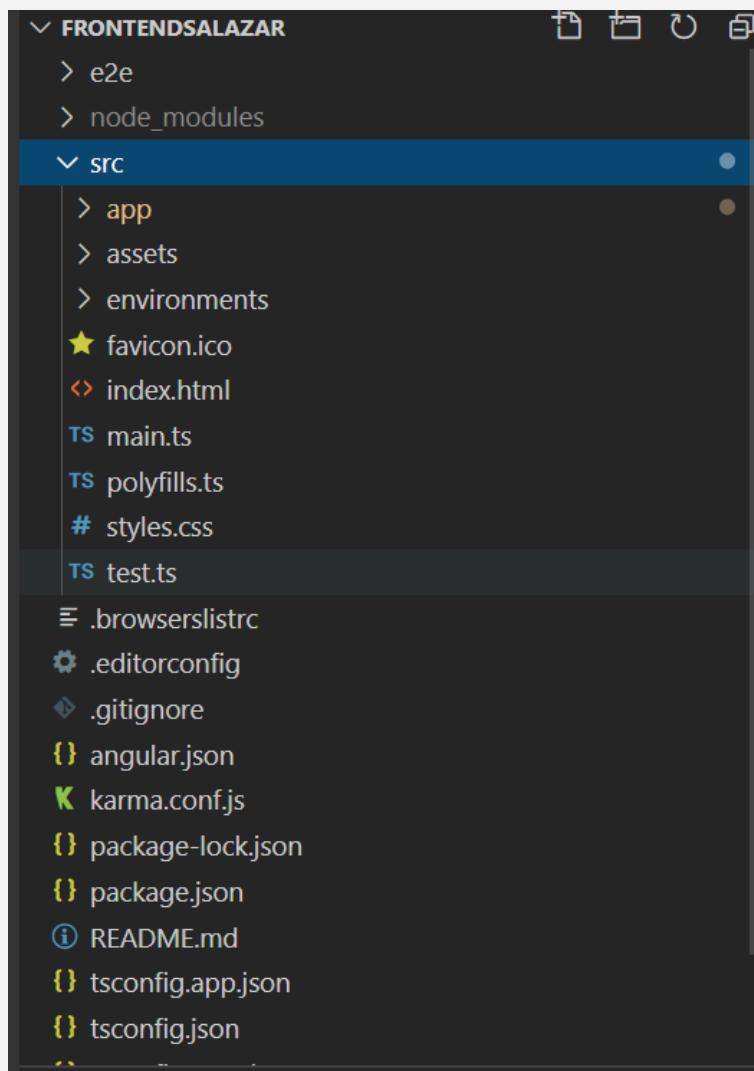
**Modelo de datos:**



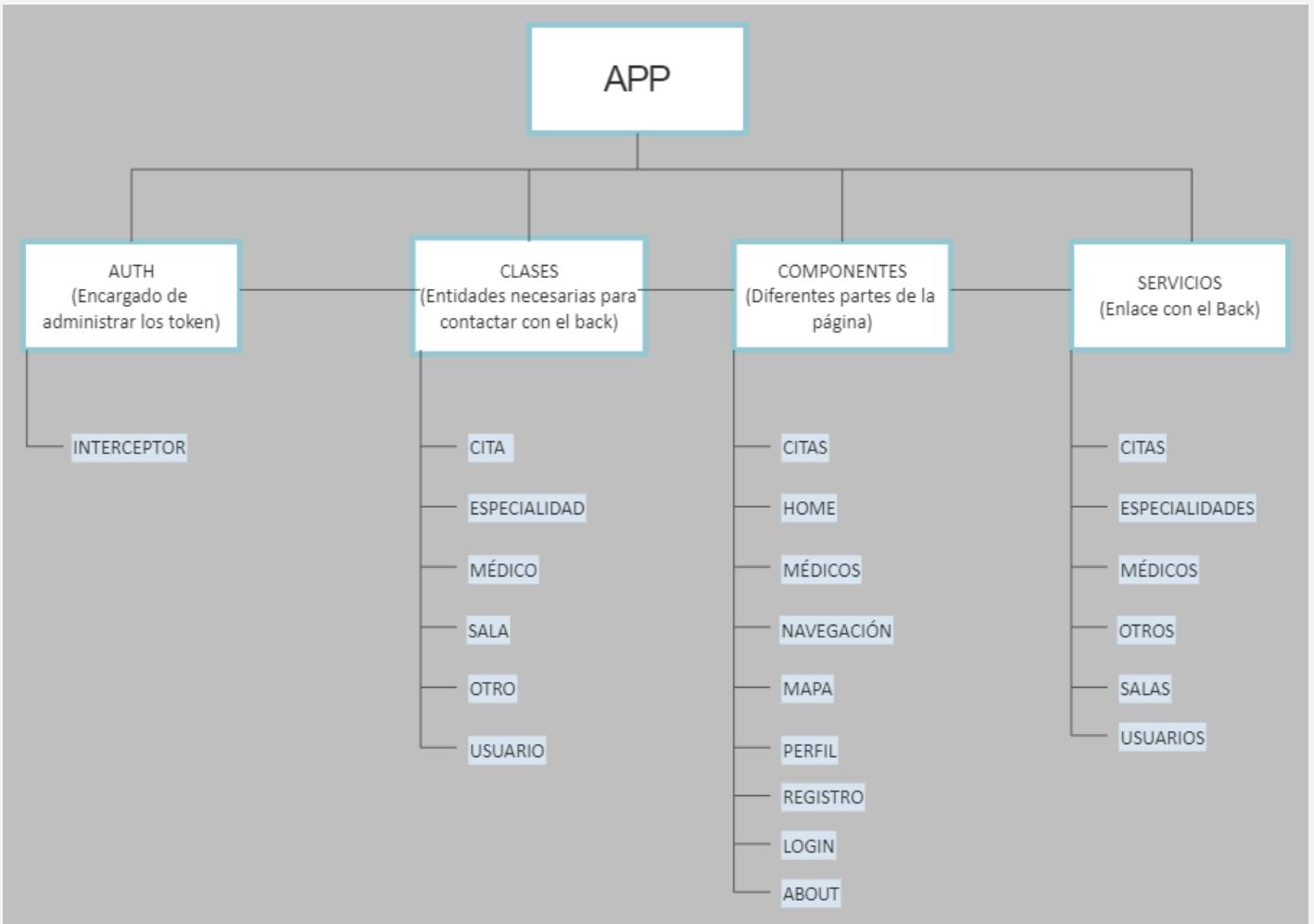
La estructura de la aplicación es muy similar tanto en front como en back, por parte del back tenemos entidades, controladores y repositorios, los cuales son necesarios para que symfony manipule los datos, así como un archivo para manejar el Json Web Token y los temas de autenticación.



Por parte del front, tenemos una estructura similar, con la diferencia de que tenemos antes de entrar a la carpeta de la aplicación el index.html así como los estilos base del proyecto y otros archivos json que configuran nuestro proyecto de react, así como la carpeta node\_modules que gestiona todos los paquetes npm que utilizamos, y assets que contiene las imágenes locales que se usarán para el fondo etc...



Bajando de capa nos encontramos con la siguiente estructura en la carpeta App



**Auth:** Contiene el token-interceptor que se ocupará de comprobar si el usuario se encuentra logueado a la hora de acceder a ciertas rutas, y todo lo relacionado con el JWT.

**Clases:** Son las entidades utilizadas por la aplicación, comunes a las del back y la base de datos, necesarias para que se entiendan correctamente.

**Componentes:** Son cada una de las partes que contiene la página, y que juntos conforman la página en su totalidad.

**Servicios:** Son archivos de typescript que se comunican con la api (back) mediante peticiones. Estos servicios son usados por los componentes para formarse con los datos necesarios y mostrarlos en sus archivos html correspondientes.

## 8. Mejoras futuras.

Como futuras mejoras se podría añadir un rol de administrador/médico para gestionar las salas disponibles, citas , así como una pestaña para generar informes sobre los usuarios existentes que hayan tenido citas. Además se podría implementar elegir el centro en el que pedir cita, cambiar el estado de las citas a completadas para tener un historial online y limitar el número de citas por sala durante el mismo día para evitar aglomeraciones.

Entre otras mejoras se podría pedir cita filtrando por especialidad en lugar de por un médico concreto en caso de tener varios con la misma especialidad, así como un chat directo con el médico o alguien encargado de solucionar dudas sobre el centro y el sistema de citas.

