



9-6-2023

Documentación Proyecto Integrado

2ºDAW



Manuel Ruiz Jiménez

Lista De Cambios

-- Version 0.1 --

- Desarrollamos los archivos sql para crear una base de datos MySql
- Creamos la documentacion del Anteproyecto.
- Empezamos Api en SpringBoot.

-- version 0.5 --

- Documentacion hasta 3.6
- Debido a errores en la Api cambiamos de SpringBoot a Laravel
- Realizamos las migraciones desde Laravel

-- version 0.9 --

- Api completa con seguridad
- Comienzo de Front en Angular

-- Version 1.0 --

- Desarrollo Front simple para probar funcionamiento

-- Version 1.1 --

- Incorporación de elementos de mayor complejidad puede aumentar la calidad del proyecto.

-- version 1.2

- Terminamos de Documentar
- modificacion del home de Angular para separar alumnos de docentes.

Índice

DOCUMENTACIÓN DEL PROYECTO	1
1.- INTRODUCCIÓN	1
2.- ESTUDIO DE VIABILIDAD	1
2.1.- DESCRIPCIÓN DEL SISTEMA ACTUAL	1
2.2.- DESCRIPCIÓN DEL SISTEMA NUEVO	1
2.3.- IDENTIFICACIÓN DE REQUISITOS DEL SISTEMA	1
2.4.- DESCRIPCIÓN DE LA SOLUCIÓN	3
2.5.- PLANIFICACIÓN DEL PROYECTO	4
2.6.- ESTUDIO DEL COSTE DEL PROYECTO	5
3.- ANÁLISIS DEL SISTEMA DE INFORMACIÓN	5
3.1.- IDENTIFICACIÓN DEL ENTORNO TECNOLÓGICO	5
3.2.- MODELADO DE DATOS	6
3.3.- IDENTIFICACIÓN DE LOS USUARIOS PARTICIPANTES Y FINALES	7
3.4.- IDENTIFICACIÓN DE SUBSISTEMAS DE ANÁLISIS	8
3.5.- ESTABLECIMIENTO DE REQUISITOS	9
3.6.- DIAGRAMAS DE ANÁLISIS	11
3.7.- DEFINICIÓN DE INTERFACES DE USUARIO	12
4.- CONSTRUCCIÓN DEL SISTEMA	16
5.- CONCLUSIONES (OTROS USOS, TAREAS SIN FINALIZAR, PROBLEMAS ENCONTRADOS Y SOLVENTADOS, QUE HE APRENDIDO...)	17
6.- GLOSARIO DE TÉRMINOS	17
7.- BIBLIOGRAFÍA	18

DOCUMENTACIÓN DEL PROYECTO

1.- Introducción

Mi proyecto tratara sobre una web que gestionará la FCT de alumnos, en la que los usuarios que sean profesores gestionan tanto a los alumnos como a las distintas empresas en la que se podrán realizar la FCT, y los usuarios alumnos solo podrán gestionar sus propios datos y subir sus CV, además de comprobar el estado de sus candidaturas, para el back-end lo gestionaremos con una API REST realizada en Laravel, que gestionará una base de datos MySQL en phpmyadmin, y que será consumida en el front-end con Angular

2.- Estudio de Viabilidad

2.1.- Descripción del Sistema Actual

El sistema actual de gestión del alumnado para FCT, se realiza a través hojas de cálculo para registrar y organizar los datos de los estudiantes, así como asignarles empresas para las prácticas. Sin embargo, este método manual puede ser algo limitado tanto en eficiencia como seguimiento.

2.2.- Descripción del Sistema Nuevo

Pensamos en gestionar todo esto a través de una web que simplificará la forma de hacerlo, en la cual, los docentes gestionan a los alumnos y a las distintas empresas en las que se pueden realizar la fct, teniendo un registro de las candidaturas de los alumnos, y los alumnos podrán gestionar sus datos y ver sus candidaturas a tiempo real.

2.3.- Identificación de Requisitos del Sistema

2.3.1.- *Requisitos de información*

Guardaremos los datos de los usuarios que serán o Docentes o Alumnos, además guardaremos los datos de las distintas empresas con sus sedes y llevaremos un registro de las candidaturas que tendrá cada usuario.

Se guardará en una base de datos que consta de 5 tablas:

Tabla Users cuenta con 10 campos:

- id → Identificador numérico (hasta 4 cifras) de la tabla, obligatorio
- password → contraseña para inicio de sesión (cadena de hasta 100 caracteres siendo mínimo 6), obligatorio
- nombre → Nombre del usuario (cadena de hasta 15 caracteres), obligatorio
- apellidos → Apellidos del usuario (cadena de hasta 30 caracteres), obligatorio
- edad → Edad del usuario (hasta 3 cifras), obligatorio
- email → Email del usuario (cadena de hasta 50 caracteres), obligatorio y único

- dni → DNI del usuario (cadena de hasta 9 caracteres), obligatorio y único
- telefono → Teléfono del usuario (cadena de hasta 9 caracteres), único
- rol → Tipo de usuario (cadena de hasta 13 caracteres tomando solo los valores “docente” o “alumno”), obligatorio

Tabla Empresas cuenta con 4 campos:

- id → Identificador numérico (hasta 4 cifras) de la tabla, obligatorio
- nombre → nombre de la empresa (cadena de hasta 15 caracteres), obligatorio
- cif → Identificador único de cada empresa (cadena de hasta 9 caracteres), obligatorio y único
- num_empleado → número de empleados de la empresa (hasta 6 cifras), obligatorio

Tabla Alumnos cuenta con 3 campos:

- id → Identificador numérico (hasta 4 cifras) de la tabla referente al id del usuario al que se refiere, obligatorio
- CV → nombre de archivo pdf (cadena de hasta 25 caracteres), obligatorio
- PROFESOR_SEGUIMIENTO_ID → Identificador de usuario con rol docente (hasta 4 cifras), obligatorio

Tabla Sedes cuenta con 5 campos:

- id → Identificador numérico (hasta 4 cifras) de la tabla, obligatorio
- nombre → Nombre de la sede (cadena de hasta 15 caracteres), obligatorio
- direccion → dirección de la sede (cadena de hasta 150 caracteres), obligatorio
- telefono → telefono de la sede (cadena de hasta 9 caracteres), obligatorio
- empresa_id → Identificador de la empresa a la que pertenece (hasta 4 cifras), obligatorio

Tabla Candidatura cuenta con 4 campos:

- id → Identificador numérico (hasta 4 cifras) de la tabla, obligatorio
- estado → Estado de la candidatura que podrá ser “Aceptada”, “Denegada” o “Pendiente” (cadena de hasta 15 caracteres), obligatorio, por defecto a la hora de asignar candidatura estará en pendiente.
- user_id → Identificador del usuario al que pertenece la candidatura (hasta 4 cifras), obligatorio
- empresa_id → Identificador de la empresa a la que se ha realizado la candidatura (hasta 4 cifras), obligatorio

2.3.2.- *Requisitos funcionales.*

Por un lado, tenemos un perfil de docente que puede:

- Gestionar las empresas, con sus distintas sedes.
- Gestionar los usuarios, tanto alumnado como docentes.
- Asignar a los distintos alumnos a distintas empresas.
- Gestión de las candidaturas de los alumnos.

Por otra parte, tendremos un perfil de alumno que:

- Gestionar su perfil.
- Subir su CV sólo en formato PDF.
- Ver el estado de sus candidaturas.

2.3.3.- *Otros Requisitos*

Requisitos de escalabilidad: La aplicación será capaz de manejar un gran número de usuarios y solicitudes a medida que el sistema crezca.

Requisitos de concurrencia: La aplicación debe ser capaz de manejar múltiples solicitudes de usuarios simultáneamente sin disminuir la velocidad o la calidad del servicio.

Requisitos de compatibilidad: La aplicación debe ser compatible con diferentes navegadores y sistemas operativos.

Requisitos de Seguridad: La aplicación debe de asegurar la confidencialidad de los datos mediante el uso de medidas de seguridad tales como la encriptación de datos y autenticación de usuarios.

2.4.- Descripción de la solución

Pensamos en gestionar todo esto a través de una web que simplificará la forma de hacerlo, en la cual, los docentes gestionan a los alumnos y a las distintas empresas en las que se pueden realizar la fct, teniendo un registro de las candidaturas de los alumnos, y los alumnos que podrán gestionar sus datos y ver sus candidaturas a tiempo real.

2.5.- Planificación del proyecto

2.5.1.- *Equipo de trabajo*

El equipo cuenta con 6 personas que pueden desarrollar uno o más de los siguientes roles:

1. Líder de proyecto: responsable de liderar y coordinar el equipo de proyecto, así como de asegurar el cumplimiento de los plazos y los requisitos del proyecto. Se encargará de la gestión de riesgos y de tomar decisiones estratégicas para el proyecto.
2. Desarrollador de back-end: responsable del desarrollo de la API REST y la base de datos MySQL en phpmyadmin.
3. Desarrollador de front-end: responsable del desarrollo de la interfaz de usuario utilizando Angular. Se encargará de la integración de la API REST en la interfaz de usuario.
4. Diseñador gráfico: responsable de la creación de los diseños de la interfaz de usuario.
5. Especialista en seguridad: responsable de garantizar la seguridad de la aplicación y de protegerla contra posibles vulnerabilidades.
6. Especialista en pruebas: responsable de realizar pruebas de la aplicación para asegurar su calidad y su cumplimiento de los requisitos.

Decir que la documentación la realizará cada desarrollador a la hora de crear su parte, además de la seguridad que la implementaran los mismos desarrolladores.

2.5.2.- *Planificación temporal*

Se intentará seguir la siguiente planificación:

1. Definición de requisitos y alcance (2 semanas)
 - a. Identificar los requisitos y el alcance del proyecto en detalle.
 - b. Establecer los plazos y los objetivos.
2. Diseño de la arquitectura y la base de datos (2 semanas)
 - a. Diseñar la arquitectura de la aplicación y la base de datos.
 - b. Establecer las tecnologías a utilizar.
3. Desarrollo del back-end (6 semanas)
 - a. Implementar la API REST y la base de datos.
 - b. Realizar pruebas y depuración.
4. Desarrollo del front-end (6 semanas)
 - a. Diseñar y desarrollar la interfaz de usuario utilizando Angular.
 - b. Integrar la API REST en la interfaz de usuario.
 - c. Realizar pruebas y depuración.
5. Pruebas y control de calidad (2 semanas)
 - a. Realizar pruebas exhaustivas de la aplicación.
 - b. Identificar y corregir errores y problemas.

6. Documentación y entrega del proyecto (1 semana)
 - a. Documentar el código y los procesos relevantes.
 - b. Realizar la entrega del proyecto.

2.6.- Estudio del coste del proyecto

El coste del proyecto sería más o menos el siguiente, decir que los desarrolladores hacen bastantes horas ya que no solo se encarga de desarrollar, sino que también documentan e implementan la seguridad.

1. Líder de proyecto: Considerando un salario de 18€/hora y 80 horas dedicadas al proyecto.
2. Desarrollador de back-end y desarrollador de front-end: Considerando un salario de 14€/hora y 280 horas dedicadas cada uno al proyecto.
3. Diseñador gráfico: Considerando un salario de 11€/hora y 80 horas dedicadas al proyecto.
4. Especialista en pruebas: Considerando un salario de 11€/hora y 80 horas dedicadas al proyecto.

Costo total = (18€ x 80 horas) + (14€ x 280 horas) + (14€ x 280 horas) + (11€ x 80 horas) + (11€ x 80 horas)

Costo total = 1.440€ + 3.920€ + 3.920€ + 880€ + 880€

Costo total = 11.040€

Para los sueldos he buscado el puesto en una plataforma conocida como [glassdoor](https://www.glassdoor.es) que te da el sueldo medio al año, y con ello calculado cuanto es la hora.

3.- Análisis del Sistema de Información

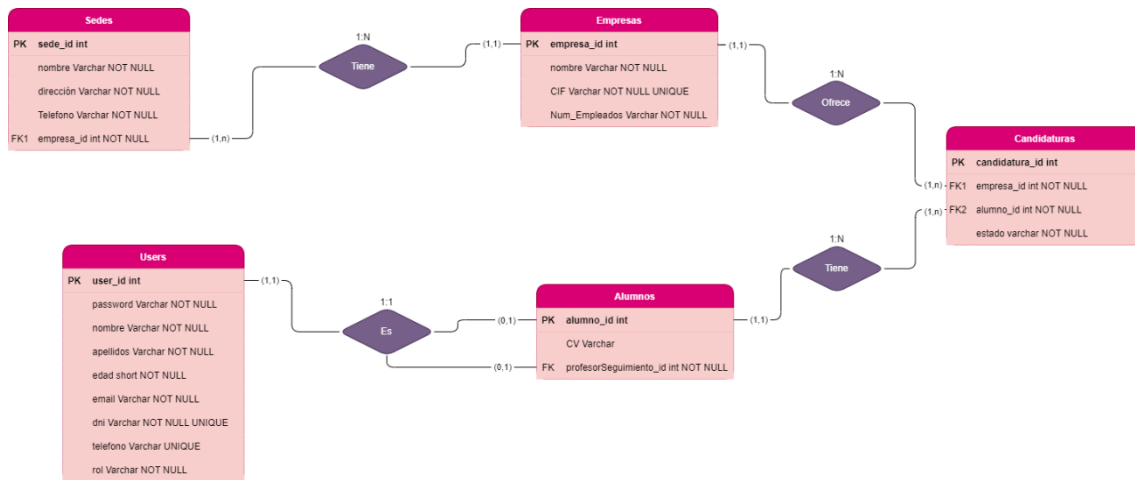
3.1.- Identificación del entorno tecnológico

Los componentes tecnológicos que se utilizarían en el proyecto:

1. Servidor web: Apache (lanzado desde XAMPP)
2. Lenguaje de programación: php (para el back-end)
3. Framework de desarrollo web: Laravel (para el back-end)
4. Gestor de bases de datos: MySQL (para el back-end)
5. Herramienta de gestión de la base de datos: phpMyAdmin (para el back-end)
6. Herramienta de pruebas de API: Postman (para el back-end)
7. Framework de desarrollo de aplicaciones web: Angular (para el front-end)
8. Lenguaje de marcado: HTML (para el front-end)
9. Hojas de estilo en cascada: CSS (para el front-end)
10. Lenguaje de programación: TypeScript (para el front-end)

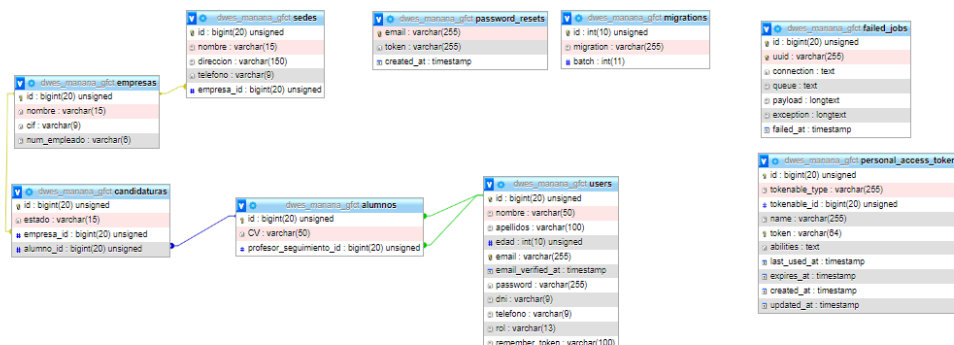
3.2.- Modelado de datos

3.2.1.- Modelo Entidad-Relación



He utilizado draw.io para realizar el diagrama (adjunto en la carpeta draw.io).

3.2.2.- Esquema de la base de datos



Esta Base de datos se obtiene de realizar las migraciones desde Laravel (Completamente necesario realizarlo y después añadir mínimo un usuario como docente, utilizar el de Datos de Prueba).

3.2.3.- Datos de prueba

```
/*-----
Insert Table USERS
-----*/

INSERT into
USERS(ID,PASSWORD,NOMBRE,APELLIDOS,EDAD,EMAIL,DNI,TELEFONO,ROL)
values(1,"$2y$10$Re46YodTG08r2rYiks641uvshrwSe5RkdTu70Phr/H0v4FQLJL3GG",
"manuel", "Ruiz Jiménez", 21, "manuelruiz@email.com", "77834164S",
"67584316S", "docente");
```

```
INSERT into
USERS(ID,PASSWORD,NOMBRE,APELLIDOS,EDAD,EMAIL,DNI,TELEFONO,ROL)
values(2,"$2y$10$Re46YodTG08r2rYiks641uvshrwSe5RkdTu70Phr/H0v4FQLJL3GG",
"Prueba", "Jiménez", 21, "Prueba@email.com", "74334164S", "656782265",
"alumno");

/*-----
      La contraseña es: 123456
-----*/

/*-----
      Insert Table EMPRESAS
-----*/

INSERT into EMPRESAS(ID,NOMBRE,CIF,NUM_EMPLEADO)
values(1,"empresa","B76345879",1000);

/*-----
      Insert Table ALUMNOS
-----*/

INSERT into ALUMNOS(ID,CV,PROFESOR_SEGUIMIENTO_ID) values(2,null,1);

/*-----
      Insert Table SEDES
-----*/

INSERT into SEDES(ID,NOMBRE,DIRECCION,TELEFONO,EMPRESA_ID)
values(1,"Desarrollo","C/ Prueba Nª3","546827965",1);

/*-----
      Insert Table CANDIDATURAS
-----*/

INSERT into CANDIDATURAS(ID,ESTADO,ALUMNO_ID,EMPRESA_ID)
values(1,"pendiente",2,1);
```

3.3.- Identificación de los usuarios participantes y finales

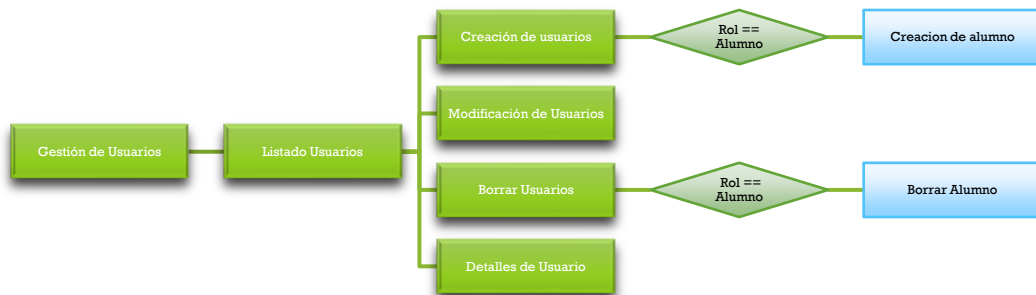
- Alumnos que deben realizar la FCT (solo gestionaran sus datos, subirán su CV y revisaran sus candidaturas).

- Profesores encargados de la gestión de la FCT en la institución educativa (Gestionaran todos los datos, desde el alta de una empresa u otro usuario, hasta el hecho de asignar una empresa a un alumno en su lista de candidaturas).

3.4.- Identificación de subsistemas de análisis

Para usuario docente:

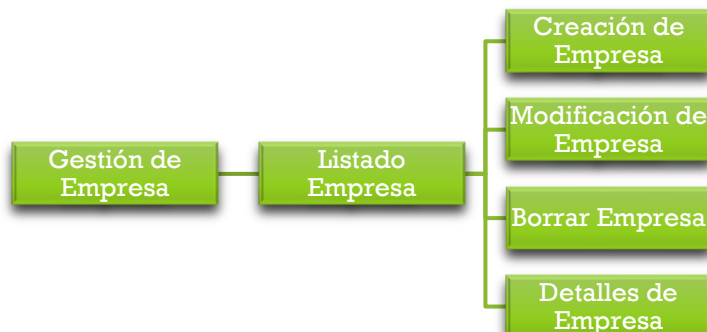
1. Gestión de Usuarios → Subsistema encargado de gestionar todo lo relacionados con todo tipo de usuarios.



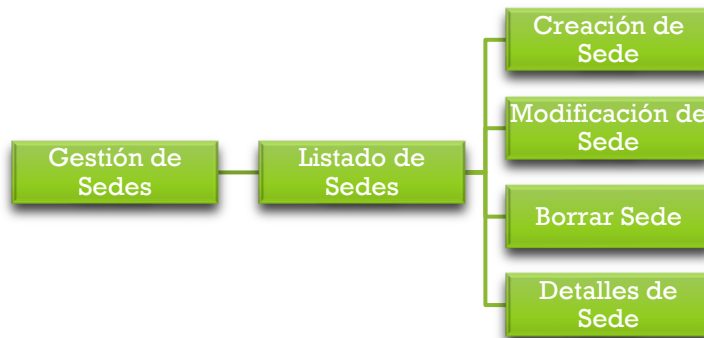
2. Gestión de Alumnos → Subsistema encargado de gestionar información concreta de los alumnos como el CV y el Id del profesor de seguimiento.



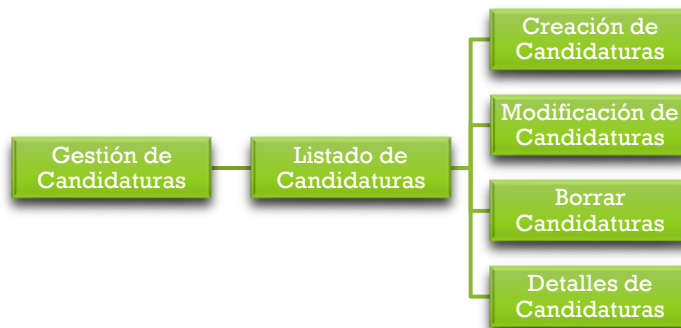
3. Gestión de Empresas → Subsistema encargado de gestionar todo lo relacionados con las empresas.



4. **Gestión de Sedes** → Subsistema encargado de gestionar todo lo relacionado con las sedes de cada empresa.



5. **Gestión de Candidaturas** → Subsistema encargado de gestionar la información de las candidaturas de los alumnos.



Los usuarios Alumnos solo utilizarán algunas funciones de estos subsistemas, las cuales son:

-Ver y modificar datos de su usuario, ya sean datos generales o específicos como puede ser subir su Curriculum vitae, aunque no podrá cambiar al profesor de seguimiento.

-Listado de sus propias candidaturas

3.5.- Establecimiento de requisitos

Para las gestiones se deberá tener en cuenta el rol del usuario.

Gestiones:

1. Usuarios:

a. Como profesor:

Lo primero será listar los diferentes usuarios ordenados primero por profesores y después por alumnos, desde la lista podrá ver detalles del usuario, modificar la información, borrar al usuario o crear uno nuevo.

b. Como alumno:

Podrá ver y editar sus datos personales.

2. Alumnos

a. Como profesor:

Lo primero será listar los diferentes alumnos, pudiendo filtrar por los alumnos a los que hace el seguimiento, en la lista podrá seleccionar los alumnos y ver su información o también modificar la información del alumno seleccionado.

b. Como alumno:

Podrá ver y editar sus datos personales.

3. Empresas

a. Como profesor:

Lo primero será listar las diferentes empresas, desde la lista podrá ver detalles de la empresa, modificar la información, borrar la empresa o dar de alta a una.

4. Sedes

a. Como profesor:

Lo primero será listar las diferentes Sedes ordenadas por la empresa a la que pertenece, y desde la lista podrá ver detalles de la sede, modificar la información, borrar la sede o dar de alta a una.

5. Candidaturas

a. Como profesor:

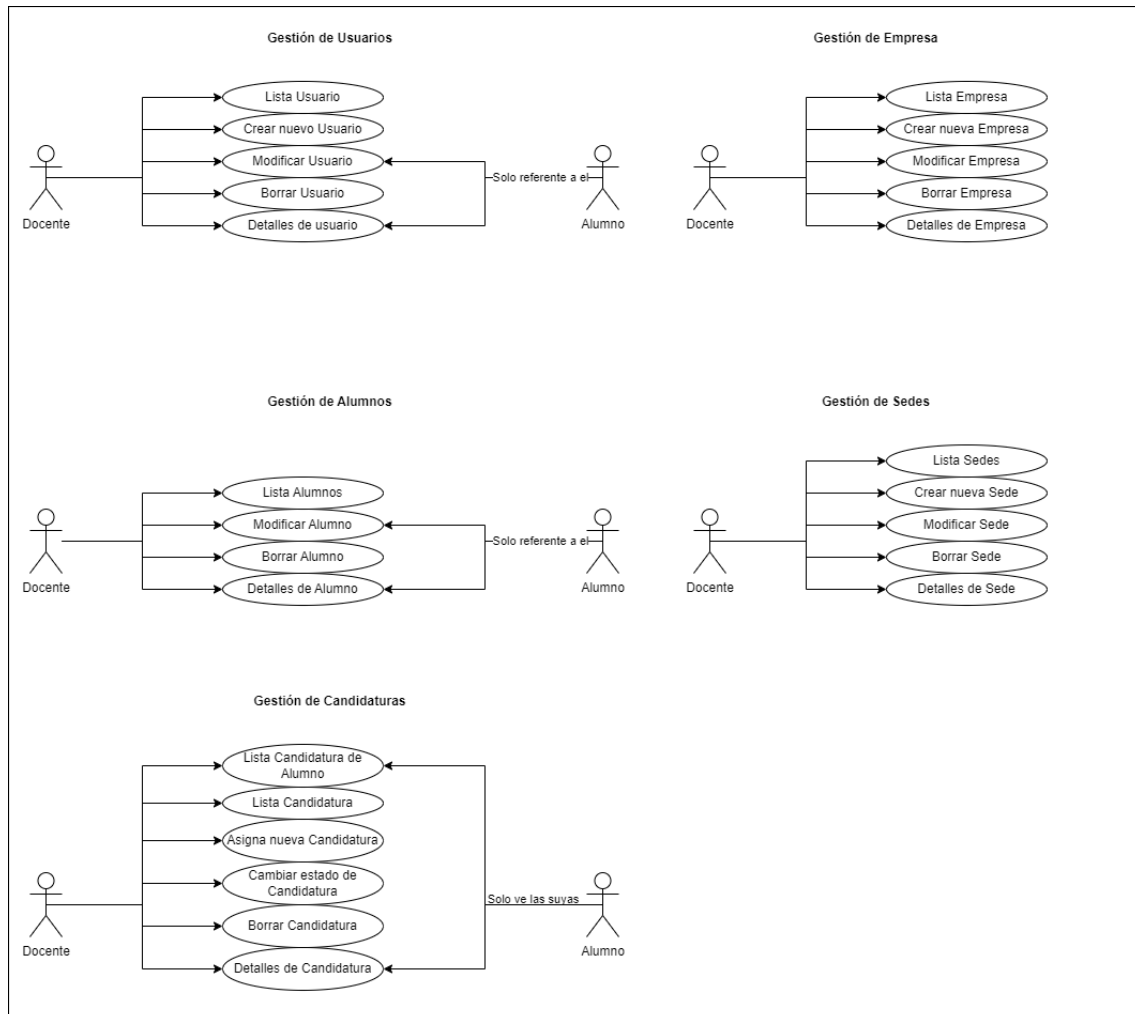
Podrá ver todas las candidaturas, pudiendo filtrar por la de los alumnos a quien realiza el seguimiento, además podrá asignar una nueva candidatura a un alumno o también podrá cambiar el estado de la candidatura a aceptada o denegada.

b. Como alumno:

Solo vera sus propias candidaturas y los detalles de las mismas.

3.6.- Diagramas de Análisis

Yo he utilizado un diagrama de caso de uso:



Realizado desde la herramienta draw.io (adjunto en la carpeta draw.io).

3.7.- Definición de interfaces de usuario

3.7.1.- Especificación de principios generales de interfaz

Mi Interfaz se basa en un estilo minimalista, utilizando colores claros para mantener una apariencia limpia y sin distracciones

3.7.2.- Especificación de formatos individuales de la interfaz de pantalla

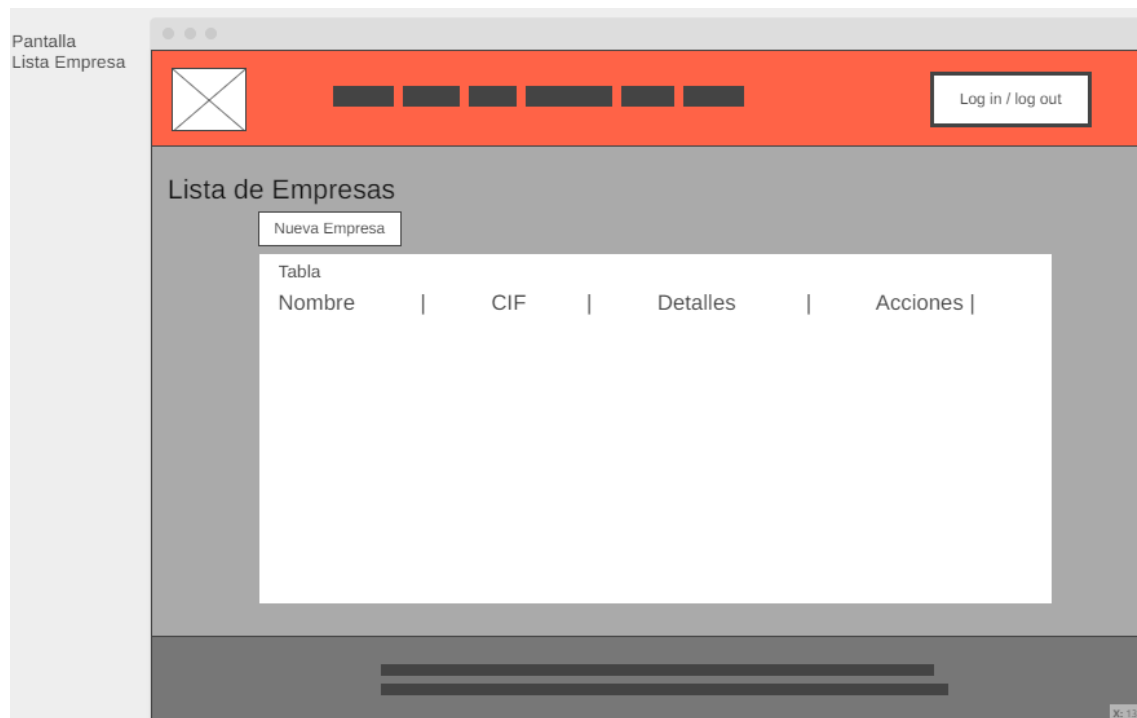
Listados:

Listado usuario



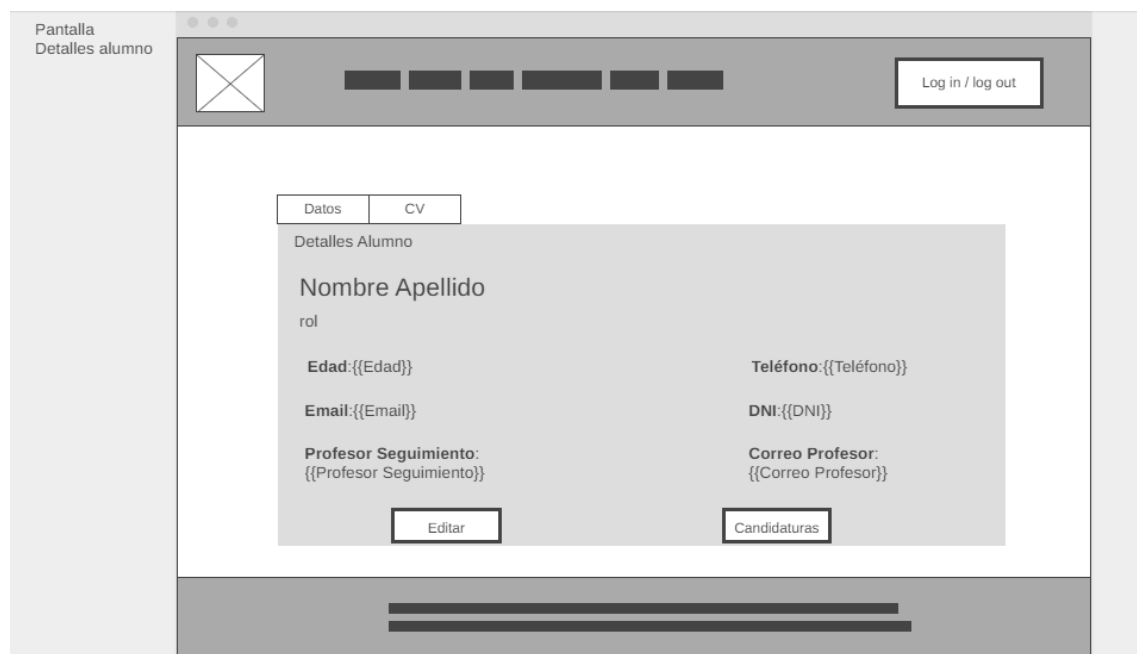
Otros listados:

(todos siguen este formato, pero con sus propios datos como es normal)



Detalles:

La sección de detalles será con un bloque en el centro y según donde sea tendrá pestañas que contendrán datos de otras tablas



En este caso tendría CV para ver el curriculum del alumno, además este en concreto tiene un botón extra que nos llevan a la lista de candidaturas del alumno en concreto del que estamos viendo los datos

Update:

Esta sección será similar a la anterior salvo que sustituiremos el contenido por un formulario donde vendrá cada parámetro actual del usuario para que ahora lo edites

Pantalla
Update alumno

Actualizar Usuario

Nombre

Apellidos

Edad

Email

DNI

Teléfono

Guardar Cancelar

Create:

Esta sección será similar a la anterior salvo que los campos de los datos vienen vacios

Pantalla
Create usuario

Registrar Usuario

Nombre

Apellidos

Password

Email

DNI

Teléfono

Edad

Rol

Guardar Cancelar

Login:

The mockup shows a web browser window titled 'Pantalla Login'. The header bar is grey and contains a 'Log in / log out' button on the right. The main content area is white and features a centered login form. The form consists of three stacked input fields: the first is labeled 'Email', the second is labeled 'Password', and the third is a button labeled 'Login'. The footer bar is grey and contains two horizontal lines.

3.7.3.- Identificación de perfiles de usuario

Alumno:

Solo podrá acceder a los detalles de su propio usuario, a la lista de sus candidaturas y los detalles de las mismas

Docente:

Podrá ver los todas las interfaces que dispone la web

3.7.4.- Especificación de formatos de impresión

El único formato disponible para la impresión será pdf ya que los únicos archivos admitidos serán de dicho formato

3.7.5.- Especificación de la navegabilidad entre pantallas

Alumno:

Cuenta con un header al que le aparece solo los enlaces de sus datos y sus candidaturas, y a través de esas pantallas se mueve a la actualización de sus datos y a los detalles de sus candidaturas.

Docente:

Cuenta con un header al que aparece los enlaces a los distintos listados, y a través de esas pantallas se mueve a todas las demás.

4.- Construcción del Sistema

Aquí tenemos alojado el repositorio de GitHub https://github.com/ManuelRuji/PI-Gestion_FCT

Requisitos

Para utilizar este proyecto necesitaremos una serie de herramientas:

- Node --> Tenerlo instalado para cuando sea necesario realizar comandos npm. <https://nodejs.org/es/download>
- XAMP --> Lo utilizaremos por su BBDD MySQL que deba tener cargadas las migraciones de Laravel. <https://www.apachefriends.org/es/download.html>
- Composer --> lo utilizaremos para iniciar la parte del back con Laravel (comando 'php artisan serve') <https://getcomposer.org/download/>
- Angular CLI --> Lo utilizaremos para iniciar la parte del front con Angular (comando 'ng serve')

```
npm install -g @angular/cli
```

Uso

iniciamos el módulo de MySQL de XAMP

Dentro de la ruta API/GFCT_API ejecutamos

```
php artisan serve
```

Dentro de la ruta Front(Angular)/GFCT_Front ejecutamos

```
ng serve
```

Al hacerlo se ejecutará el proyecto en <http://localhost:4200/>.

5.- Conclusiones (otros usos, tareas sin finalizar, problemas encontrados y solventados, que he aprendido...)

Problemas encontrados:

Tuve bastantes problemas en el login ya que no se guardaba bien el Token, lo solucione editando dos métodos del front con el que guardo ahora el Token.

Además, a la hora de solicitar un listado aparece un console.error, no influye en el uso de la aplicación, y no me ha dado tiempo a arreglarlo.

Roadmap:

Para las siguientes versiones tendremos las siguientes nuevas características:

- Mayor estilo CSS (aunque seguiría siendo minimalista).
- implementar editar el profesor de seguimiento.
- Rework completo de como se ve y funciona la sección de sedes.
- Añadir filtros para buscar los datos en cada sección

Por último, decir que este proyecto me ha servido bastante para reforzar lo que sabia de Laravel y con el que he podido aprender a utilizar Angular.

6.- Glosario de términos

API: Conjunto de reglas para que las aplicaciones se comuniquen entre sí.

Front-end: Parte visible y con la que interactúan los usuarios en una aplicación o sitio web.

Back-end: Parte no visible que maneja la lógica y el almacenamiento de datos de una aplicación o sitio web.

Mock up: Representación visual de un diseño antes de ser desarrollado.

Roadmap: Planificación estratégica de un proyecto o producto.

Rework: Modificación de una parte ya existente de un proyecto o producto.

Log in: Acceder a una cuenta o sistema mediante autenticación.

Log out: Finalizar una sesión en una cuenta o sistema.

Header: Sección superior de una página web o mensaje de solicitud HTTP.

7.- Bibliografía

<https://app.diagrams.net/>

<https://chat.openai.com/>

<https://www.glassdoor.es/member/home/index.htm>

<https://danielk.tech/home/angular-how-to-add-a-loading-spinner>

<https://cssloaders.github.io/>

<https://getbootstrap.com/docs/5.3/examples/>

<https://bootstrapcreative.com/resources/bootstrap-5-cheat-sheet-classes-index/>