

A thick dark green vertical bar runs down the left side of the page. A light green arrow points to the right from this bar, containing the date. Below the arrow, several thin, curved lines in dark green and light grey sweep upwards from the bottom left corner.

20-4-2023

# Documentación Proyecto Integrado

2ºDAW

Manuel Ruiz Jiménez  
IES VELAZQUEZ

# Índice

<b>DOCUMENTACIÓN DEL PROYECTO</b>	<b>1</b>
<b>1.- INTRODUCCIÓN</b>	<b>1</b>
<b>2.- ESTUDIO DE VIABILIDAD</b>	<b>1</b>
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
<b>3.- ANÁLISIS DEL SISTEMA DE INFORMACIÓN</b>	<b>5</b>
3.1.- IDENTIFICACIÓN DEL ENTORNO TECNOLÓGICO	5
3.2.- MODELADO DE DATOS	6
3.3.- IDENTIFICACIÓN DE LOS USUARIOS PARTICIPANTES Y FINALES	8
3.4.- IDENTIFICACIÓN DE SUBSISTEMAS DE ANÁLISIS	8
3.5.- ESTABLECIMIENTO DE REQUISITOS	10
3.6.- DIAGRAMAS DE ANÁLISIS	11

# 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 Spring Boot, 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

Ahora mismo solo se gestiona los datos manualmente a través de hojas de cálculo

### 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
- nickname→nombre único de cada usuario (cadena de hasta 20 caracteres), obligatorio
- passwd → contraseña para inicio de sesión (cadena de hasta 100 caracteres), 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
- 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:

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

Por otra parte, tendremos un perfil de alumno que:

- Podrá gestionar su perfil.
- Podrá subir su CV sólo en formato PDF.
- Verá el estado de sus candidaturas.

### 2.3.3.- *Otros Requisitos*

**Requisitos de escalabilidad:** La aplicación debe ser capaz de manejar un gran número de usuarios y transacciones a medida que el sistema crezca. Esto significa que debe ser escalable y tener la capacidad de aumentar el número de servidores o recursos del sistema según sea necesario.

**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 1 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. Se encargará de la gestión de datos y de la lógica de negocio de la aplicación.
3. Desarrollador de front-end: responsable del desarrollo de la interfaz de usuario utilizando Angular. Se encargará de la experiencia del usuario y 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 y de la experiencia del usuario de la aplicación.
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 exhaustivas de la aplicación para asegurar su calidad y su cumplimiento de los requisitos.
7. Especialista en documentación: responsable de documentar el código y los procesos relevantes para que otros desarrolladores puedan trabajar en el proyecto en el futuro.
8. Especialista en implementación: responsable de la implementación y el despliegue de la aplicación en el servidor.

Decir que la documentación la realizará cada desarrollador a la hora de crear su parte, además la seguridad la implementaran los mismos desarrolladores, salvo la seguridad del servidor que la realizará el especialista en implementación.

### 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: 50€/hora, dedicará 80 horas.
2. Desarrollador de back-end: 40€/hora, dedicará 280 horas.
3. Desarrollador de front-end: 40€/hora, dedicará 280 horas.
4. Diseñador gráfico: 35€/hora, dedicará 80 horas.
5. Especialista en pruebas: 35€/hora, dedicará 80 horas.
6. Especialista en implementación: 40€/hora, dedicará 40 horas.

Costo total = (50€ x 80 horas) + (40€ x 280 horas) + (40€ x 280 horas) + (35€ x 80 horas) + (35€ x 80 horas) + (40€ x 40 horas)

Costo total = 4.000€ + 11.200€ + 11.200€ + 2.800€ + 2.800€ + 1.600€

Costo total = 33.600€

## 3.- Análisis del Sistema de Información

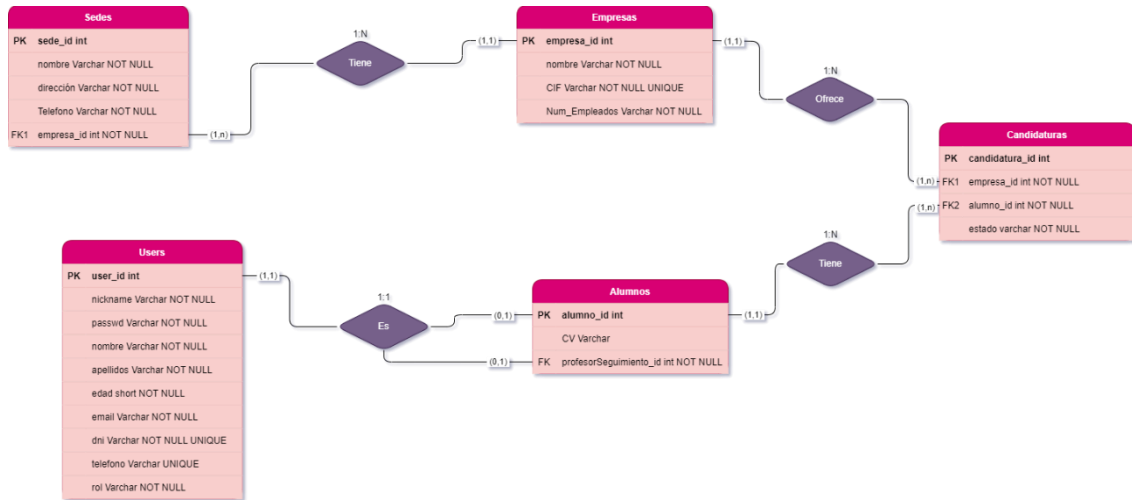
### 3.1.- Identificación del entorno tecnológico

Los componentes tecnológicos que se utilizarían en el proyecto:

1. Servidor web: Apache (lanzado desde XAMPP)
2. Lenguaje de programación: Java (para el back-end)
3. Framework de desarrollo web: Spring Boot (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)

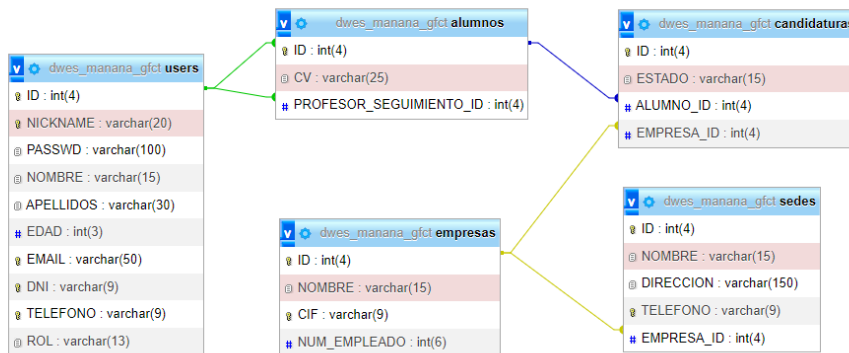
## 3.2.- Modelado de datos

### 3.2.1.- Modelo Entidad-Relación



He utilizado [draw.io](https://draw.io) para realizar el diagrama (adjunto en la carpeta draw.io).

### 3.2.2.- Esquema de la base de datos



He adjuntado el SQL tanto de creación como alteración e inserción (adjunto en la carpeta sql).



### 3.2.3.- Datos de prueba

```
/*-----  
    Insert Table USERS  
-----*/  
  
INSERT into  
USERS(ID,NICKNAME,PASSWD,NOMBRE,APELLIDOS,EDAD,EMAIL,DNI,TELEFONO,ROL)  
values(1,"admin","admin123","manuel","Ruiz  
Jimenez",21,"manuelruizjimenezmrj01@gmail.com","77834164S","675843165",  
"docente");  
  
INSERT into  
USERS(ID,NICKNAME,PASSWD,NOMBRE,APELLIDOS,EDAD,EMAIL,DNI,TELEFONO,ROL)  
values(2,"estudiante","est123","prueba","prueba_apellido",21,"prueba@gmai  
l.com","74832164L","675751234","alumno");  
  
/*-----  
    Insert Table EMPRESAS  
-----*/  
  
INSERT into EMPRESAS(ID,NOMBRE,CIF,NUM_EMPLEADO)  
values(1,"empresa","B76345879",10000);  
  
/*-----  
    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 N°3","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

#### 1. Usuarios 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).

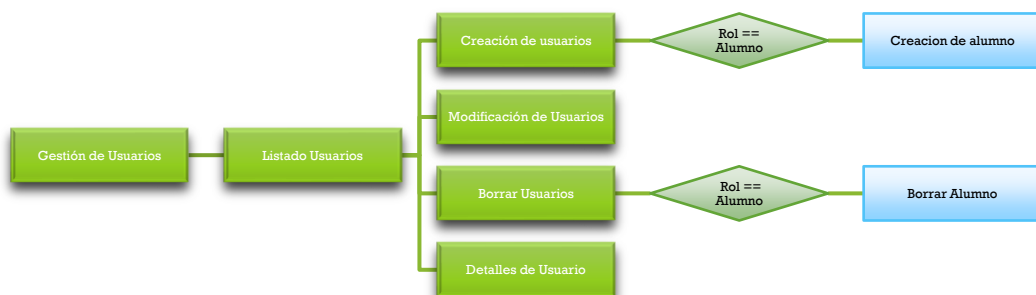
#### 2. Usuarios participantes:

- Desarrolladores de back-end
- Desarrolladores de front-end
- Diseñadores de interfaz de usuario

### 3.4.- Identificación de subsistemas de análisis

Para usuario docente:

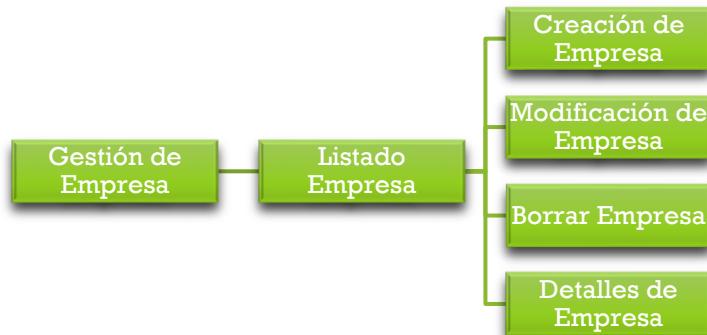
1. Gestión de Usuarios → Subsistema encargado de gestionar todo lo relacionados con todo tipo de usuarios.



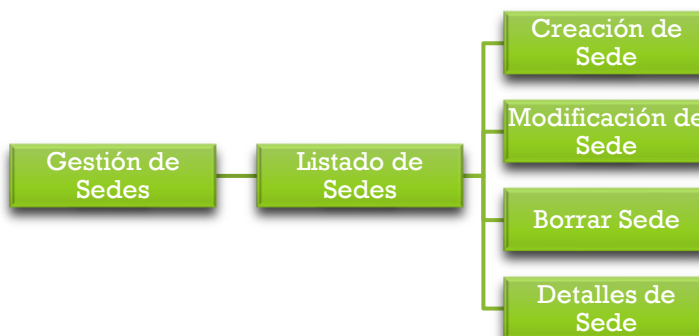
2. Gestión de Alumnos → Subsistema encargado de gestionar información concreta de los alumnos como el CV y el Id del profesor de seguimiento.



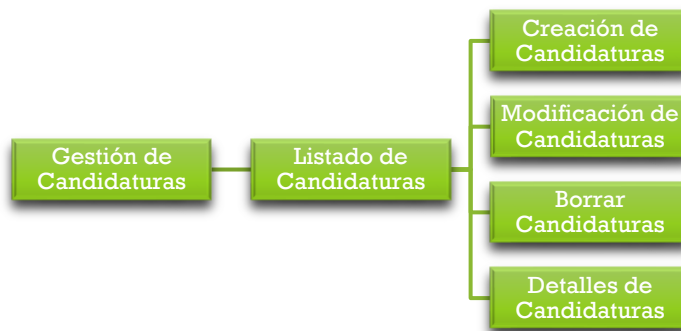
3. **Gestión de Empresas** → Subsistema encargado de gestionar todo lo relacionados con las empresas.



4. **Gestión de Sedes** → Subsistema encargado de gestionar todo lo relacionados 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 o 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:

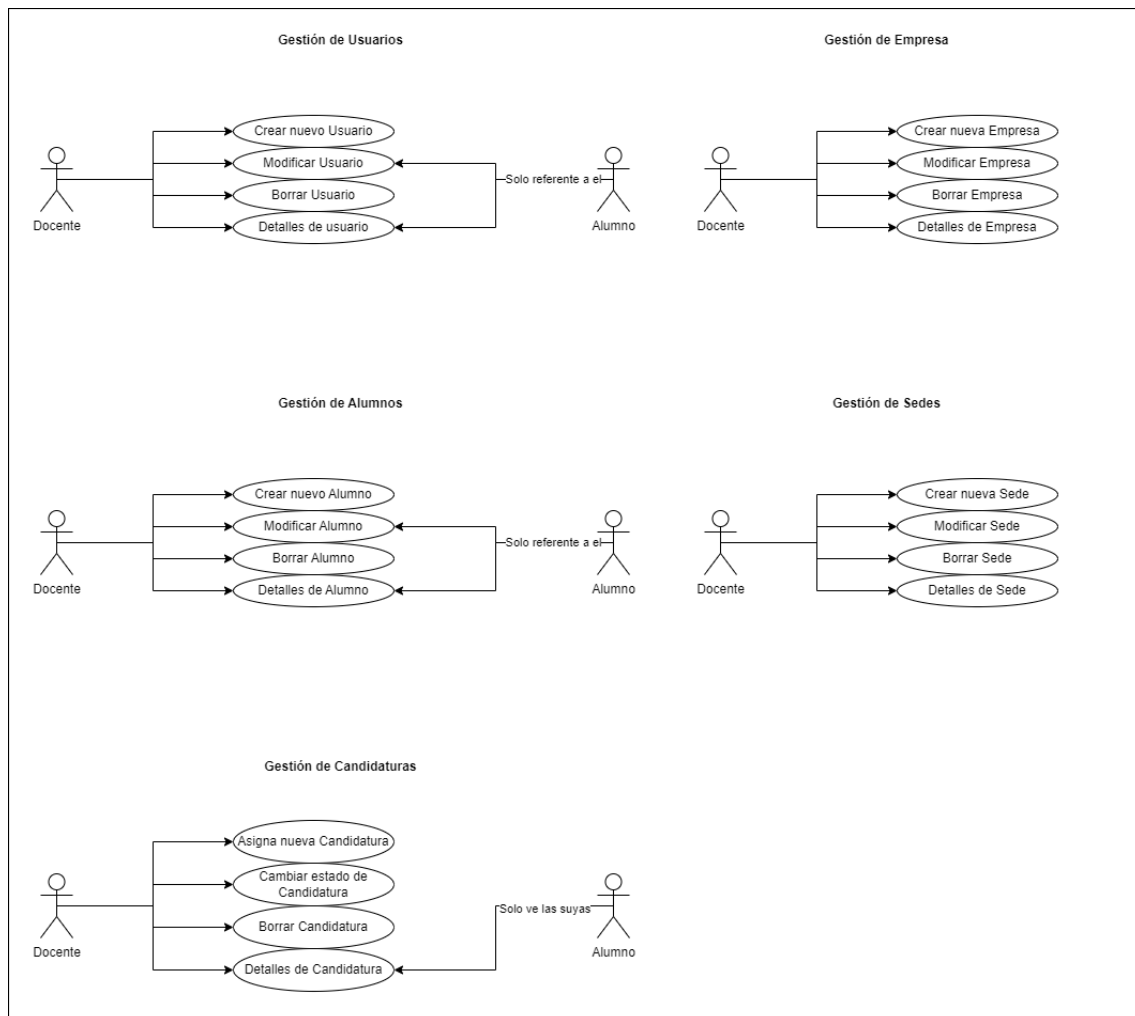
Lo primero será listar las diferentes Sedes ordenadas por la empresa a la que pertenece, y desde la lista podrá ver detalles de la sede, modificar la información, borrar la sede o dar de alta a una.
5. Candidaturas
  - a. Como profesor:

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.

## 3.6.- Diagramas de Análisis

Yo he utilizado un diagrama de caso de uso:



Realizado desde la herramienta [draw.io](https://draw.io) (adjunto en la carpeta draw.io).