# PROYECTO INTEGRADO DAW

# FCT GESTIÓN APP



***Miguel Mondéjar González***

***Proyecto Integrado Ciclo Superior Desarrollo de Aplicaciones Web***

***IES Velázquez – Curso 2022/23***

ÍNDICE.

[**PROYECTO INTEGRADO DAW** 1](#_Toc137129911)

[FCT GESTIÓN APP 1](#_Toc137129912)

[Lista de Cambios 4](#_Toc137129913)

[1.- Introducción 5](#_Toc137129914)

[2.- Estudio de Viabilidad 6](#_Toc137129915)

[2.1.- Descripción del Sistema Actual 6](#_Toc137129916)

[2.2.- Descripción del Sistema Nuevo 6](#_Toc137129917)

[2.3.- Identificación de Requisitos del Sistema 6](#_Toc137129918)

[2.3.1.- Requisitos de información 6](#_Toc137129919)

[2.3.2.- Requisitos funcionales. 7](#_Toc137129920)

[2.3.3.- Otros Requisitos 8](#_Toc137129921)

[2.4.- Descripción de la solución 8](#_Toc137129922)

[2.5.- Planificación del proyecto 9](#_Toc137129923)

[2.5.1.- Equipo de trabajo 9](#_Toc137129924)

[2.5.2.- Planificación temporal 9](#_Toc137129925)

[2.6.- Estudio del coste del proyecto 9](#_Toc137129926)

[3.- Análisis del Sistema de Información 11](#_Toc137129927)

[3.1.- Identificación del entorno tecnológico 11](#_Toc137129928)

[3.2.- Modelado de datos 12](#_Toc137129929)

[3.2.1.- Modelo Entidad-Relación 12](#_Toc137129930)

[3.2.2.- Esquema de la base de datos 13](#_Toc137129931)

[3.2.3.- Datos de prueba 13](#_Toc137129932)

[3.3.- Identificación de los usuarios participantes y finales 14](#_Toc137129933)

[3.4.- Identificación de subsistemas de análisis 14](#_Toc137129934)

[3.5.- Establecimiento de requisitos 15](#_Toc137129935)

[3.6.- Diagramas de Análisis 16](#_Toc137129936)

[3.7.- Definición de interfaces de usuario 17](#_Toc137129937)

[3.7.1.- Especificación de principios generales de interfaz 17](#_Toc137129938)

[3.7.2.- Especificación de formatos individuales de la interfaz de pantalla 18](#_Toc137129939)

[3.7.3.- Identificación de perfiles de usuario 26](#_Toc137129940)

[3.7.4.- Especificación de formatos de impresión 27](#_Toc137129941)

[3.7.5.- Especificación de la navegabilidad entre pantallas 27](#_Toc137129942)

[4.- Construcción del Sistema 29](#_Toc137129943)

[5.- Conclusiones (otros usos, tareas sin finalizar, problemas encontrados y solventados, que he aprendido...) 30](#_Toc137129944)

[6.- Glosario de términos 31](#_Toc137129945)

[7.- Bibliografía 32](#_Toc137129946)

[Apéndices 33](#_Toc137129947)

# Lista de Cambios.

30/04/2023

* Cambios en la Base de datos.
  + Longitud contraseña (usuarios) y localidad (sedes).

20/05/2023

* Cambios en la Base de datos.
  + Eliminación de la columna email en Sedes.
  + Añadido ON CASCADE DELETE en Candidaturas y Sedes.
  + Añadido UNIQUE en email de Usuarios y usuario\_id de Curriculums.

01/06/2023

* Cambios en la Base de datos.
  + Eliminación de fecha\_inicio y fecha\_fin.

# 1.- Introducción.

Mi proyecto integrado para el ciclo superior de Desarrollo de Aplicaciones Web consiste en la creación de una página web para la gestión de la FCT. La idea principal es diseñar una plataforma que facilite la gestión de esta formación por parte de los usuarios, que se espera que sean principalmente estudiantes y profesores.

Para llevar a cabo el proyecto, utilizaré varios lenguajes. En primer lugar, utilizaré HTML, CSS y Bootstrap, un framework de diseño de páginas web, para crear una página web responsiva y atractiva con una estética moderna y elegante.

Para la persistencia de datos utilizaré MySQL, que es un sistema de gestión de bases de datos relacionales muy popular y ampliamente utilizado en el mundo del desarrollo web.

Para la parte servidor (backend) utilizaré Laravel, que es un framework de código abierto basado en PHP, con el cual gestionaré principalmente la creación de la API. Laravel me ayudará a implementar funciones avanzadas, como autenticación de usuarios y gestión de bases de datos.

Por último, para la parte cliente (frontend) utilizaré JavaScript, que es un lenguaje de programación de alto nivel y dinámico que se utiliza principalmente para crear interactividad en las páginas web. Con este lenguaje, podré crear animaciones, validar formularios, agregar funciones dinámicas a la página y hacer peticiones a la API.

# 2.- Estudio de Viabilidad.

## 2.1.- Descripción del Sistema Actual.

Actualmente, en nuestro centro educativo se gestiona la FCT de manera manual. Los profesores encargados de esta tarea contactan con las empresas colaboradoras a través de correo electrónico o formularios para la asignación de un alumno a una empresa.

El alumno que es contactado por una empresa no sabe el estado de su candidatura hasta que un docente o la propia empresa le contacte, haciendo que sea complicado para el estudiante el seguimiento de su proceso de selección.

## 2.2.- Descripción del Sistema Nuevo.

La nueva plataforma para la gestión de la FCT permitirá a los docentes buscar y solicitar prácticas de forma sencilla para los alumnos, permitirá una gestión más eficiente y rápida. Los profesores tendrán acceso a una interfaz en la que podrán asignar a un alumno con una empresa. El sistema permitirá a los estudiantes hacer un seguimiento del progreso de la candidatura a través de una interfaz en línea. Además, los usuarios podrán actualizar sus datos, el usuario con rol alumno podrá subir su CV.

En resumen, el sistema nuevo permitirá la automatización de procesos y la eliminación de los procesos administrativos manuales en términos de tiempo y recursos.

## 2.3.- Identificación de Requisitos del Sistema.

### 2.3.1.- Requisitos de información.

Se necesitará almacenar en nuestro sistema la siguiente información:

Datos de los estudiantes.

* Descripción: Información personal de los docentes.
* Campos: nombre, apellidos, fecha de nacimiento, DNI, correo electrónico, teléfono y contraseña.
* Tipo de datos: Texto, fecha, numérico.
* Otras observaciones: se debe permitir la modificación de los datos personales por parte del docente.

Datos de los estudiantes.

* Descripción: Información personal de los estudiantes, incluyendo su CV.
* Campos: nombre, apellidos, fecha de nacimiento, DNI, correo electrónico, teléfono, contraseña y CV.
* Tipo de datos: Texto, fecha, numérico.
* Otras observaciones: se debe permitir la modificación de los datos personales por parte del estudiante y docente.

Datos de las empresas.

* Descripción: Información sobre las empresas que ofrecen formación en centros de trabajo.
* Campos: nombre de la empresa, CIF y correo electrónico.
* Tipo de datos: Texto, numérico.
* Otras observaciones: se debe permitir la modificación de los datos de la empresa por parte del docente.

Datos de las sedes.

* Descripción: Información sobre las sedes de una empresa.
* Campos: nombre de la sede, dirección, localidad, provincia, código postal, teléfono y correo electrónico.
* Tipo de datos: Texto, numérico.
* Otras observaciones: se debe permitir la modificación de los datos de la sede por parte del docente.

Datos de las candidaturas.

* Descripción: Información sobre las candidaturas de los estudiantes a las empresas, incluyendo el estado de la candidatura (pendiente, aceptada o rechazada), la empresa solicitada, etc.
* Campos: estado de la candidatura, empresa solicitada y alumno.
* Tipo de datos: Texto, fecha, numérico.
* Otras observaciones: se debe permitir la modificación del estado de la candidatura por parte del docente.

Datos del currículum:

* Descripción: Información sobre el currículum de los estudiantes, incluyendo la formación académica, la experiencia laboral, las habilidades, etc.
* Campos: id del alumno y nombre del documento.
* Tipo de datos: Texto, numérico.
* Otras observaciones: solo se debe permitir la subida del currículum en formato PDF por parte del estudiante.

### 2.3.2.- Requisitos funcionales.

* Gestión de empresas y sedes: agregar, modificar, eliminar y ver información relevante de empresas y sedes.
* Gestión de alumnos: agregar, modificar, eliminar y ver información relevante de alumnos.
* Gestión de candidaturas: permitirá a docentes gestionar candidaturas de alumnos a empresas.
* Asignación de alumnos a empresas: asignar alumnos a empresas y gestionar el estado de la asignación.
* Gestión de perfiles de alumnos: gestionar información personal.
* Gestión de CV: gestionar y visualizar el formato PDF.
* Visualización de estado de candidaturas: permitirá a alumnos ver el estado de sus candidaturas y asignaciones a empresas.
* Autenticación y autorización: sistema y gestión de permisos y roles.

### 2.3.3.- Otros Requisitos.

* Seguridad: El sistema debe garantizar la seguridad de la información almacenada, así como el acceso controlado a las distintas funcionalidades del sistema.
* Escalabilidad: El sistema debe estar diseñado para poder crecer y adaptarse a nuevas necesidades en el futuro.
* Usabilidad: El sistema debe ser fácil de usar y comprensible para los usuarios, tanto docentes como alumnos, con una interfaz amigable y clara.
* Disponibilidad: El sistema debe estar disponible siempre, con un mínimo de tiempo de inactividad planificado para el mantenimiento y actualizaciones necesarias.

## 2.4.- Descripción de la solución.

La solución propuesta es una aplicación web que permita y facilita la gestión de la FCT de los alumnos de ciclos de un instituto. La aplicación contará con dos perfiles de usuario: el perfil de docente y el perfil de alumno.

La aplicación contará con una base de datos que gestionará todos los aspectos de la gestión de formación en centro de trabajo (datos de empresas, alta de sedes de empresas, gestión de alumnos, asignación de alumnos a empresa, etc.).

La aplicación web se desarrollará con los siguientes lenguajes: HTML, CSS, JavaScript, Laravel y MySQL para la base de datos.

La aplicación contará con medidas de seguridad para garantizar la privacidad y protección de los datos de los usuarios, como la autenticación de usuarios y el uso de conexiones seguras. Además, se realizarán copias de seguridad regulares para prevenir la pérdida de datos en caso de fallos técnicos.

## 2.5.- Planificación del proyecto.

### 2.5.1.- Equipo de trabajo.

Para el proyecto he pensado que se necesitará varios desarrolladores de software, el equipo quedaría así planificado.

* Un desarrollador encargado del diseño y desarrollo del frontend.
* Un desarrollador encargado del diseño y desarrollo del backend.
* Un desarrollador encargado del diseño y desarrollo de la base de datos.
* El jefe, coordinador y administrador del proyecto.

### 2.5.2.- Planificación temporal.

He desarrollado la siguiente planificación, pienso que sería la idónea para realizar el proyecto.

* Semana 1: Investigación y análisis de requisitos.
* Semana 2-3: Diseño y configuración de la base de datos y planificación de la arquitectura del proyecto.
* Semana 4-5: Desarrollo del backend utilizando Laravel.
* Semana 6-7: Desarrollo del frontend utilizando HTML, CSS, Bootstrap y JavaScript.
* Semana 8-9: Integración del frontend con el backend y pruebas de integración.
* Semana 10: Realización de pruebas de aceptación y corrección de errores.
* Semana 11: Terminar documentación, entrega y defensa del proyecto.

## 2.6.- Estudio del coste del proyecto.

He utilizado una hoja de cálculo (Excel) para llevar a cabo este estudio. En la izquierda está la descripción de lo realizado cada semana y a la derecha están las horas aproximadas para la realización de dichas tareas.

He buscado información sobre el salario medio de un programador en España, en esta [página](https://es.talent.com/salary?job=programador#:~:text=Descubre cuál es el salario medio para Programador&text=El salario programador promedio en,hasta € 38.500 al año.), he visto que el salario medio por hora es de 14,62€. Al tener 4 trabajadores, así saldría el proyecto con las siguientes horas.

Abajo del cálculo sale 3 apartados, los voy a explicar:

* Total horas: suma total de las horas de arriba.
* Total horas por €: horas totales multiplicado por 14,62 (euros por hora), lo que cuesta un único trabajador.
* Total horas por € por trabajadores: recuento total entre las horas, dinero por hora y por los 4 trabajadores.



# 3.- Análisis del Sistema de Información.

## 3.1.- Identificación del entorno tecnológico.

El entorno tecnológico para el desarrollo y explotación para el sistema de gestión de FCT incluye los siguientes componentes:

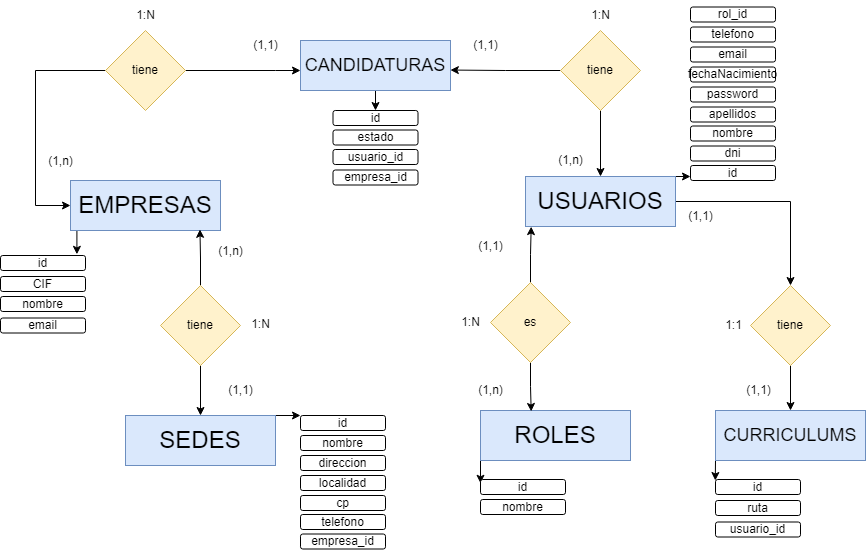
* Lenguajes de programación:
  + HTML, CSS y Bootstrap para el diseño de la interfaz de usuario.
  + JavaScript para agregar funcionalidades interactivas por la parte cliente y API.
  + PHP para el desarrollo de la parte servidor con el framework de Laravel.
  + SQL para la gestión de la base de datos MySQL.
* Frameworks:
  + Laravel: un framework de PHP que facilita el desarrollo de aplicaciones web robustas y escalables, además de contar con una gran cantidad de funcionalidades, como la gestión de autenticación, manejo de bases de datos, entre otros.
  + Bootstrap: un framework de diseño web que ayuda a construir interfaces modernas y responsivas, de manera sencilla y eficiente.
* Herramientas de desarrollo:
  + IDE: se utiliza Visual Studio Code.
  + Servidor web: se utiliza Apache.
  + Sistema Gestor de Bases de Datos: se utiliza MySQL.
  + Control de versiones: Git.
* Entorno de explotación:
  + Se necesita un servidor web y una base de datos para alojar la aplicación.
  + El sistema debe ser compatible con los navegadores web modernos y debe ser capaz de manejar múltiples solicitudes simultáneamente.
* Requisitos y versiones:
  + XAMPP Control Panel v3.3.0.
  + Apache v2.4.53.
  + PHP v8.1.6.
  + phpMyAdmin v5.2.0.
  + Composer v2.5.7.
  + Laravel v9.
  + Bootstrap v5.

En resumen, el entorno tecnológico del sistema de gestión de la FCT es bastante amplio e incluye una variedad de herramientas y tecnologías para construir una aplicación web robusta y escalable.

## 3.2.- Modelado de datos.

### 3.2.1.- Modelo Entidad-Relación.

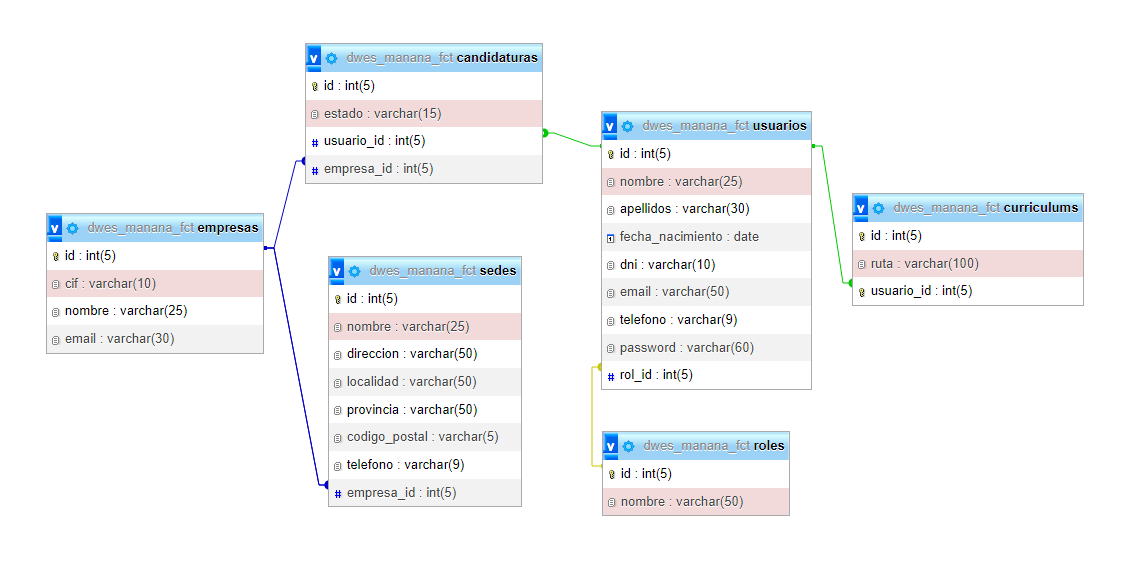
He realizado un diagrama E/R usando la aplicación web de drawio. Nombre del archivo: ER\_FCT.drawio.pdf.



* La tabla **Usuarios** tiene una relación uno-a-muchos con la tabla Candidaturas, ya que un usuario (rol alumno) puede tener varias candidaturas y una candidatura es de un usuario. Tiene una relación uno-a-muchos con la tabla Roles, un usuario puede tener un solo rol, pero los roles tienen muchos usuarios. Y, por último, una relación uno-a-uno con la tabla Curriculums, un usuario (rol\_alumno) tiene un solo CV y un CV solo pertenece a un usuario.
* La tabla **Empresas** se relaciona con la tabla Candidaturas mediante el campo empresa\_id, relación uno-a-muchos ya que una candidatura solo tiene una empresa y una empresa tiene varias candidaturas. Una relación uno-a-muchos la tabla Sedes ya que una empresa puede tener uno o varias sedes, pero una sede pertenece a una empresa.
* La tabla **Roles** permite gestionar los diferentes roles de usuarios que pueden acceder al sistema (alumno y profesor). Un usuario solo tiene un rol y un rol puede tener muchos usuarios.
* La tabla **Sedes** tiene una relación uno-a-muchos con Empresas. Una sede tiene una empresa, pero una empresa puede tener una o muchas sedes.
* La tabla **Candidaturas** tiene una relación uno-a-uno con Empresas y Alumnos. Una candidatura tiene una empresa y un alumno.
* La tabla **Curriculums** tiene una relación uno-a-uno con la tabla Usuarios, un usuario (rol\_alumno) tiene un solo CV y un CV solo pertenece a un usuario.

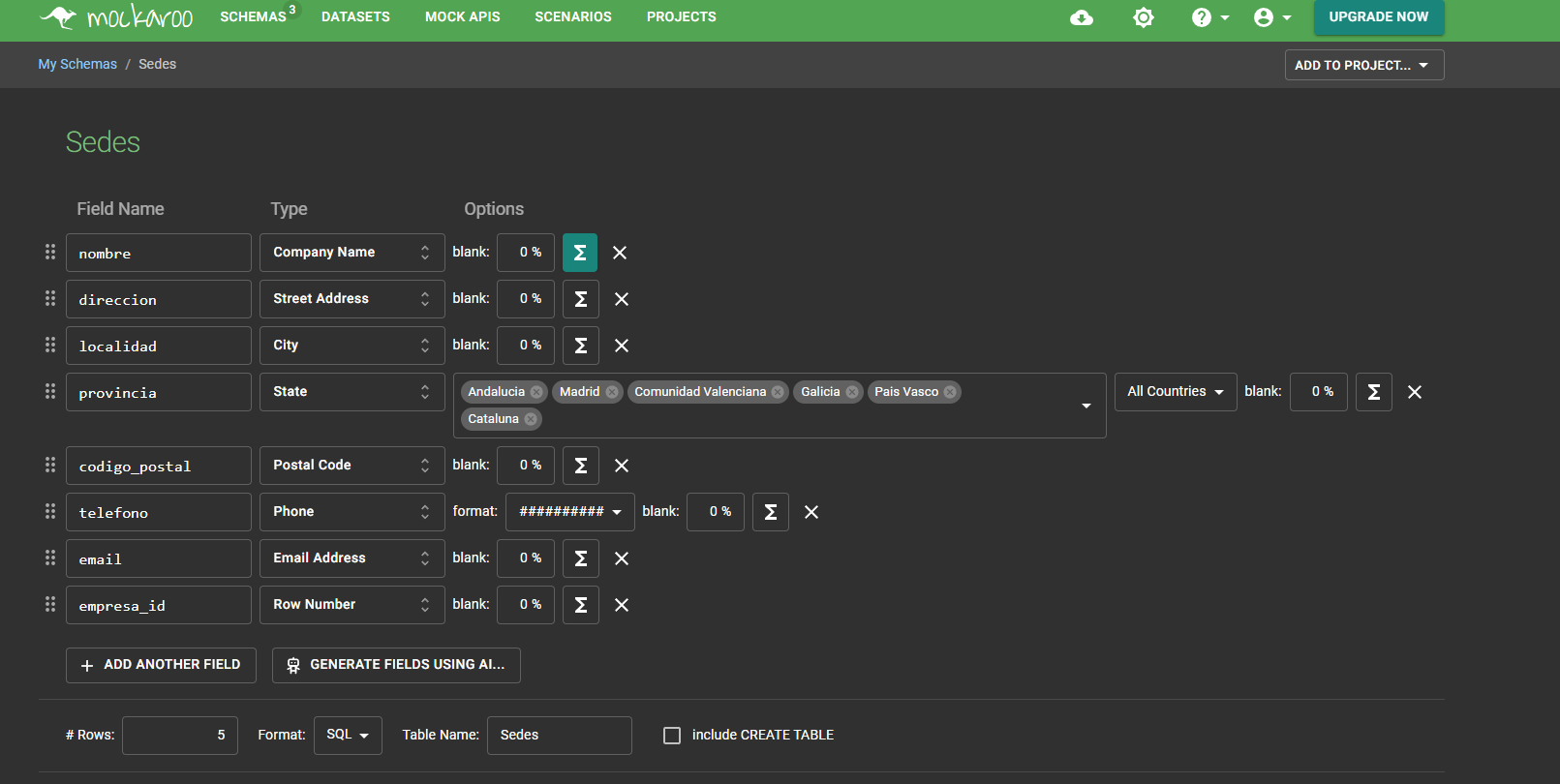
### 3.2.2.- Esquema de la base de datos.

Se adjunta el archivo SQL con el esquema de la base de datos. Nombre del archivo: fct\_create.sql. En él se puede ver el tipo de cada atributo de todas las tablas, también adjunto una imagen donde se puede ver la relación y los mismos tipos.



### 3.2.3.- Datos de prueba.

He usado la página de Mockaroo (<https://www.mockaroo.com/>) para generar datos. Nombre del archivo: fct\_insert.sql.



## 3.3.- Identificación de los usuarios participantes y finales.

Tendremos dos tipos de usuarios: Profesores y alumnos.

* **Profesor**. Podrá gestionar las empresas y sus distintas sedes, gestión del alumnado y asignar a los distintos alumnos dados de alta a las distintas empresas, teniendo en cuenta que un alumno puede estar en más de una empresa y que durante todo el proceso de gestión la candidatura podrá ser aceptada o rechazada por la empresa.
* **Alumno**. Podrá gestionar su perfil, subir su CV sólo en formato PDF y verá el estado de sus candidaturas, asignación a las distintas empresas.

La separación de los usuarios en dos tipos diferentes es fundamental para la gestión y asignación de los alumnos a las empresas, así como para la visualización y gestión de los perfiles personales.

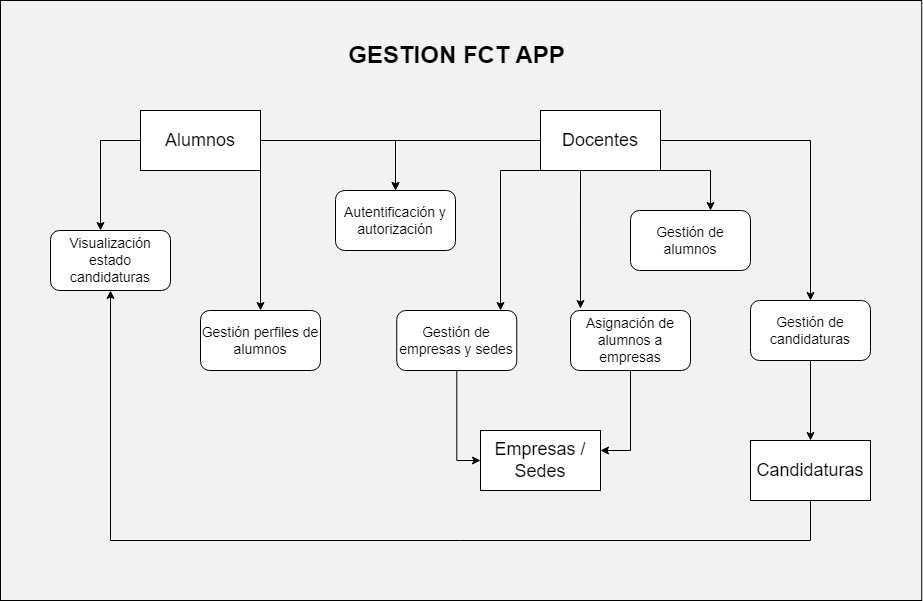
La plataforma permitirá una gestión más eficiente y organizada de la FCT, lo que proporcionará una experiencia más fluida y eficaz tanto para los estudiantes como para los profesores.

## 3.4.- Identificación de subsistemas de análisis.

Así he dividido el sistema en subsistemas.

* Usuarios.
  + Subsistema de autenticación y autorización: permitirá la autenticación al sistema.
* Docente.
  + Gestión de empresas y sedes: agregar, modificar y eliminar empresas y sedes.
  + Subsistema de gestión de alumnos: agregar, modificar y eliminar alumnos.
  + Subsistema de gestión de candidaturas: visualizar y gestionar las candidaturas.
  + Subsistema de asignación de alumnos a empresas: asignar a los alumnos a las distintas empresas.
  + Subsistema de autenticación y autorización: permitirá la autenticación al sistema.
* Alumnos.
  + Subsistema de gestión de perfiles de alumnos: gestionar su perfil.
  + Subsistema de gestión de CV: gestionar y visualizar su CV.
  + Subsistema de visualización de estado de candidaturas: visualizar el estado de sus candidaturas.
* Empresas.
* Candidaturas.

Así quedaría el diagrama de subsistemas:



## 3.5.- Establecimiento de requisitos.

* Subsistema de gestión de empresas y sedes: permitirá a los docentes agregar, modificar y eliminar empresas y sus sedes, así como visualizar su información relevante.
* Subsistema de gestión de alumnos: permitirá a los docentes agregar, modificar y eliminar alumnos, así como visualizar su información relevante.
* Subsistema de gestión de candidaturas: permitirá a los docentes visualizar y gestionar las candidaturas de los alumnos a las distintas empresas.
* Subsistema de asignación de alumnos a empresas: permitirá a los docentes asignar a los alumnos a las distintas empresas y gestionar el estado de la asignación.
* Subsistema de gestión de perfiles de alumnos: permitirá a los alumnos gestionar su perfil, incluyendo su información personal (nombre, apellidos, email, …), preferencias y subir o cambiar su CV.
* Subsistema de visualización de estado de candidaturas: permitirá a los alumnos visualizar el estado de sus candidaturas y la asignación a las distintas empresas.
* Subsistema de autenticación y autorización: permitirá la autenticación de los usuarios (docentes y alumnos) en el sistema y la gestión de permisos y roles.

## 3.6.- Diagramas de Análisis.

Diagrama de caso de uso de mi sistema. Con gestión me refiero a añadir, modificar, eliminar o visualizar a los alumnos, empresas, sedes y candidaturas.

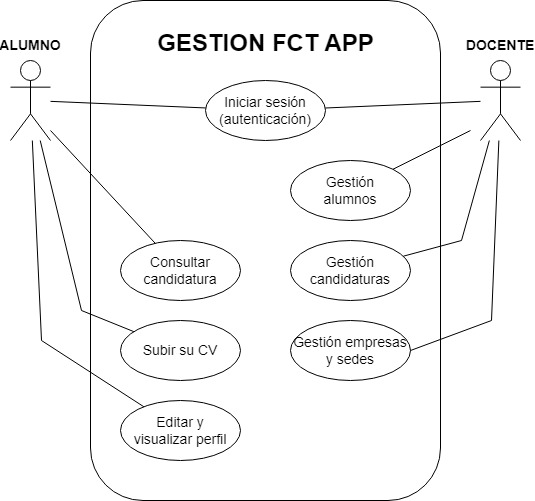
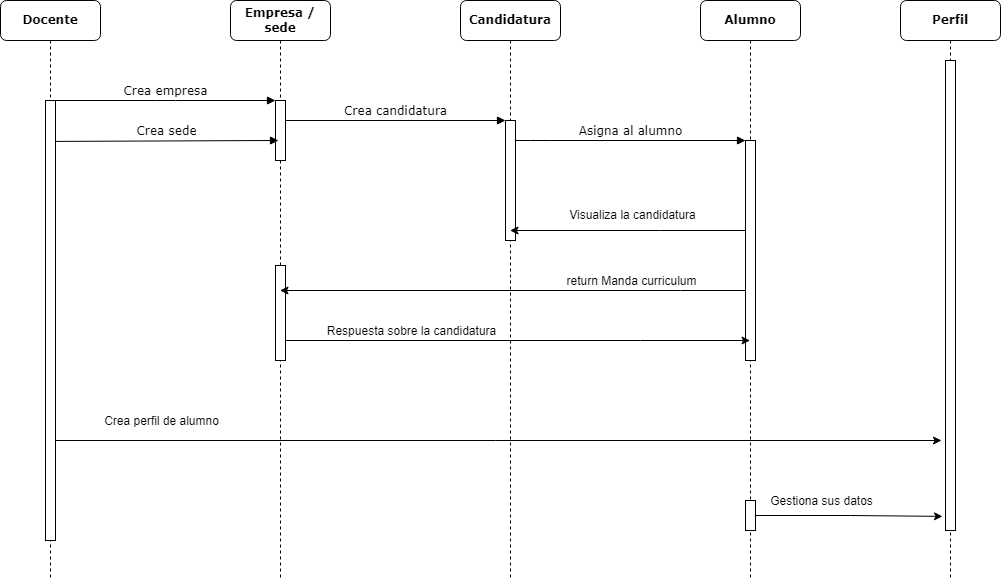


Diagrama de secuencias de mi sistema, donde el profesor es el encargado de crear una empresa y hacer la gestión de la candidatura con el alumno, después el alumno tiene que subir su CV y la empresa le daría una respuesta.



## 3.7.- Definición de interfaces de usuario.

### 3.7.1.- Especificación de principios generales de interfaz.

* Claridad y simplicidad: La interfaz es clara y fácil de entender para cualquier usuario. He priorizado la simplicidad en la presentación de la información.
* Consistencia: Los elementos comunes tienen una apariencia y funcionamiento coherentes en todas las partes del sistema.
* Retroalimentación adecuada: Mensajes de confirmación, notificaciones de error y actualizaciones de estado.
* Facilidad de uso: La interfaz es intuitiva y fácil de usar.
* Minimizar la carga cognitiva: Se ha tenido en cuenta la carga cognitiva del usuario, evitando la sobrecarga de información. He buscado la simplicidad y la facilidad de comprensión.
* Accesibilidad: La interfaz es accesible para usuarios con diferentes habilidades y necesidades. Esto implica considerar el contraste de colores, el tamaño de fuente y otras características que mejoran la experiencia para personas con discapacidades.
* Diseño responsivo: La interfaz se adapta a diferentes dispositivos y tamaños de pantalla. El diseño responsivo garantiza que la interfaz se vea y funcione correctamente en dispositivos móviles, tablets y pantallas de escritorio.

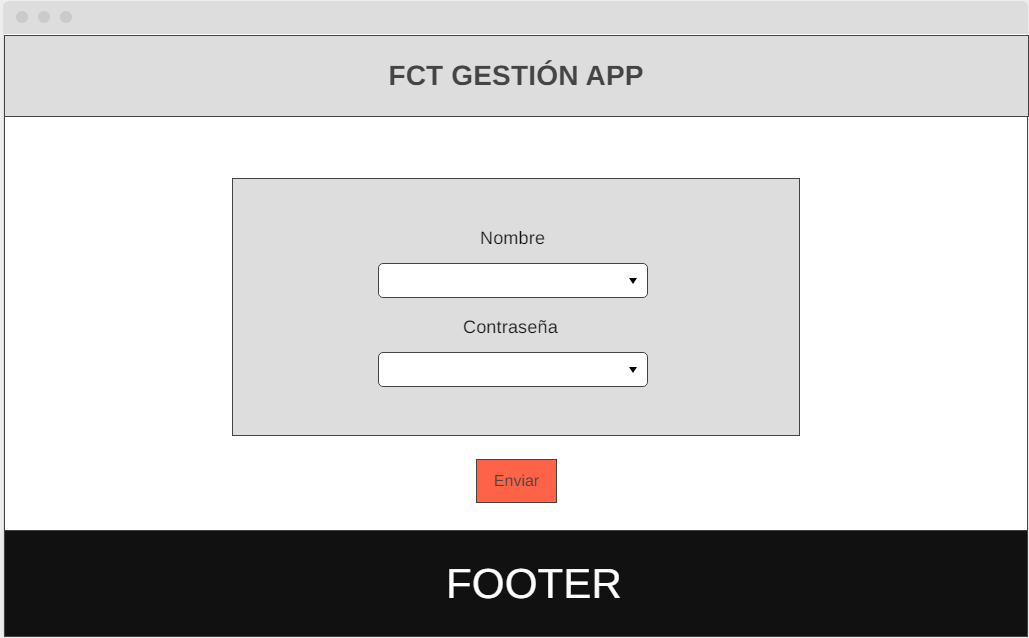
### 3.7.2.- Especificación de formatos individuales de la interfaz de pantalla.

Para la realización de los diseños de cada interfaz de pantalla he usado la aplicación web de Wireframe.cc.

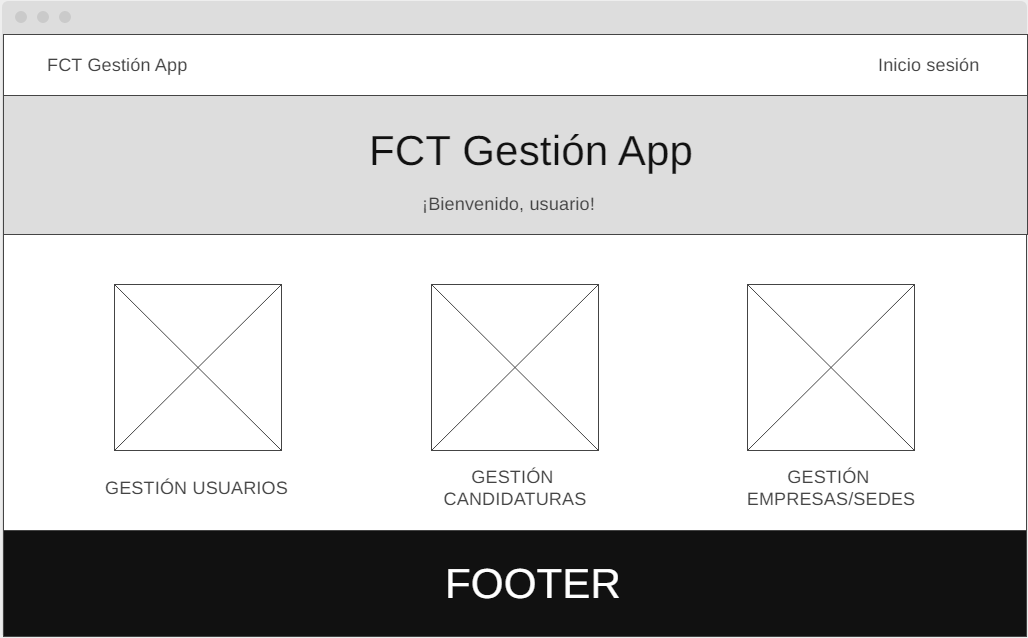
* Pantalla principal.
  + Nombre del archivo: index.
  + Se adapta a cualquier pantalla.



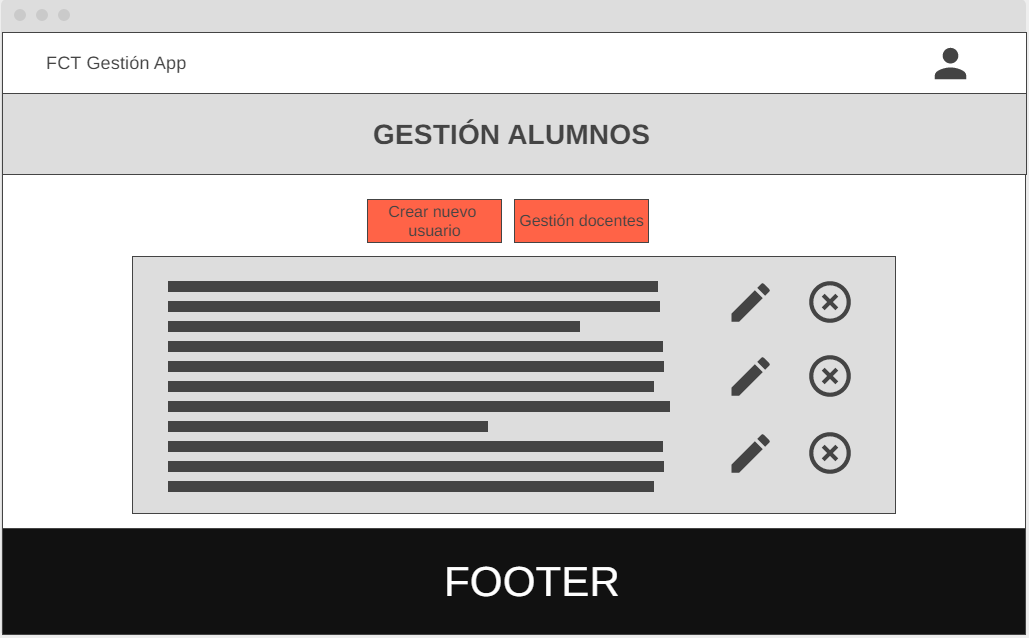
* Log in.
  + Nombre del archivo: login.
  + Se adapta a cualquier pantalla.



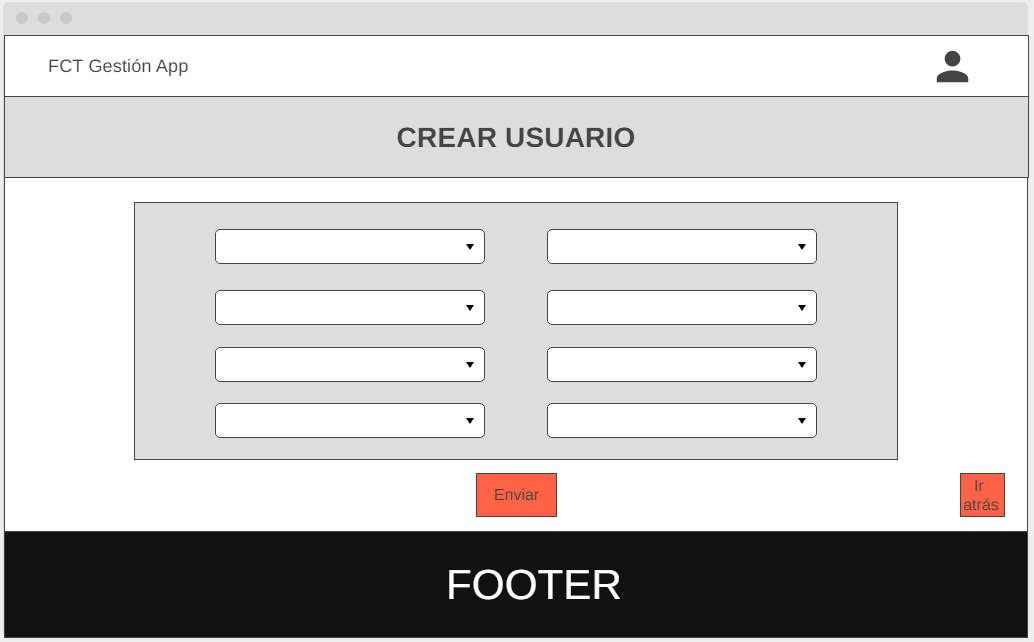
* Docente.
  + Inicio profesor.
    - Nombre del archivo: inicio\_profesor.
    - Se adapta a cualquier pantalla.



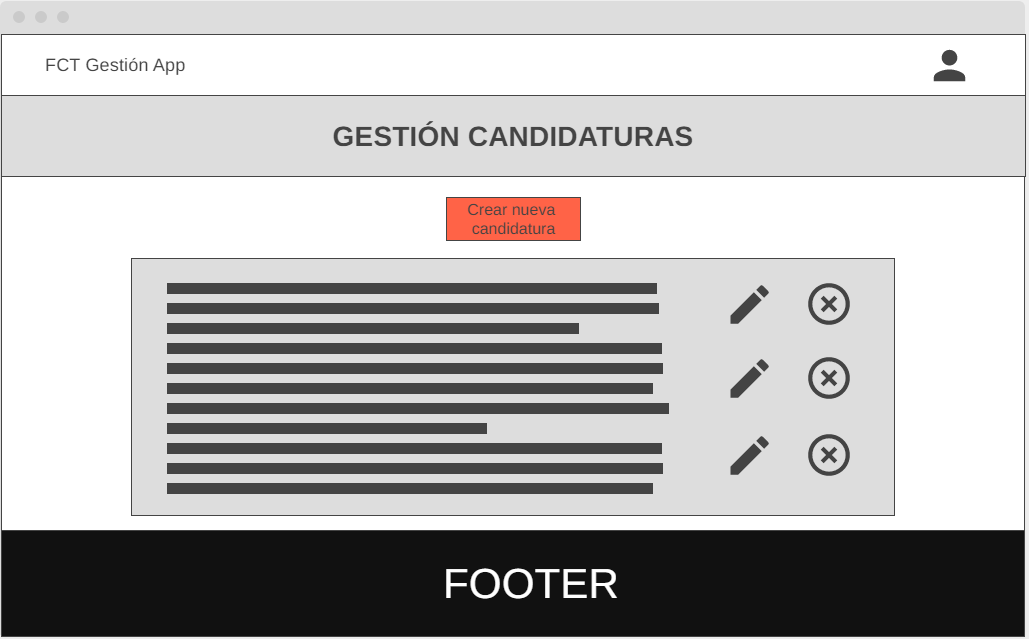
* + Gestión alumnos.
    - Nombre del archivo: gestion\_alumnos.
    - Se adapta a cualquier pantalla.



* + Gestión usuarios formulario.
    - Nombre del archivo: gestion\_alumnos\_form.
    - Se adapta a cualquier pantalla.



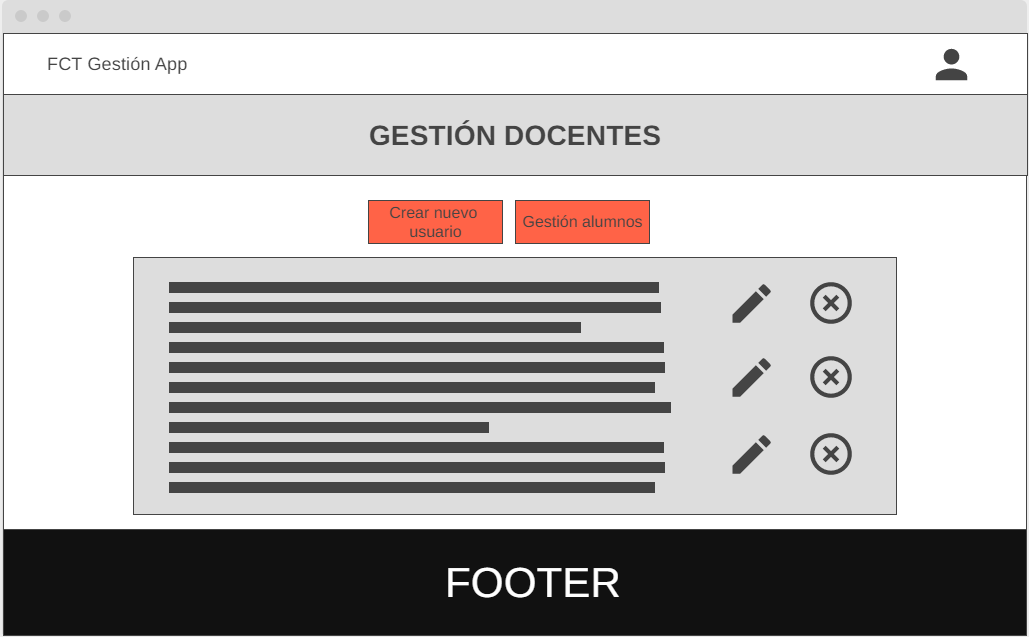
* + Gestión candidaturas.
    - Nombre del archivo: gestion\_candidaturas.
    - Se adapta a cualquier pantalla.



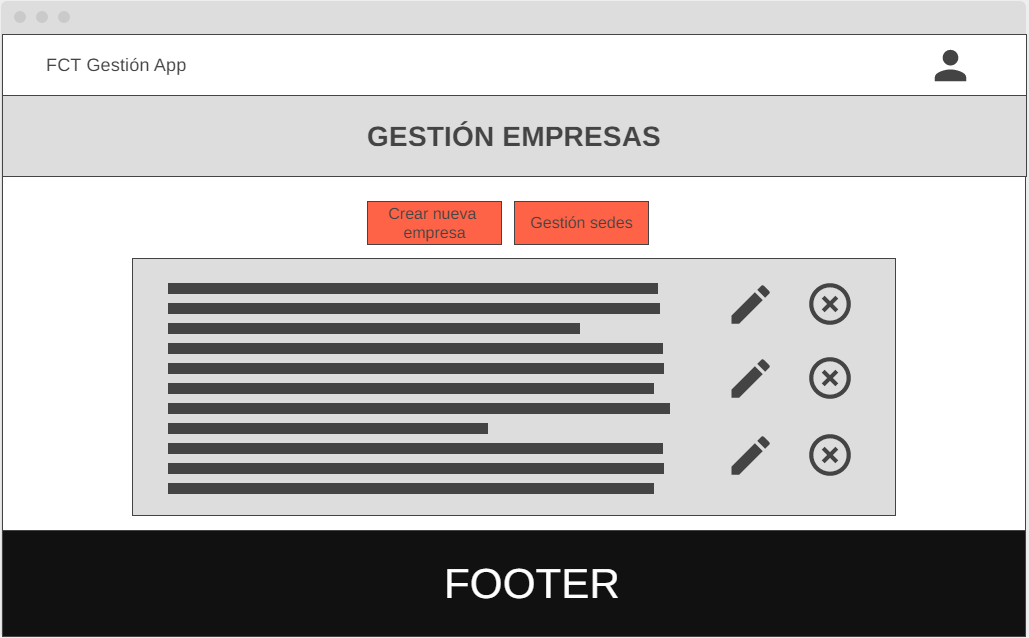
* + Gestión candidaturas formulario.
    - Nombre del archivo: gestion\_candidaturas\_form.
    - Se adapta a cualquier pantalla.



* + Gestión docentes.
    - Nombre del archivo: gestion\_docentes
    - Se adapta a cualquier pantalla.



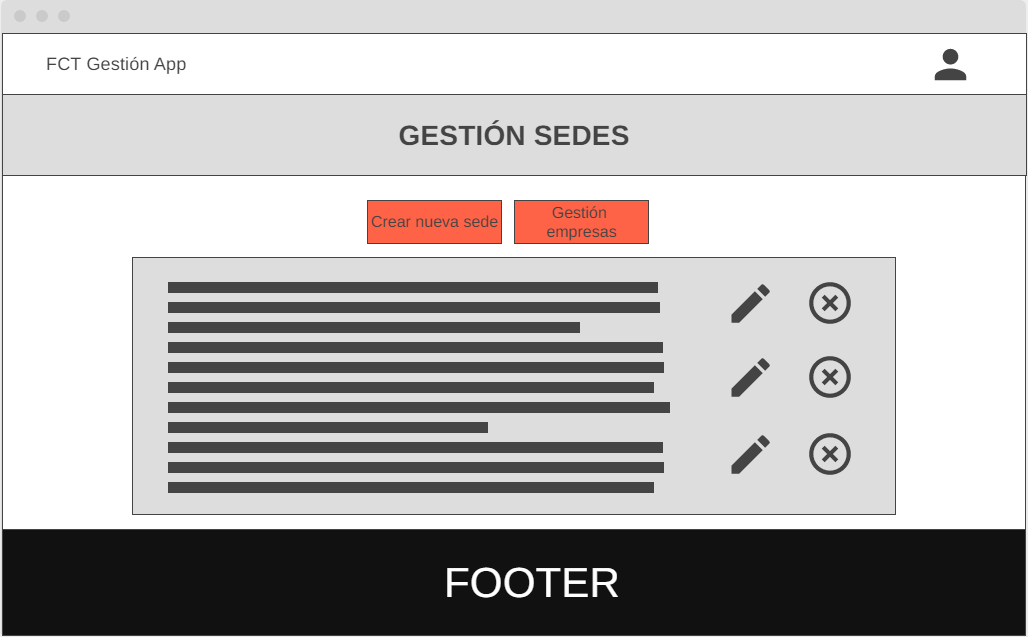
* + Gestión empresas.
    - Nombre del archivo: gestion\_empresas.
    - Se adapta a cualquier pantalla.



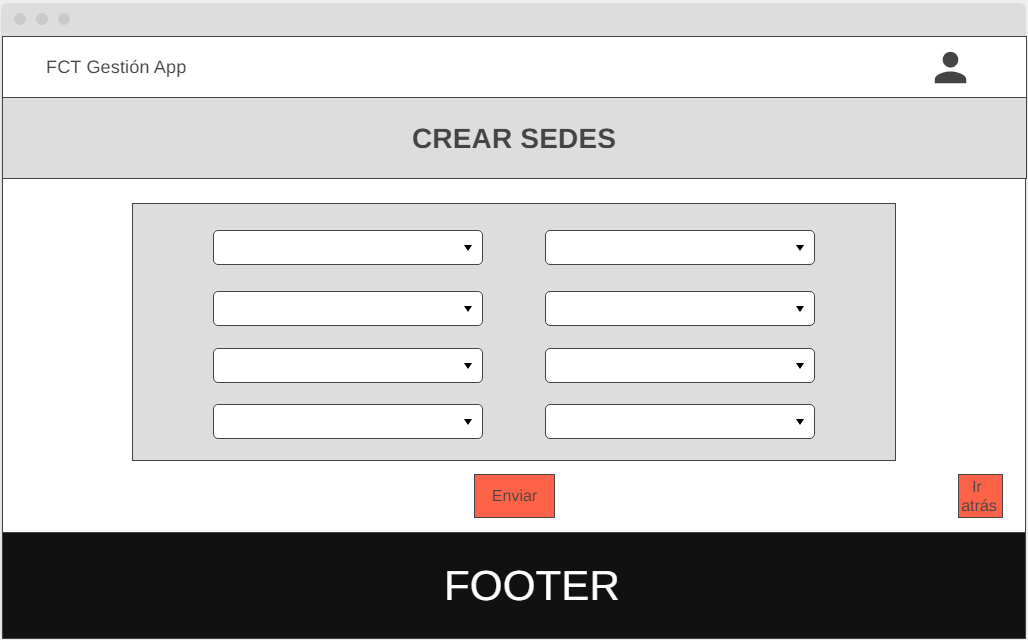
* + Gestión empresas formulario.
    - Nombre del archivo: gestion\_empresas\_form.
    - Se adapta a cualquier pantalla.



* + Gestión sedes.
    - Nombre del archivo: gestion\_sedes.
    - Se adapta a cualquier pantalla.



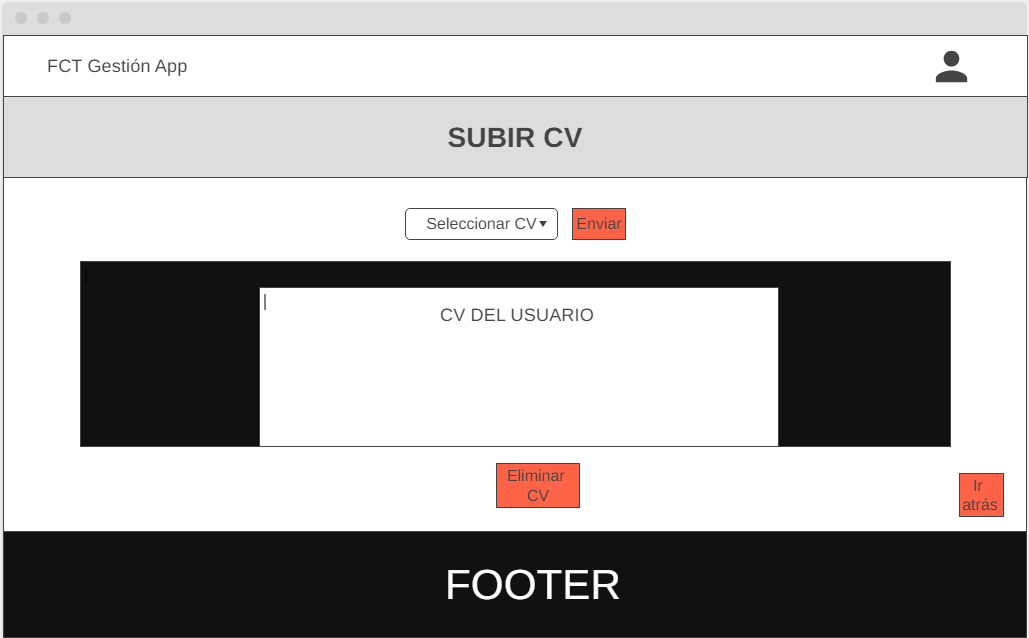
* + Gestión sedes formulario.
    - Nombre del archivo: gestion\_sedes\_form.
    - Se adapta a cualquier pantalla.



* Alumno.
  + Inicio alumno.
    - Nombre del archivo: inicio\_alumno.
    - Se adapta a cualquier pantalla.



* + Gestión CV.
    - Nombre del archivo: gestion\_cv.
    - Se adapta a cualquier pantalla.



* Perfil.
  + Nombre del archivo: perfil\_usuario.
  + Se adapta a cualquier pantalla.



* Acceso denegado.
  + Nombre del archivo: acceso\_denegado.
  + Se adapta a cualquier pantalla.



### 3.7.3.- Identificación de perfiles de usuario.

Como ya he comentado anteriormente, hay dos perfiles de usuarios, el perfil profesor y el perfil alumno. Cada usuario tendrá páginas donde no puede acceder el otro usuario, pero también habrá varias páginas donde serán compartidas en ambos perfiles.

Compartidas:

* Página de inicio (no necesita iniciar sesión).
* Página de log in (no necesita iniciar sesión).
* Página de perfil de usuario.
  + Cambia detalles con JavaScript
* Página de acceso denegado.

Profesor:

* Página de inicio de profesor.
* Página gestión de alumnos.
* Página gestión de docentes.
* Página gestión de empresas.
* Página gestión de sedes.
* Página gestión de candidaturas.
* Página crear usuarios.
* Página crear empresas.
* Página crear sedes.
* Página crear candidaturas.

Alumno:

* Página de inicio de alumno.
* Página de gestión del CV del alumno.

### 3.7.4.- Especificación de formatos de impresión.

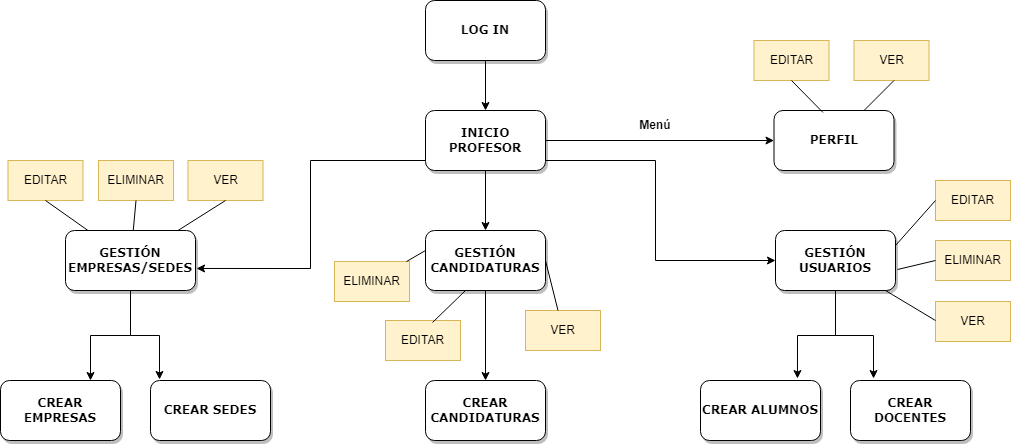
Como tal, mi programa no genera ningún documento, tiene la capacidad de imprimir los PDF existentes asociados a los usuarios almacenados en la base de datos.

El proceso de impresión se realiza a través de la gestión del CV del alumno. Cuando un alumno accede a esta página, el programa recupera la ruta del archivo PDF asociado al usuario desde la base de datos. Si existe un PDF para el usuario en cuestión, se realiza la impresión gracias a una etiqueta embed que carga el PDF en el navegador.

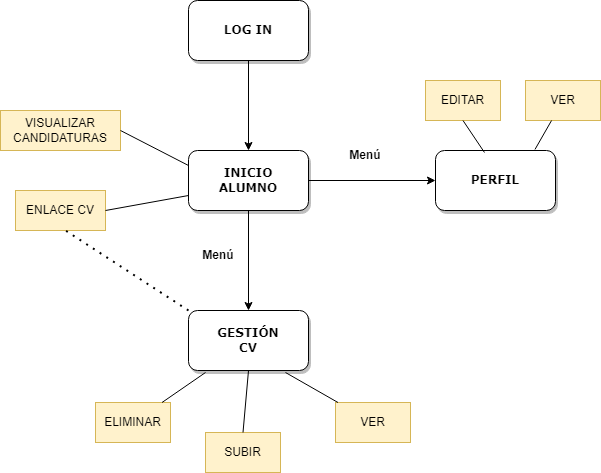
### 3.7.5.- Especificación de la navegabilidad entre pantallas.

Cada perfil se podrá mover por la aplicación de una forma diferente, he diseñado un diagrama donde se ve perfectamente por donde puede navegar cada usuario.

* Navegabilidad del profesor.
  + En todas las páginas hay un botón para volver a la página anterior según la navegabilidad así reflejada en el diagrama.



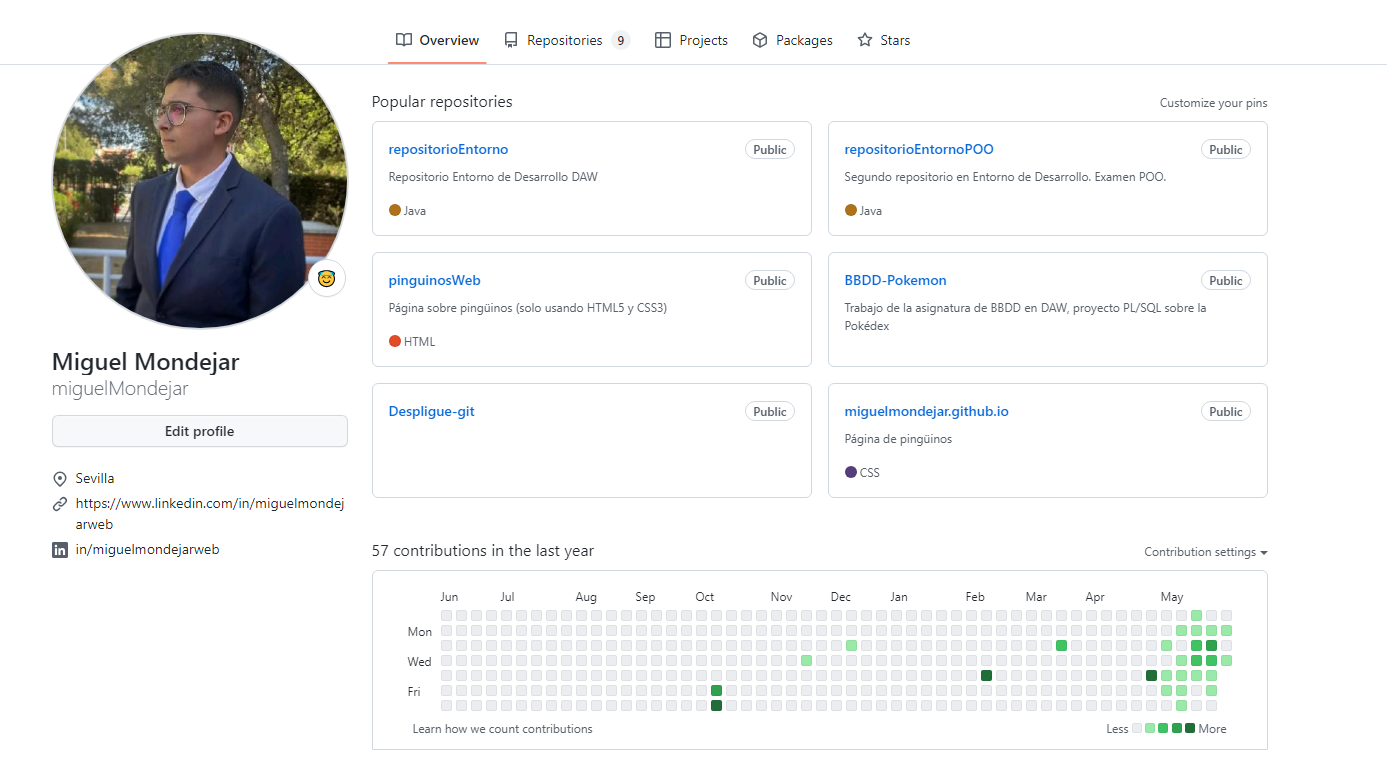
* Navegabilidad del alumno.
  + En todas las páginas hay un botón para volver a la página anterior según la navegabilidad así reflejada en el diagrama.



# 4.- Construcción del Sistema.

El uso de un sistema de control de versiones como Git y la plataforma GitHub contribuyen a una gestión más eficiente y ordenada del código fuente, promoviendo las buenas prácticas de desarrollo y facilitando la colaboración en equipo.

En mi GitHub personal he subido el código, en él se puede ver también todos los commits que he realizado con la herramienta de Git.



<https://github.com/miguelMondejar/Proyecto_Integrado>

Para hacer cada commit he usado los siguientes comandos en una terminal bash de Git:

* git add .
* git commit -m “version 1…”
* git push

Si tenía cambios no subidos a mi directorio:

* git pull

# 5.- Conclusiones.

A lo largo del proyecto me he encontrado problemas de todo tipo, problemas que me ha costado mucho resolver, problemas que con una simple búsqueda en Google he podido solucionarlo, problemas de error de sintaxis o incluso incurrencias mías. Dejo una lista de los errores más destacados que he tenido.

* Error al inicio de sesión. Estuve varios días intentando resolver un problema del inicio de sesión con JWTAuth, resulta que las contraseñas en base de datos deben estar encriptadas porque si no da error.
* Nombre de las tablas de Usuarios y Curriculums me da error en los modelos de Laravel. Por defecto coge la tabla Users y el modelo Curriculum al pasarlo al plural me lo ponía con un nombre raro.
* Error llamadas POST a la API con JavaScript.
  + Al pasar el valor de los campos no lo ponía entre comillas y provocaba fallo.
  + No ponía todos los headers necesarios.
* No borraba correctamente usuarios o empresas.
  + No borraba ya que tenía candidaturas asociadas, esto lo solucioné configurando la base de datos con ON DELETE CASCADE.
* Botón editar perfil.
  + Fallaba al dar al botón, era por el nombre del id de un elemento.
* Poner email y contraseña dos veces para que haga el log in.
  + Error solucionado al desplegar en con apache.

Los errores te hacen más fuertes y estos me han servido como experiencia ya que me han permitido aprender y seguir desarrollándome como desarrollador web.

# 6.- Glosario de términos.

* **Interfaz de usuario:** Es el medio a través del cual los usuarios interactúan con un sistema o una aplicación. Puede incluir elementos visuales, como botones y formularios, así como interacciones táctiles o de voz.
* **Backend**: Es la parte de la aplicación que se encarga del procesamiento y gestión de datos. Incluye la lógica de negocio, la gestión de bases de datos y la comunicación con otros sistemas.
* **Frontend**: Es la parte de la aplicación que se encarga de la presentación de la interfaz de usuario. Incluye la estructura, diseño y funcionalidad visual que los usuarios pueden ver y con la que pueden interactuar.
* **Base de datos**: Es un conjunto estructurado de datos que se organiza y almacena de manera sistemática. Permite la gestión eficiente y segura de la información utilizada por el sistema.
* **Framework**: Es un conjunto de herramientas, bibliotecas y pautas que proporcionan una estructura y funcionalidad predefinidas para el desarrollo de aplicaciones web.
* **API** (Interfaz de Programación de Aplicaciones): Es un conjunto de reglas y protocolos que permite la comunicación y la interacción entre diferentes componentes de software. Proporciona una forma estandarizada de acceder a funcionalidades o datos de un sistema.
* **Log in**: El término "Log in" se refiere al proceso de acceder a un sistema o aplicación mediante la introducción de credenciales válidas, generalmente un nombre de usuario y una contraseña.
* **IDE**: Entorno de Desarrollo Integrado, es una herramienta de software que proporciona un entorno de programación completo para los desarrolladores de software.
* **Git**: Sistema de control de versiones distribuido, proporciona un conjunto de comandos y herramientas que permiten a los desarrolladores realizar un seguimiento de los cambios en su código.
* **Commit**: En Git es una acción que registra y guarda los cambios realizados en un repositorio.

# 7.- Bibliografía.

Documentación sobre Laravel: <https://laravel.com/docs/10.x>

Ayuda para JavaScript: <https://javascript.info/>

Plantilla Bootstrap: <https://startbootstrap.com/>

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

Mockaroo: <https://www.mockaroo.com/>

Wireframe <https://wireframe.cc/>

Ayuda crear diagrama de casos de uso <https://support.microsoft.com/es-es/office/crear-un-diagrama-de-casos-de-uso-uml-92cc948d-fc74-466c-9457-e82d62ee1298>

Ayuda crear diagrama de secuencias <https://es.venngage.com/blog/diagrama-de-actividades/>

Salario medio desarrollador [https://es.talent.com/salary?job=programador#:~:text=Descubre%20cu%C3%A1l%20es%20el%20salario%20medio%20para%20Programador&text=El%20salario%20programador%20promedio%20en,hasta%20%E2%82%AC%2038.500%20al%20a%C3%B1o](https://es.talent.com/salary?job=programador#:~:text=Descubre cuál es el salario medio para Programador&text=El salario programador promedio en,hasta € 38.500 al año)

Ayuda para Git <https://rogerdudler.github.io/git-guide/index.es.html>

# Apéndices.

Diagramas.

* Diagrama Entidad-Relación.
* Diagrama de Subsistemas.
* Diagrama de caso de uso.
* Diagrama de secuencias.
* Diagrama navegabilidad docente.
* Diagrama navegabilidad profesor.

Diseño de las interfaces.

* Interfaz inicio.
* Interfaz log in.
* Interfaz acceso denegado.
* Interfaz inicio profesor.
* Interfaz inicio alumno.
* Interfaz perfil usuario.
* Interfaz gestión usuarios.
* Interfaz gestión docentes.
* Interfaz crear usuarios.
* Interfaz gestión candidaturas.
* Interfaz crear candidaturas.
* Interfaz gestión empresas.
* Interfaz gestión sedes.
* Interfaz crear empresas.
* Interfaz crear sedes.
* Interfaz gestión CV.

Base de datos.

* SQL create table.
* SQL insert.

Documentos.

* Estudio coste de proyecto.