



CENTRE D'ESTUDIS AULA CAMPUS
CICLO FORMATIVO DE GRADO SUPERIOR
DESARROLLO DE APLICACIONES WEB
Proyecto de Fin de Ciclo

Asistente para el PFC

Roberto Martínez Rodríguez

Tutor: Mari Cardells Cifre

Curso: CURSO 2022/2023

Índice

Introducción.....	3
Resumen.....	3
Justificación del proyecto.....	4
Objetivos.....	4
Desarrollo del proyecto.....	4
Tecnologías escogidas y justificación.....	4
Fundamentación teórica.....	4
Tecnologías escogidas.....	4
Diseño de la aplicación.....	5
Casos de uso.....	5
Acciones realizadas.....	7
Puesta en marcha del entorno de trabajo.....	7
Mockup de la aplicación.....	7
Implementación de las funcionalidades.....	9
Resultados y Análisis.....	14
Arquitectura de la aplicación.....	14
Estructura del proyecto.....	14
Manual de despliegue.....	15
Instrucciones de despliegue.....	15
Temporalización.....	17
Conclusiones.....	17
Líneas de investigación futuras.....	17
Bibliografía.....	18

Introducción

Este proyecto consiste en el desarrollo de una aplicación web destinada a la asistencia para la gestión de tareas y tiempo del Proyecto de Fin de Ciclo (PFC).

La aplicación se desarrolla en web, de forma que los usuarios podrán acceder sin necesidad de registro previo, solamente con conexión a Internet.

Resumen

Esta aplicación consta de dos partes, un frontend en donde se encuentra toda la lógica para que el usuario tenga una experiencia sencilla, y el backend, donde se guardarían todos los datos relacionados con el proyecto.

En la parte del frontend, se encuentran todos los archivos .vue y javascript donde se han creado todas las configuraciones de todas las funciones y todos los estilos visuales junto con un store para ir guardando todos los datos a medida que se van creando las funcionalidades para ver si su funcionamiento es correcto antes de implementar los métodos para conectar con la base de datos.

En la parte de backend, se encuentran todos los archivos php para el control y comunicación de las dos partes para la gestión de los datos. En esta parte se muestran los datos guardados cuando se hace alguna acción, como la de crear, editar o borrar, para que los guarde y muestre en pantalla.

Por la parte visual, ha sufrido bastantes cambios por las implementaciones a lo largo del proceso de creación de los distintos lenguajes como vue o bootstrap ya que cada uno tienen códigos distintos a la hora de crear los diferentes archivos y comunicarlos entre si.

Y por último, la aplicación tiene un store para comprobar que VUE está en funcionamiento y te permite crear, editar, borrar y guardar los datos creados en el frontend; y una base de datos donde se guardarán los datos y se mostrarán desde el backend en el frontend de la aplicación.

This application consists of two parts, a frontend where all the logic is located so that the user has a simple experience, and the backend, where all the data related to the project would be stored.

In the frontend part, there are all the .vue and javascript files where all the configurations of all the functions and all the visual styles have been created together with a store to save all the data as the functionalities are created for see if it works correctly before implementing the methods to connect to the database.

In the backend part, there are all the php files for the control and communication of the two parts for data management. This part shows the data saved when an action is done, such as creating, editing or deleting, so that it can be saved and displayed on the screen.

On the visual side, it has undergone many changes due to the implementations throughout the creation process of the different languages 1such as vue or bootstrap since each one has different codes when creating the different files and communicating them with each other.

And finally, the application has a store to check that VUE is working and allows you to create, edit, delete and save the data created in the frontend; and a database where the data will be saved and displayed from the backend in the frontend of the application.

Justificación del proyecto

Es un proyecto en el que se puede completar el PFC y entender el procedimiento de trabajo, estudiar elementos nuevos y entenderlos. Se demuestra la utilización de todos los conocimientos adquiridos durante el curso y estudiar nuevos para mejorar y ampliar los conocimientos dados durante el curso.

Objetivos

El objetivo principal de este proyecto es crear una aplicación/asistente para la gestión de la memoria del proyecto con salida en formato editable, que pueda ser exportado a una suite informática como Microsoft Office o LibreOffice.

El objetivo principal se desglosa en los siguientes objetivos secundarios:

- Gestión del documento
- Gestión de tiempo y tareas del trabajo personal
- Generador de documento
- Investigar y conocer otras herramientas similares
- Conocer y trabajar con las nuevas tecnologías implementadas como Vue y mejorar las de Bootstrap

Desarrollo del proyecto

Tecnologías escogidas y justificación

Fundamentación teórica

Mirando distintas posibilidades, no hay ninguna propuesta similar al proyecto pero sí ejemplos de distintas aplicaciones parecidas en funciones pero mucho más avanzadas como Jira, TimeCap, etc.

Tecnologías escogidas

Se utilizará un entorno de trabajo utilizando docker y que implementa el servidor de aplicaciones, la aplicación web en sí desarrollada en PHP, y una base de datos en D-Beaver. El desarrollo contempla el control de código en un repositorio que estaría ubicado en GitHub.

La aplicación se desarrolla utilizando el framework de trabajo Symfony en su versión 6. También cabe destacar el uso de Bootstrap y Vue para los diseños visuales de las páginas web.

Se utiliza el framework de trabajo de Symfony pues:

- Utiliza un concepto de Modelo-Vista-Controlador¹.
- Te simplifica algunos de los procesos para la conexión con la base de datos.
- Incorpora la gestión de las vistas utilizando un sistema de plantillas: twig.

Se utilizará Visual Studio Code como IDE para el desarrollo del proyecto ya que puede utilizarse en el desarrollo de los lenguaje de programación que vamos a utilizar, y dispone de diferentes herramientas que nos ayudarán a la hora de definir los distintos componentes del proyecto y la opción de visualización en navegador de lo que hayamos programado.

Diseño de la aplicación

Vamos a realizar el diseño de la aplicación definiendo los casos de uso que se van a implementar. Se considera un *caso de uso* como una secuencia de acciones que da lugar al resultado buscado. Expresaran los requisitos funcionales en el contexto de nuestro sistema.

Casos de uso

Página Home	No existe documento asociado al proyecto	
Descripción	Se informa al usuario que no hay documento y se le da la opción de crearlo	
Acciones	Pasos	Acción
	1	El usuario puede acceder a la zona de crear el proyecto

Página Home	Existe documento asociado al proyecto	
Descripción	Se presenta al usuario el documento	
comentarios	Se formatea visualmente en un estilo definido por defecto	

Página PFC	Añadir sección	
Descripción	El usuario puede añadir secciones al proyecto	
Acciones	Pasos	Acción
	1	El usuario hace click en el botón para añadir
	2	Rellena la información correspondiente al título de la sección
	3	El usuario guarda las modificaciones
	4	El usuario puede deshacer los cambios
Condiciones	1- El proyecto tiene que existir	
Comentarios	Solo se permite texto	

Si se quiere crear una sección con contenido (sin subsecciones), en el sistema hay que crear una subsección sin título con el contenido que se quiere meter en la sección

¹ Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Ver **Referencia MVC**

Página PFC	Borrar sección	
Descripción	El usuario puede borrar una sección del proyecto	
Acciones	Pasos	Acción
	1	El usuario hace click en el botón para borrar
	2	El sistema le presenta un ventana de confirmación
	3	El usuario acepta y se borra
	4	Si el usuario no acepta no se confirma el borrado
Condiciones	1- El proyecto tiene que existir 2- La sección tiene que existir	
Comentarios	Se borra todo el contenido de la sección, incluida subsecciones	

Página Tareas	Filtrar por categoría	
Descripción	El usuario podrá filtrar la tarea por las distintas partes de la lista	

Página Tareas	Búsqueda por nombre	
Descripción	El usuario podrá buscar la tarea por su nombre	

Página Tareas	Orden de las tareas	
Descripción	El usuario podrá ordenar de manera ascendente o descendente las tareas	

Página Tareas	Crear tarea	
Descripción	El usuario podrá crear una nueva tarea	
Acciones	Pasos	Acción
	1	El usuario hace click en el botón para añadir
	2	El sistema le presenta un ventana con las distintas partes de la tarea
	3	El usuario acepta y se crea la tarea

Página Tareas	Editar tarea	
Descripción	El usuario podrá editar la tarea seleccionada	
Acciones	Pasos	Acción
	1	El usuario hace click en el botón para editar
	2	El sistema abrirá las distintas partes de la tarea en un textarea para poder editar
	3	El usuario acepta el cambio
	4	Si el usuario no edita nada, no se hará ningún cambio
Condiciones	1- La tarea tiene que existir	

Página Tarea	Borrar tarea	
Descripción	El usuario podrá borrar la tarea seleccionada	
Acciones	Pasos	Acción
	1	El usuario hace click en el botón para borrar
	2	El sistema abrirá una ventana de confirmación de la acción
	3	Si el usuario no confirma, no se borrará la tarea
Condiciones	1- La tarea tiene que existir	

Página Tarea	Ver tarea
Descripción	El usuario podrá ver los datos de la tarea

Página Tarea	Paginado de Tareas
Descripción	El usuario podrá pasar de páginas una vez el mínimo de tareas creadas sea superado

Acciones realizadas

Puesta en marcha del entorno de trabajo

Dificultad al conectar el IDE con el código, conectar el código programado con la instancia docker.

Dificultad a la hora de crear las estructuras del mockup por temas de css y js.

Una vez decidida la arquitectura de máquinas a implementar en docker, el mayor problema que me he encontrado es conseguir conectar el código que se va a ejecutar en la instancia de PHP con el IDE de trabajo. Para poder hacerlo, en vez de utilizar una imagen docker genérica de PHP, definimos un fichero Dockerfile específico en el que a parte de instalar librería necesaria y la herramienta *composer* para la gestión de librería de terceros de Symfony, especificamos de forma directa la ubicación de código en esa máquina:

```
WORKDIR /var/www/symfony
```

De esta manera dentro del fichero docker-compose.yml, pudimos enlazar los volúmenes adecuados.

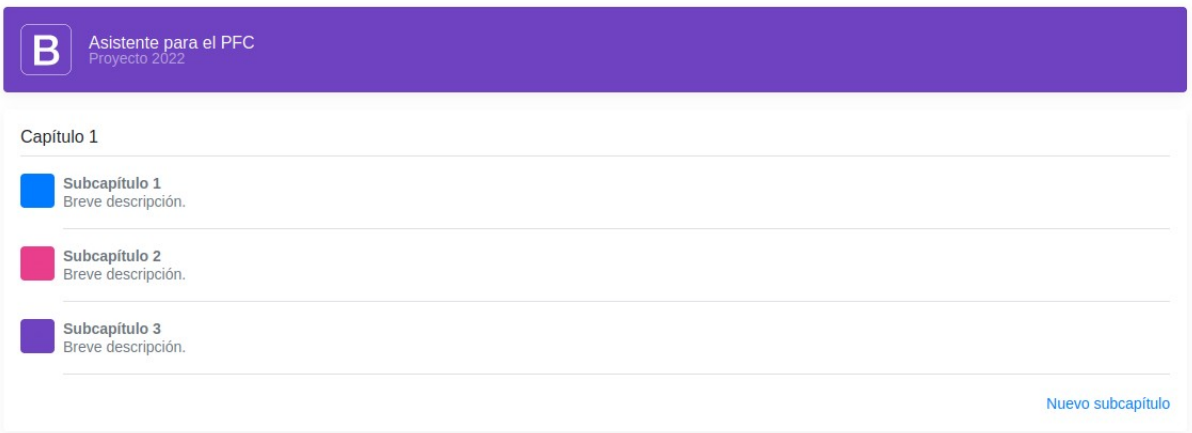
```
volumes:
  - ../:/var/www/symfony/
```

Mockup de la aplicación

La aplicación va a disponer de un menú en la parte superior con enlace a las funcionalidades principales. Incluye un buscador por texto para el documento.



La zona del proyecto tendrá un diseño parecido a este:



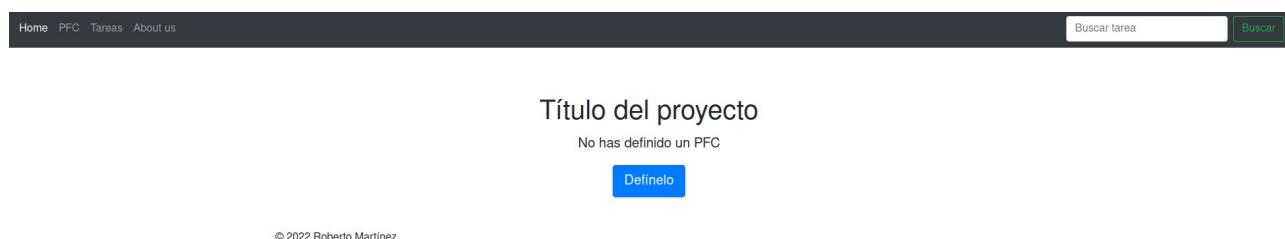
Y la zona de tareas así:



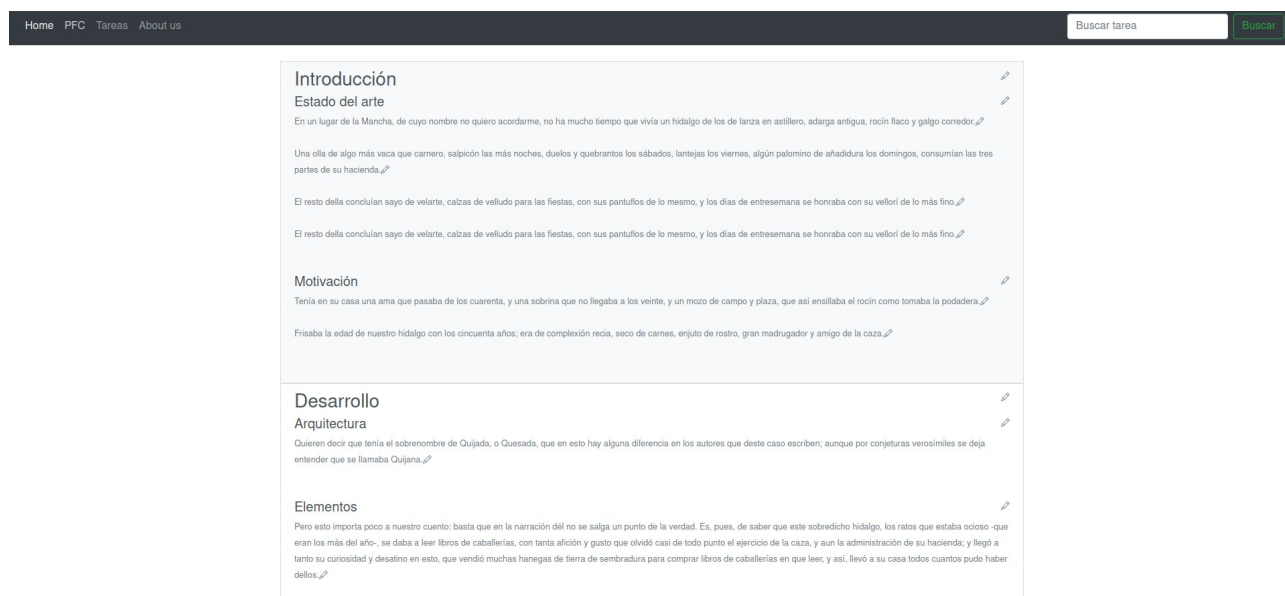
Implementación de las funcionalidades

Una vez finalizada la idea de cómo sería el mockup de la aplicación y pensando en las distintas funcionalidades que tendría, se ha cambiado el mockup anterior por la implementación de Vue en el proyecto, a la vez que se implementaron los distintos botones para las distintas funcionalidades de las distintas páginas del proyecto.

Estos serían los distintos mockups de las distintas páginas del proyecto con los distintos botones funcionales. Para la página Home, en ella se vería el documento del proyecto. Al no haber nada, se muestra un mensaje y un botón para ir a la zona del proyecto para crear el documento.



El botón te lleva a la zona del proyecto en donde se creará las distintas partes del documento y en donde cara al futuro se podrá editar. Este sería el diseño con las distintas funcionalidades y ejemplo de como se vería:



Se han puesto unos ejemplos para poder observar los distintos elementos y cambios que se vayan realizando durante el proceso de elaboración del documento.

Y por último el prototipo de la zona de creación de tareas con las distintas funcionalidades. El estilo de la página sería el siguiente:

Ordena

sin orden

Asc

Orden

asc

Buscar

Limpiar

Filtrar por ☐ Nombre ☐ Prioridad ☐ Completada

Por página

5

Tareas: 12

<<

<

1

2

3

>

>>

+

Nombre	Descripción	Prioridad	Completada	Actions
Dickerson Macdonald	Lorem ipsum	Alta	Yes	<div><div></div><div></div><div></div><div></div></div>
Larsen Shaw	Mauris auctor mattis	Media	No	<div><div></div><div></div><div></div><div></div></div>
Mini Navarro	Proin fermentum mi sed	Baja	No	<div><div></div><div></div><div></div><div></div></div>
Geneva Wilson	Nam in tempor massa.	Alta	No	<div><div></div><div></div><div></div><div></div></div>
Jami Carney	In aliquet lacinia ornare	Baja	Yes	<div><div></div><div></div><div></div><div></div></div>

© 2022 Roberto Martínez

Como se puede apreciar, hay distintas funciones. Unas principales que serían las de añadir, editar, borrar y ver tiempo de la tarea, y otras secundarias que serían los distintos filtros para la búsqueda de las tareas.

Cada filtro tiene una funcionalidad definida. En ellos puedes ordenar las distintas tareas por Nombre, Descripción, Prioridad y si está completada. También se puede ordenar en orden ascendente o descendente.

Ordena

Nombre

Asc

Orden

asc

Filtrar

Limpiar

Filtrar por ☐ Nombre ☐ Prioridad ☐ Completada

Por página

5

Tareas: 0

<<

<

1

>

>>

+

Nombre	Descripción	Prioridad	Completada	Actions
Dick Dunlap	Curabitur lorem eros	Alta	No	<div><div></div><div></div><div></div><div></div></div>
Dickerson Macdonald	Lorem ipsum	Alta	Yes	<div><div></div><div></div><div></div><div></div></div>
Essie Dunlap	Cras interdum malesuada odio	Alta	No	<div><div></div><div></div><div></div><div></div></div>
Geneva Wilson	Nam in tempor massa.	Alta	No	<div><div></div><div></div><div></div><div></div></div>
Genevieve Wilson	Vestibulum ante ipsum primis	Media	No	<div><div></div><div></div><div></div><div></div></div>

© 2022 Roberto Martínez

Ordena

Descripción

Asc

Orden

asc

Filtrar

Limpiar

Filtrar por ☐ Nombre ☐ Prioridad ☐ Completada

Por página

5

Tareas: 0

<<

<

1

>





















>>

+

Nombre	Descripción	Prioridad	Completada	Actions
Essie Dunlap	Cras interdum malesuada odio	Alta	No	<div><div></div><div></div><div></div><div></div></div>
Larsen Shaw	Curabitur eget fermentum ex	Baja	Yes	<div><div></div><div></div><div></div><div></div></div>
Dick Dunlap	Curabitur lorem eros	Alta	No	<div><div></div><div></div><div></div><div></div></div>
Mitzi Navarro	Etiam ac lacus lorem.	Alta	No	<div><div></div><div></div><div></div><div></div></div>
John Carney	Fusce viverra non leo vel tincidunt	Baja	Yes	<div><div></div><div></div><div></div><div></div></div>





















© 2022 Roberto Martínez

Ordena	Prioridad	Asc	Orden	asc
Filtrar		Limpiar	Filtrar por	<input type="checkbox"/> Nombre <input type="checkbox"/> Prioridad <input type="checkbox"/> Completada
Por página	5		Tareas: 0	1

Nombre	Descripción	Prioridad	Completada	Actions
Dickerson Macdonald	Lorem ipsum	Alta	Yes	   
Geneva Wilson	Nam in tempor massa.	Alta	No	   
Essie Dunlap	Cras interdum malesuada odio	Alta	No	   
Thor Macdonald	Vivamus pharetra arcu id neque	Alta	Yes	   
Mitzi Navarro	Etiam ac lacus lorem.	Alta	No	   

© 2022 Roberto Martínez

Ordena	Completada	Asc	Orden	asc
Filtrar		Limpiar	Filtrar por	<input type="checkbox"/> Nombre <input type="checkbox"/> Prioridad <input type="checkbox"/> Completada
Por página	5		Tareas: 0	1

Nombre	Descripción	Prioridad	Completada	Actions
Larsen Shaw	Mauris auctor mattis	Media	No	   
Mini Navarro	Proin fermentum mi sed	Baja	No	   
Geneva Wilson	Nam in tempor massa.	Alta	No	   
Essie Dunlap	Cras interdum malesuada odio	Alta	No	   
Mitzi Navarro	Etiam ac lacus lorem.	Alta	No	   

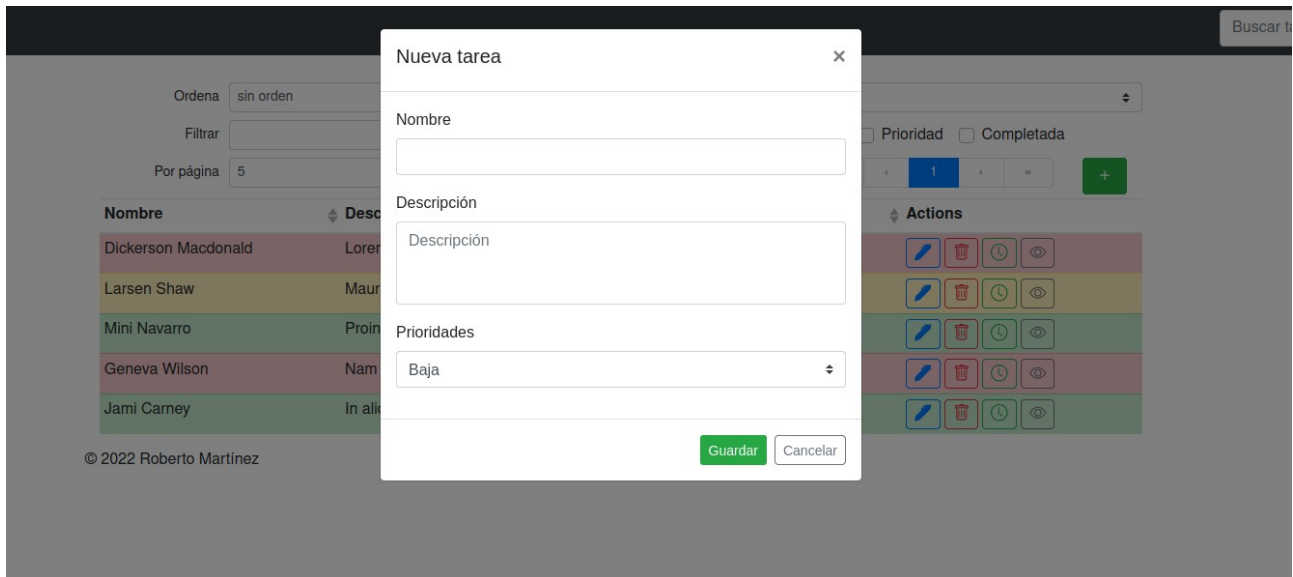
© 2022 Roberto Martínez

También hay filtros para ordenar por las mismas categorías que el filtro anterior pero con estos buscas en específico la tarea que quieras seleccionar, ya sea por nombre, prioridad o si está completada.

Se puede modificar el tamaño de la lista de las tareas para que se vean un determinado número de ellas junto con un marcador del número total de tareas que se hayan realizado.

Para las funciones principales, la forma de utilizarlos es a base de modales, es decir, en vez de que la función vaya a otra página distinta, está configurado para que salga una ventana emergente para trabajar directamente sobre ellas.

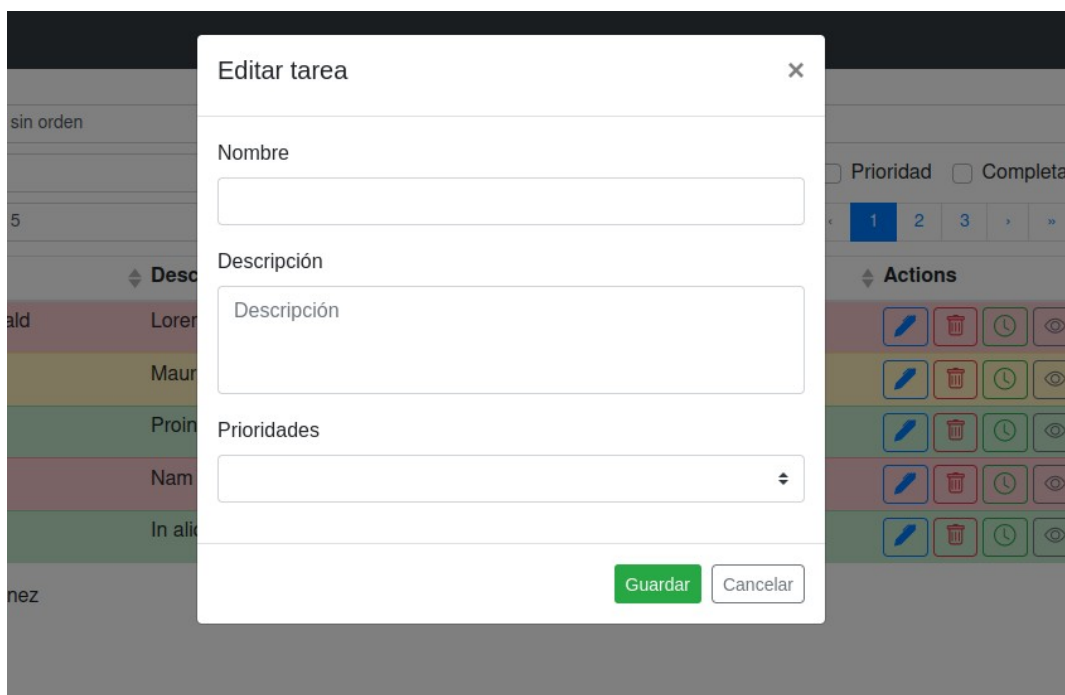
El modal de añadir tarea tiene el siguiente diseño:



The image shows a modal window titled "Nueva tarea" (New task) with a close button (X) in the top right corner. The form contains three input fields: "Nombre" (Name), "Descripción" (Description), and "Prioridades" (Priorities). The "Prioridades" field is a dropdown menu currently showing "Baja" (Low). At the bottom right of the modal are two buttons: "Guardar" (Save) in green and "Cancelar" (Cancel) in white. The background shows a blurred view of a task management interface with a table of tasks and an "Actions" column.

En ella se podrá poner nombre, una descripción detallando lo que se hará en la tarea y una opción para elegir la prioridad con que se tengan que hacer las tareas.

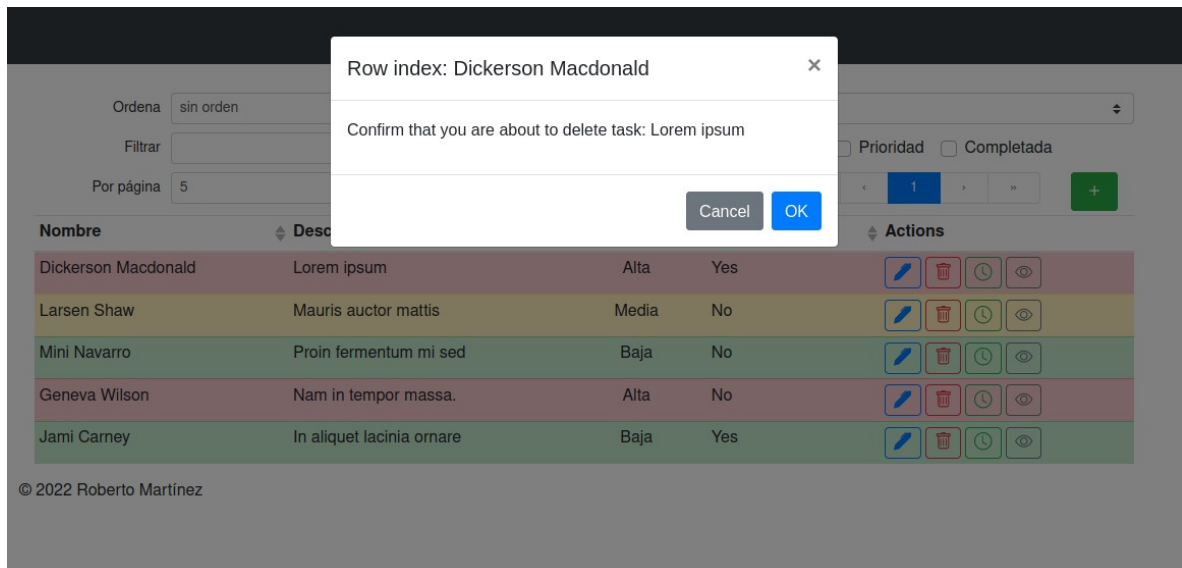
La funcionalidad de editar la tarea todavía esta en construcción ya que se detectó un problema para mostrar los datos en el modal y poder editar los distintos datos creados cuando se creó la tarea. De todos modos la parte visual del modal esta creada y se vería así:



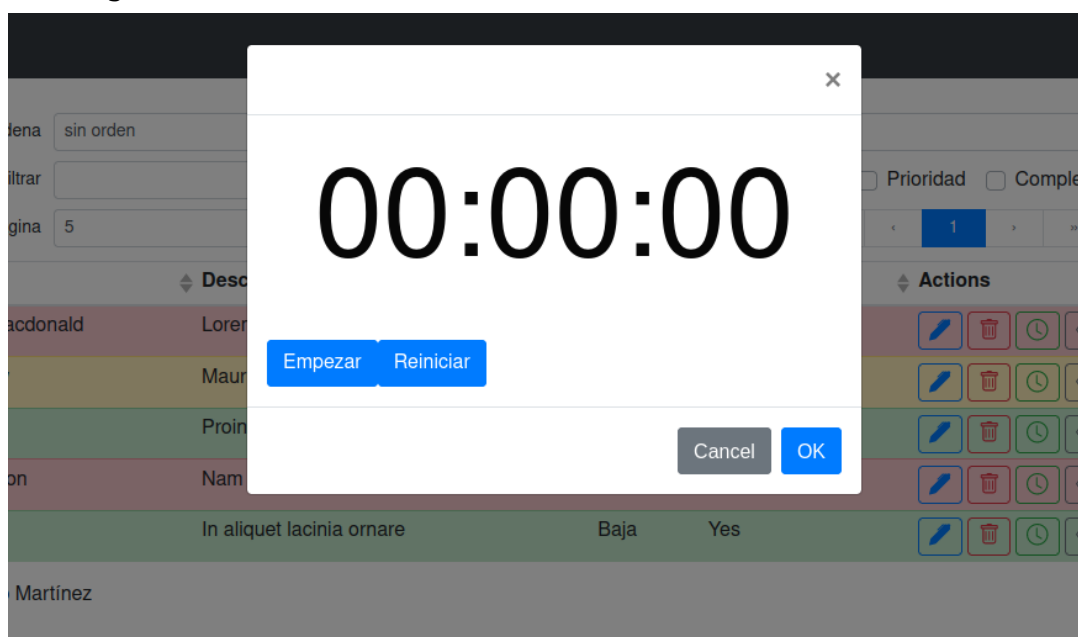
The image shows a modal window titled "Editar tarea" (Edit task) with a close button (X) in the top right corner. The form contains three input fields: "Nombre" (Name), "Descripción" (Description), and "Prioridades" (Priorities). The "Prioridades" field is a dropdown menu currently showing "Baja" (Low). At the bottom right of the modal are two buttons: "Guardar" (Save) in green and "Cancelar" (Cancel) in white. The background shows a blurred view of a task management interface with a table of tasks and an "Actions" column.

Viendo los distintos problemas que este teniendo el poder conectar esta función con el store interno, se cambiará el formato de la misma a un estilo mucho más sencillo y funcional que el anterior ya que no será un modal sino se realizarán los cambios directamente sobre la tabla.

La función de borrar por ahora esta desactivada porque necesita estar conectada al store de las tareas para que las borre del todo y no solo visualmente y una vez se recargue la página desaparezca por completo y no en ese momento. Al igual que el resto de funciones, tiene un modal de confirmación de borrado de tarea por si te equivocas de tarea y borras la que no se quería borrar.



Y por último la función de tiempo de tarea es un cronómetro en formato modal para cada tarea para registrar el tiempo dedicado a cada una de las tareas. Se puede parar el tiempo y ver el histórico de en que momento se ha hecho una pausa. Tanto esta como la anterior función tienen que ser conectadas al store de la página para poder guardar el tiempo y poder borrar la tarea cuando sea necesario. Su estilo visual es el siguiente:



Revisando la estructura del apartado del PFC, se ha podido observar que la estructura que se estaba planteando de Capítulo, Subcapítulo y Párrafo daría problemas a la hora de implementar las distintas acciones ya que se tendrían que crear funciones y botones para todos los apartados del mismo. Por esta razón, se ha cambiado la estructura a Capítulo, Subcapítulo y texto libre con la implementación de un RichTextElement (un editor de texto básico de VUE) para simplificar el proceso de edición del texto y solamente crear una sola función en vez de muchas para cada uno de los apartados del proyecto.

A la vez, seguramente los modales se cambiarán a funciones que toquen directamente a los distintos apartados de la página para hacer que los procesos sean más ligeros y no tengan bastante carga.

Resultados y Análisis

Conseguida la instalación del entorno de trabajo y todas las funcionalidades relacionadas con la base de datos.

Arquitectura de la aplicación

Estructura del proyecto

La estructura de la aplicación es de un proyecto Symfony. Las principales carpetas son:

- bin: contiene los archivos referentes a los comandos ejecutables en terminal
- config: contiene los archivos de configuración de la aplicación
- public: punto donde la aplicación se instancia para que se vea en el navegador
- src: contiene el código fuente de la aplicación
 - Controller: contiene los controladores que dan funcionalidad a la aplicación.
 - Entity: contiene los distintos ficheros .git
 - Repository: contiene los distintos ficheros .gitignore
- templates: contiene las distintas plantillas Twig
- var: contiene los archivos de caché, logs
- vendor: contiene el código de Symfony y las librerías externas.

Manual de despliegue

Instrucciones de despliegue

Lo primero que debemos hacer es instalar en el equipo servidor Docker y Docker Compose².

Dockerfile

```
FROM php:8.1-fpm-buster
ARG TIMEZONE
COPY php.ini /usr/local/etc/php/conf.d/docker-php-config.ini
RUN apt-get update && apt-get install -y \
    gnupg \
    g++ \
    procps \
```

² Mirar en Referencias donde se indica el como instalar y configurar tanto el Dockerfile como el docker-compose en Linux y Windows. Ver **Referencia 2**

```

openssl \
git \
unzip \
zlib1g-dev \
libzip-dev \
libfreetype6-dev \
libpng-dev \
libjpeg-dev \
libcuc-dev \
libonig-dev \
libxslt1-dev \
acl \
&& echo 'alias sf="php bin/console"' >> ~/.bashrc

```

RUN docker-php-ext-configure gd --with-jpeg --with-freetype

RUN docker-php-ext-install \

```

    pdo pdo_mysql zip xsl gd intl opcache exif mbstring

```

Set timezone

```

RUN ln -snf /usr/share/zoneinfo/${TIMEZONE} /etc/localtime && echo ${TIMEZONE} >
/etc/timezone \

```

```

    && printf '[PHP]\ndate.timezone    =    "%s"\n',    ${TIMEZONE}    >
/usr/local/etc/php/conf.d/tzone.ini \
    && "date"

```

```

RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --
filename=composer

```

```

WORKDIR /var/www/symfony

```

docker-compose.yml

```

version: "3"

```

```

services:

```

```

    db:

```

```

        image: mysql:8.0.23

```

```

        platform: linux/x86_64

```

```

        command: --default-authentication-plugin=mysql_native_password

```

```

        volumes:

```

```

            - ../docker_data/mysql:/var/lib/mysql"

```

```

        environment:

```

```

            MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}

```

```

            MYSQL_DATABASE: ${MYSQL_DATABASE}

```

```

            MYSQL_USER: ${MYSQL_USER}

```

```

            MYSQL_PASSWORD: ${MYSQL_PASSWORD}

```

```

        ports:

```

```

            - 3306:3306

```

```

        networks:

```

```

            - symfony

```

```

    php:

```

```

        build:

```

```

            context: ./php

```

```

            args:

```

```

                TIMEZONE: ${TIMEZONE}

```

```

        volumes:

```

```

            - ../var/www/symfony/

```

```

        networks:

```

```

            - symfony

```

```

    nginx:

```

```

        build:

```

```

            context: ./nginx

```

```

        volumes:

```

```

            - ../var/www/symfony/

```

```
ports:
  - 9080:80
networks:
  - symfony
env_file:
  - .env.nginx
```

```
networks:
  symfony:
```

Para iniciar la aplicación debemos entrar en el directorio donde se encuentre el **docker-compose.yml** mediante consola. Una vez dentro, ejecutamos el comando:

```
$ docker-compose up -d
```

Se iniciarán las distintas partes de la aplicación, aunque para poder ver la aplicación se necesita ejecutar el siguiente comando para poder mostrar en pantalla la aplicación.

```
$ docker-compose exec frontend bash
```

y una vez dentro, el siguiente comando:

```
npm run serve
```

Una vez cargados todos los datos, pondremos la siguiente url: **http://localhost:8080**. Esta url nos llevará al home de la aplicación del frontend y desde ahí se pondrán ver los distintos cambios que se vayan realizando en el código de la aplicación.

Si ya se terminó de utilizar la aplicación, para cerrar la aplicación solamente hay que cerrar navegador y en la consola ejecutar el siguiente comando:

```
$ exit
```

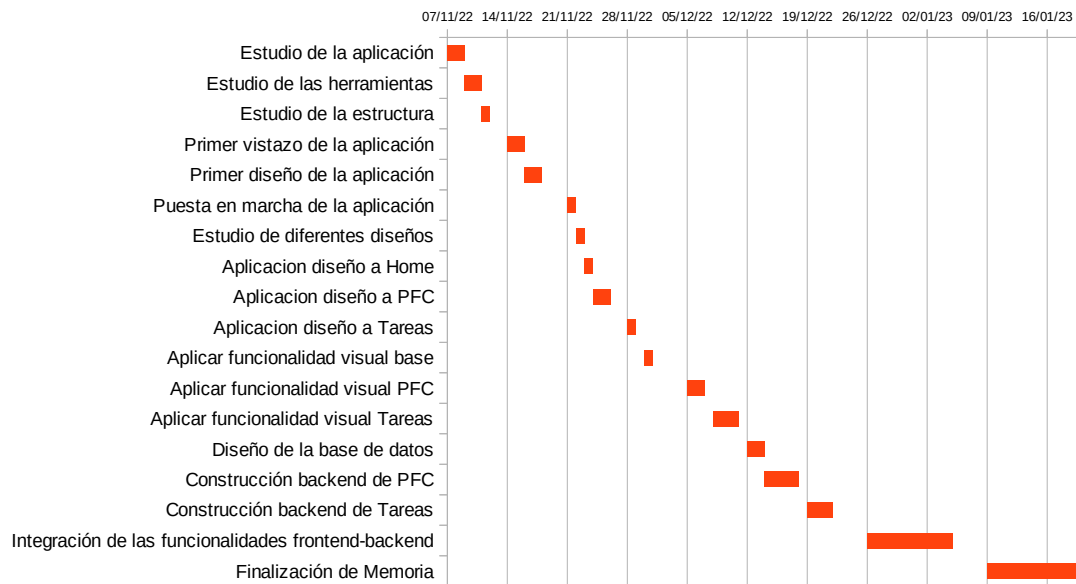
que cerrará la zona del *bash* y posteriormente:

```
$ docker-compose down
```

y se cerrarán las distintas partes de la aplicación.

Temporalización

En la siguiente imagen se muestra el Diagrama de Gantt con la previsión del trabajo a realizar:



En realidad el esfuerzo se centró en las primeras semanas del proyecto y en las dos semanas finales de 2022 y primera de 2023, dado que tuve que compaginar el desarrollo del proyecto con las prácticas en la empresa.

Conclusiones

A medida que se van implementando las distintas funcionalidades, se van cambiando los distintos formatos debido a la implementación de los componentes de Bootstrap y Bootstrap-Vue para conseguir que funcionen y se mantengan los datos sin tener que irse a otra página y hacer el proceso mucho más ligero y hacer que la experiencia de usuario (ux) sea lo más cómoda y funcional posible.

También a medida que se va avanzando en el proyecto, se descubren nuevas dificultades que el tener que resolverlas conlleva distintos cambios ya mencionados anteriormente en el proyecto y hace que se tenga que dedicar más tiempo de lo esperado a la solución de ellos.

Líneas de investigación futuras

Mirando las distintas posibilidades de mejoras que se podrían implementar en un futuro, estas serían algunas de las muchas implementaciones que se podrían añadir:

- Más implementaciones en la memoria y posibles cambios visuales y funcionales.
- Meter tablas en el contenido del documento.
- Meter imágenes en el contenido del documento.
- Histórico de asignación de tiempo por tareas.
- Observaciones en las tareas, estados de finalización.
- Crear dependencias entre tareas.
- Hacer que el editor pueda incluir mas elemnetos como imágenes, tablas, etc.
- Presentar errores en frontend del backend.

Bibliografía

[Referencia MVC] Modelo vista controlador (MVC)
<https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

[Referencia 2] Como instalar docker en Windows
<https://docs.docker.com/desktop/install/windows-install/>

[Referencia 2] Como instalar docker en Linux
<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04-es>

Estructura del fichero <https://github.com/dhurtadorosales/Final-Project-DAW>

Enlace del diseño de Bootstrap
<https://getbootstrap.com/docs/4.0/examples/offcanvas/#>

Cronómetro <https://vuejsexamples.com/a-simple-vue-timer-with-javascript-css-and-html/>

Manual bootstrap-vue <https://bootstrap-vue.org/docs>

Manual editor de texto Vue <https://www.vue2editor.com/>

Diferentes tipos de editores de Vue <https://www.vuescript.com/best-wysiwyg-rich-text-editor/>

Manual de Vue <https://es.vuejs.org/v2/guide/>

Repositorio GitHub <https://github.com/RobyRobusto83/AplicacionAyudaPFC>