



PROYECTO FINAL DE CICLO

MARC GARCÍA TEODORO - 2º DAW
MUSICAL APPS



ÍNDICE

ESTRUCTURA DEL DOCUMENTO

1. Introducción
2. Gestión
3. Transición
4. Detalles de diseño
5. Diseños originales
6. Navegación
7. Tecnologías
8. Base de datos
9. Evolución
10. Bibliografía



INTRODUCCIÓN

DESDE EL INICIO HASTA EL FINAL

Objetivo

Un portal web que sirva de escaparate para los musicales del centro IES L'OM, con información de todos los musicales realizados, galería multimedia y un lado de administración para dicho portal.

Justificación

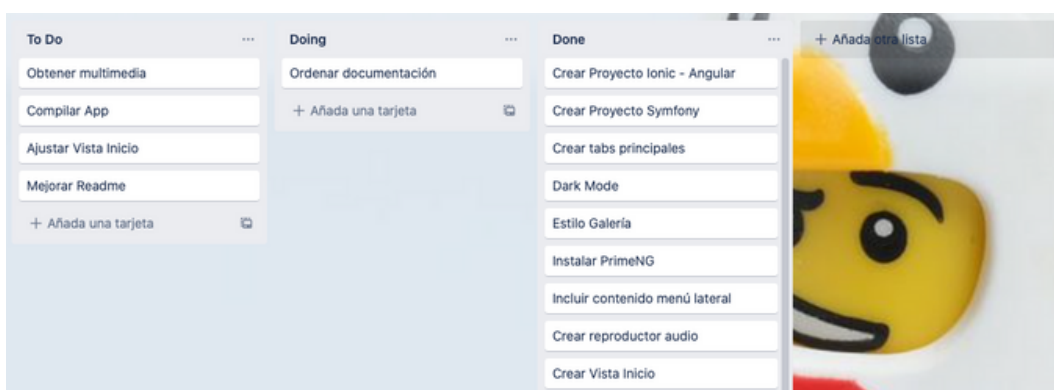
El instituto IES L'OM de Picassent es el centro de estudios en el que cursé el bachillerato y en ese tiempo me vi involucrado en un musical realizado en dicho centro, un proyecto sin ánimo de lucro en el que se insta a los alumnos a cantar, bailar y actuar mientras se trabaja con dinámicas de grupo y alumnado de diferentes cursos tiene la oportunidad de interactuar y desarrollar competencias como hablar de cara al público. Con la realización de esta aplicación, conseguirían visibilidad y un lugar al que poder acceder los antiguos participantes para ver todo lo relacionado con sus musicales.

Quisiera aprovechar este proyecto para trabajar con Ionic, tecnología que estoy empleando en las prácticas, lo que iría de la mano con Angular y librerías de dicho framework. El lado backend sería desarrollado en Symfony, tecnología trabajada durante este curso.

GESTIÓN

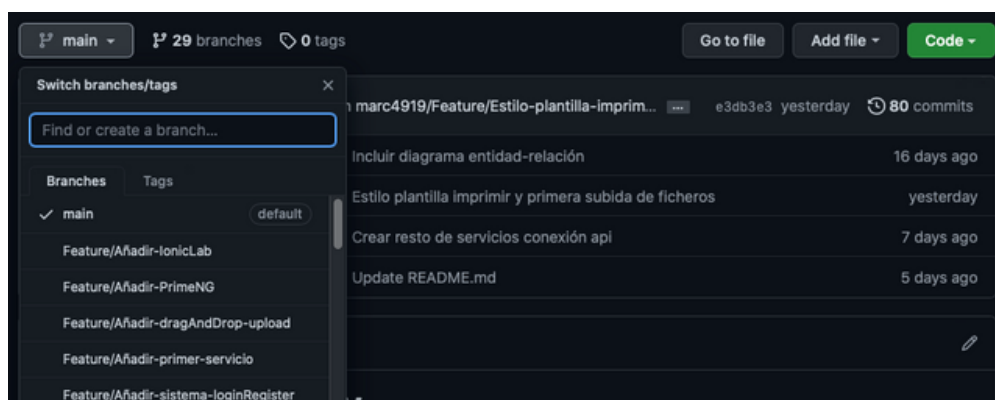
ORGANIZACIÓN DEL PROYECTO

Para la gestión del proyecto he empleado Trello, una herramienta para organizar flujos de trabajo y hacer un seguimiento del progreso de una manera visual. Para la distribución he optado por las 3 columnas de kanban estandarizadas to do, doing, done.



Por otro lado, en pos de utilizar un sistema de control de versiones he utilizado GitHub, enfocándome en el uso de una rama por cada una de las funcionalidades implementadas, garantizando así una gestión correcta evitando conflictos y pudiendo retroceder con facilidad a un punto concreto del proyecto.

<https://github.com/marc4919/Proyecto-Final-DAW>





TRANSICIÓN

MOBILE DESIGN



DE DESKTOP A MOBILE - EL CLIENTE OBJETIVO

Tras analizar detenidamente el uso que se le iba a dar a la aplicación llegué a la conclusión de que su público mayoritario iba a estar conformado por gente joven que participa o ha participado en los musicales y también por los amigos o familiares de estos. Teniendo eso en cuenta y que la finalidad de la aplicación es de consumo puesto que es posible navegar entre los distintos musicales y sus respectivas imágenes o canciones, manteniendo que en la actualidad los smartphones son los dispositivos de consumo de contenido más habituales, supuso un paso lógico enfocarse en una aplicación móvil.

Dicha aplicación está basada en el desarrollo híbrido, lo que me supuso la posibilidad de un desarrollo web adaptado, pudiendo emplear los conocimientos de las tecnologías fundamentales en el desarrollo web que había adquirido, para la realización de una aplicación que puede ser exportada tanto para ios como para android con un solo código basado en html, css y js. Ionic junto con Angular y los componentes de estos son los que han permitido un desarrollo fluido sin necesidad de conocer tecnologías nativas de los sistemas móviles.

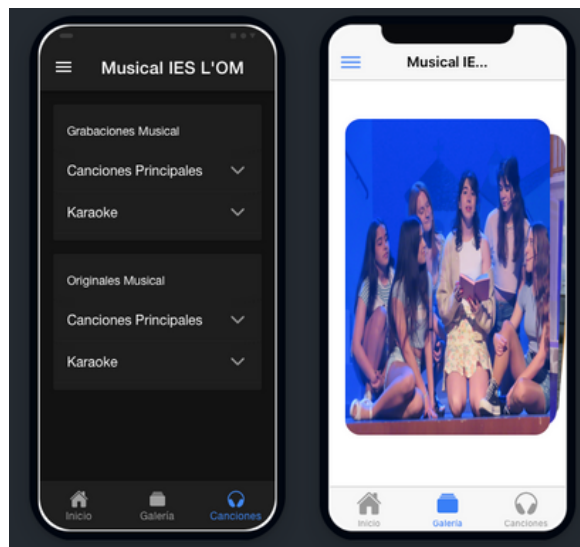


DETALLES DISEÑO

EL ESTILO

Para la aplicación mobile se han utilizado iconos nativos de Ionic y botones de primeNG. El uso de estas tecnologías garantiza una adaptación correcta a las medidas de las pantallas de los smartphones. También destaco el uso de swiper para lograr una galería de imágenes distinta que puede recordar a la conocida aplicación de Tinder a la que muchos jóvenes están habituados.

Los iconos han sido escogidos de modo que son símbolos mayormente reconocibles. En cuanto a la accesibilidad, sin contar con las descripciones que obligatoriamente mantienen todas las imágenes que se suben a la base de datos, destacaría el dark mode, imprescindible en una aplicación móvil que garantiza portabilidad.



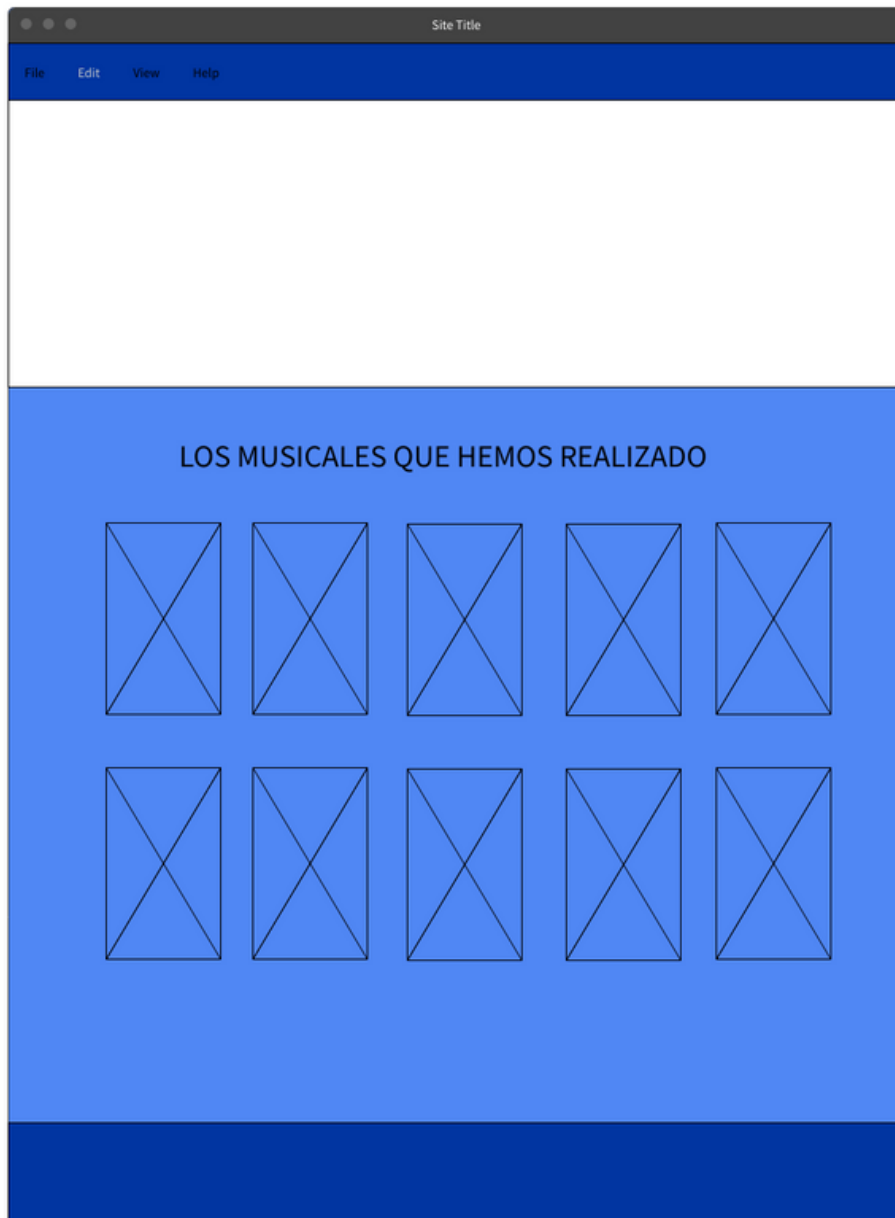
Para la aplicación de administración, las vistas se han gestionado mediante twig, combinado con el uso de bootstrap y css.

Los colores de la aplicación todavía no son los definitivos, puesto que debo todavía mediar con los gerentes del musical para una toma de decisiones sobre el estilo provisional que mantiene actualmente la página.



DISEÑOS ORIGINALES

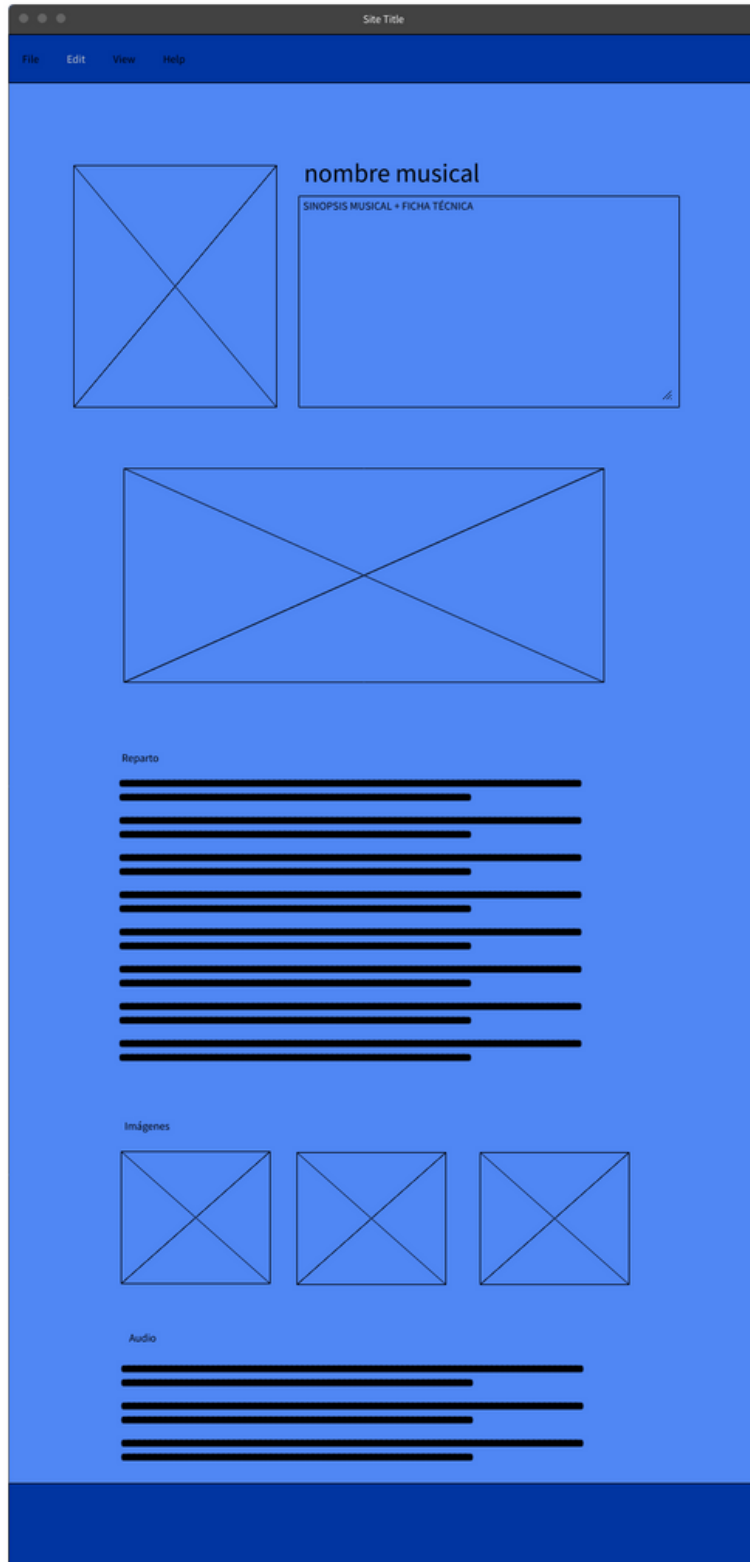
PÁGINA DE INICIO





DISEÑOS ORIGINALES

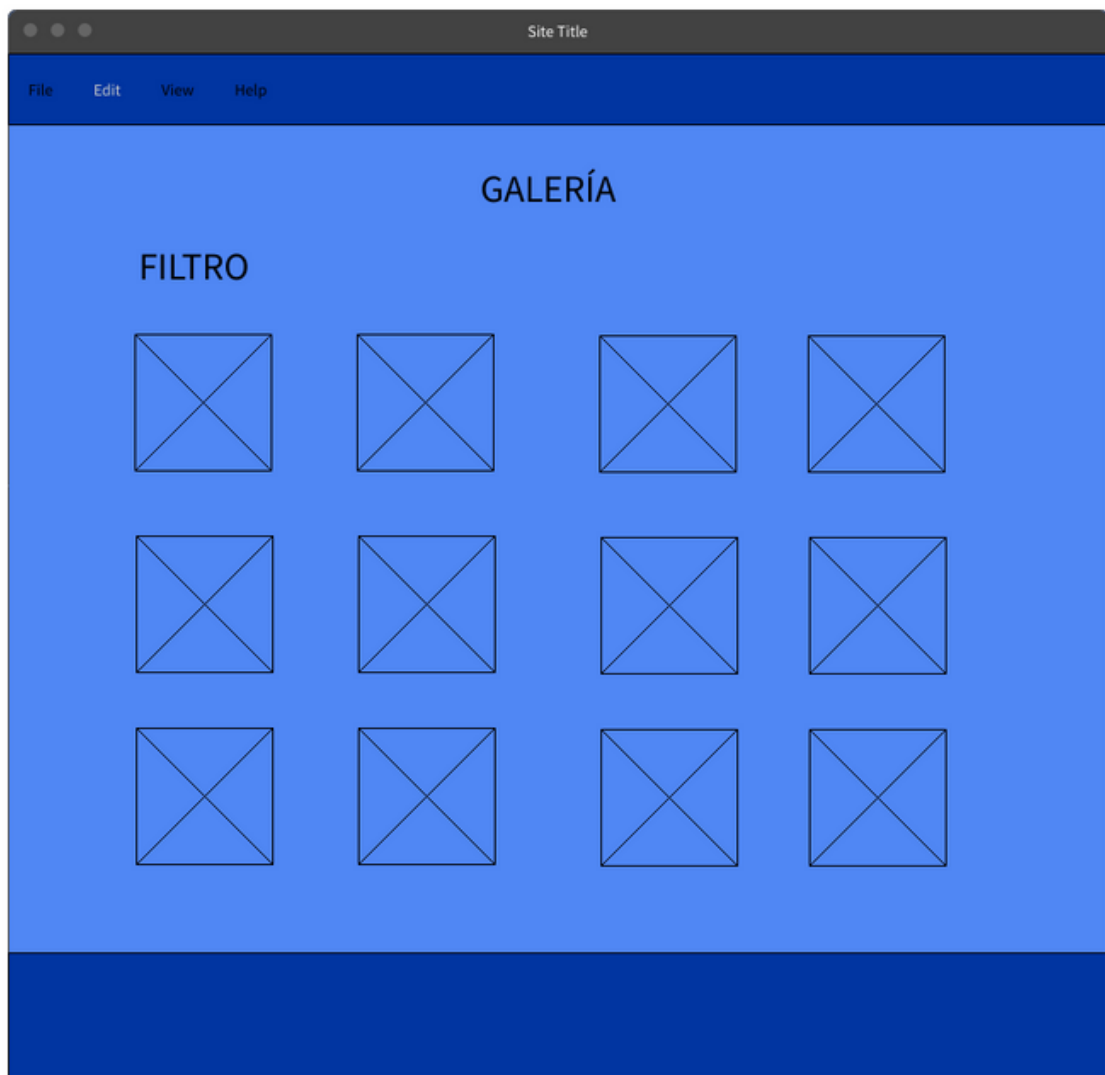
PÁGINA DE DETALLE





DISEÑOS ORIGINALES

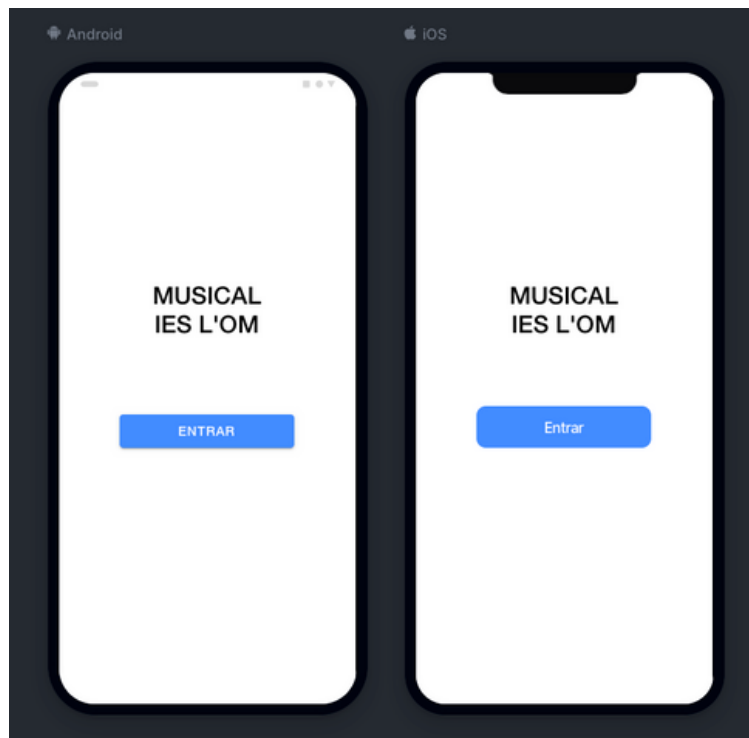
GALERÍA





NAVEGACIÓN

MOVIMIENTO ENTRE PÁGINAS



En la Aplicación Mobile encontramos una navegación basada en tabs, en la que una vez superada la página de inicio y habiendo entrado, se puede navegar entre inicio, galería y canciones a voluntad, sin ningún tipo de restricción



TECNOLOGÍAS

ESQUELETO Y MÚSCULO DEL PROYECTO



Symfony



ionic



webpack
MODULE BUNDLER



PRIME^{NG}



Desglose

Aplicación de administración: Symfony, Webpack, ApiPlatform, wkhtmltopdf

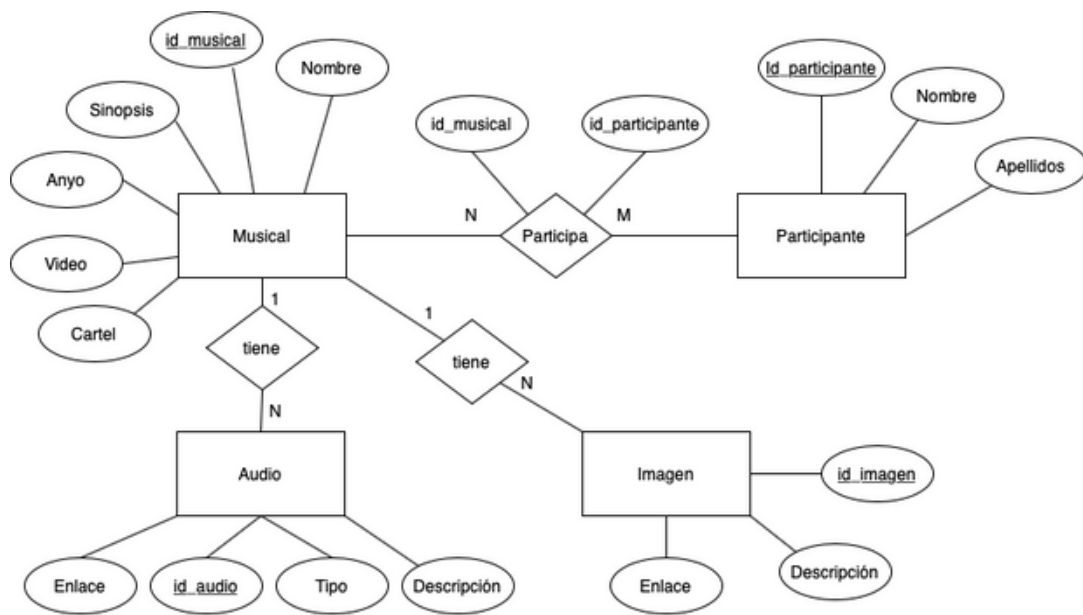
Aplicación mobile: Ionic, Angular, PrimeNG, Swiper

No incluyo aquellas tecnologías que forman parte de los propios frameworks, como puede ser twig para las plantillas. Para la estructura y el estilo ha resultado fundamental el uso de html, estilo aplicado mediante css, sass y bootstrap y por último TypeScript y php que son los lenguajes de Angular y Symfony.



BASE DE DATOS

MYSQL - BASE DE DATOS RELACIONAL



BBDD

Nos encontramos con una base de datos relacional que ha sido modelada mediante Symfony con uso de entidades y atributos, y que posteriormente ha sido migrada.



EVOLUCIÓN

DESDE EL INICIO HASTA EL FINAL

1. Creación del proyecto de Ionic utilizando Angular
2. Creación del sistema de Tabs para rooting de vistas
3. Añadido grid con cards para la página de inicio
4. Creación desplegados para las canciones
5. Creación menú lateral
6. Instalación PrimeNG para uso de componentes
7. Inclusión del dark mode de Ionic
8. Instalación Ionic Lab para visualizar el proyecto
9. Instalación de Swiper para galería de imágenes
10. Creación página de inicio previa a los tabs
11. Creación del proyecto de Symfony
12. Generación de Entidades con relaciones
13. Creación de tablas de base de datos mediante migración
14. Añadido de Api Platform al proyecto de Symfony
15. Creación sistema de subir ficheros
16. Ajuste del tamaño máximo de subida
17. Instalación de Webpack Encore para Symfony UX
18. Cambio de sistema de subida de ficheros a UX Dropzone
19. Creación de servicios en Angular para recibir datos de la Api
20. Solución CORS en Symfony
21. Maquetación plantillas de twig con uso de bootstrap
22. Añadido sistema de login y register
23. Configuración seguridad rutas Symfony
24. Traducción botones generados por Symfony
25. Instalación KnpSnappy - wkhtmltopdf para funcionalidad impresión
26. Añadido Estilo Plantilla Imprimir
27. Ajustado estructuras que reciben datos en angular



BIBLIOGRAFÍA

ENLACES A TENER EN CUENTA

- <https://symfony.com/doc/current/frontend/ux.html>
- <https://ionicframework.com/docs/components>
- <https://swiperjs.com>
- <https://devdocs.io/angular/guide/routing-overview>
- <https://api-platform.com/docs/>
- <https://latteandcode.medium.com/generando-pdfs-con-symfony-d34a516a45fe>