

# **Presentació Tècnica**

Aquest document conté una presentació tècnica del projecte, en el qual explicarem l'estructura principal i els components de l'aplicació final.

Aquest document només descriu de forma tècnica el projecte, per detalls tècnics cal consultar [el repositori](#).

## **Tecnologies utilitzades**

Aquest és un llistat de totes les tecnologies utilitzades durant el desenvolupament:

- PHP 7.3

Hem escollit aquesta versió per la seva estabilitat i compatibilitat amb Symfony.

- Apache2

Per desenvolupar en local i poder desplegar aquesta aplicació web hem utilitzat Apache i l'hem configurat de forma que sigui el més senzill possible, complint amb normes bàsiques de seguretat.

- MySQL Server

Es necessita una base de dades per qualsevol projecte, en el nostre cas hem utilitzat MySQL per la seva facilitat d'ús, estructura i nivell de persistència.

- PhpMyAdmin

Ja que utilitzem MySQL hem escollit aquest programari per administrar de forma més senzilla i visual la base de dades.

- Symfony

El nostre projecte es basa completament en el framework Symfony, que ofereix un conjunt de components que faciliten la seva implementació i té compatibilitat amb qualsevol altre framework de Frontend. Ja que coneixem aquesta tecnologia hem basat l'estructura del projecte al voltant d'aquest framework.

- Composer

Ja que durant el desenvolupament del backend utilitzem Symfony hem escollit Composer com a gestor dels paquets de PHP que utilitzem al projecte.

- Webpack Encore

Webpack és un “module bundler” de Javascript i té la funció principal de gestionar, generar i construir recursos que seran consumits pel navegador. És capaç de gestionar tot tipus d'assets, inclosos Javascript, CSS i fitxers de tipus de lletra. En el nostre cas utilitzem Webpack Encore, que adapta la tecnologia Webpack a Symfony i permet una integració fluida entre frontend i backend. També ofereix un servidor web de desenvolupament que permet fer canvis als assets i carregar noves versions d'aquests de forma automàtica i reactiva.

- Doctrine

El nostre projecte basa l'estructura de dades utilitzant el model ORM (Object Oriented Model). Aquest model tradueix la lògica interna de la plataforma i les diferents entitats que hi participen en el conjunt de taules i continguts que es troben en la base de dades. Gràcies a l'utilització d'aquest model podem interaccionar d'una forma directa amb la BD i gestionar tota la limitació de les dades en un sol lloc, a les entitats al codi. I Doctrine és la tecnologia que ens permet fer precisament això, ja que ofereix scripts per realitzar diferents processos i és molt compatible amb la nostra estructura de backend.

- Twig

Per la renderització i generació de text HTML utilitzem Twig, un motor de plantilles basat en PHP que ofereix la seva pròpia estructura de fitxers. Aquest component ofereix un conjunt d'eines que ens faciliten la renderització de dades i ens permeten integrar lògica en aquesta. Ja que utilitzem Symfony aprofitem la seva compatibilitat i fàcil integració amb aquest component.

- Node

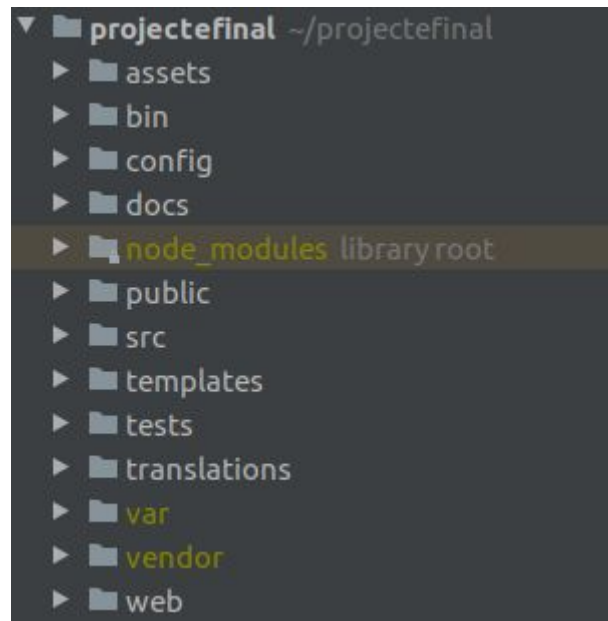
Hem utilitzat part de les funcionalitats de Node per aprofitar al màxim les característiques de Webpack Encore al nostre projecte.

- Yarn

Pel desenvolupament del frontend hem utilitzat Yarn per la gestió de paquets, plugins i llibreries en Javascript. Hem preferit aquest sobre NPM ja que ofereix millor compatibilitat amb la nostra decisió d'utilitzar Symfony+Webpack Encore.

## Estructura de fitxers: directoris importants

En la imatge següent es pot visualitzar l'estructura de fitxers del nostre projecte:



En aquest apartat només explicarem els directoris més rellevants:

- *assets/*

En aquest directori es troben els assets o recursos de JS i CSS que el Webpack utilitza per generar els assets finals que veu un navegador. Els fitxers JS representen els punts d'entrada de cada asset i defineixen les importacions de plugins i llibreries que es troben a la carpeta *node\_modules/*. Aquí es troba tot el codi relacionat amb el frontend del projecte.

- *docs/*

En aquest directori es troba la documentació relacionada amb el projecte.

- *node\_modules/*

En aquest directori es troben totes les dependències en Javascript que utilitza el projecte, bàsicament els paquets i plugins carregats pel gestor de paquets.

- *public/*

Aquest és l'únic directori públic i és el punt d'entrada inicial de la web. En aquest es troba el fitxer *index.php*, que inicia les funcionalitats de Symfony i carrega totes les rutes disponibles.

- *public/build*

En aquest directori es troben els assets generats pel Webpack i preparats per la seva consumició en un navegador.

- *src/*

En aquest directori es troba tot el codi relacionat amb el backend del projecte. Aquí es troben els diferents components que fan funcionar l'aplicació, dividits en Controladors, Entitats, Repositoris, Serveis, Migracions i DataFixtures. En la seva essència tots aquests components són classes en PHP, utilitzant els principis de la programació orientada a objectes.

- *templates/*

En aquest directori es troben els fitxers que Twig utilitzarà per renderitzar els continguts de la pàgina.

- *vendor*

En aquest directori es troben tots els paquets en PHP que necessita el nostre projecte, generats pel gestor de paquets Composer.

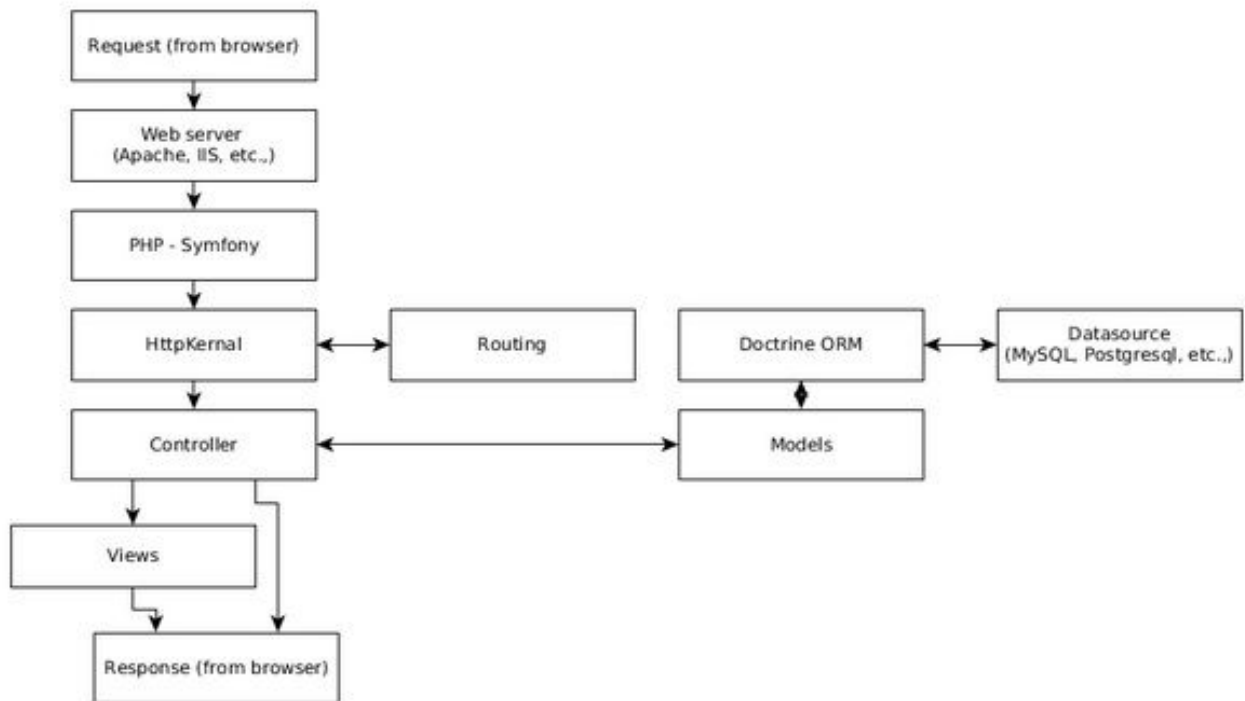
## Estructura de fitxers: fitxers de configuració

Aquests són els principals fitxers de configuració del projecte:

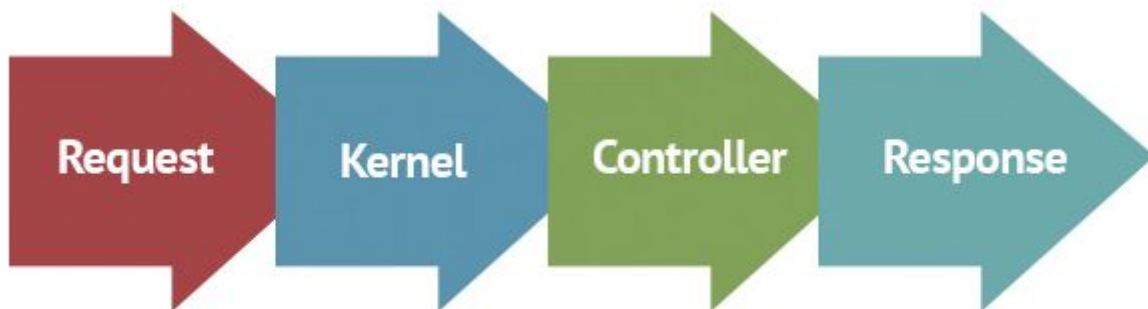
- *package.json* : Inclou les dependències en JS
- *composer.json* : Inclou les dependències en PHP
- *yarn.lock* : Inclou les dependències en JS pel gestor de paquets Yarn
- *symfony.lock* : Inclou la configuració dels components principals de Symfony
- *.env* : Inclou la configuració de l'entorn de treball i la configuració necessària per utilitzar Doctrine
- *webpack.config.js* : Inclou la configuració del Webpack del nostre projecte

## Diagrames del projecte

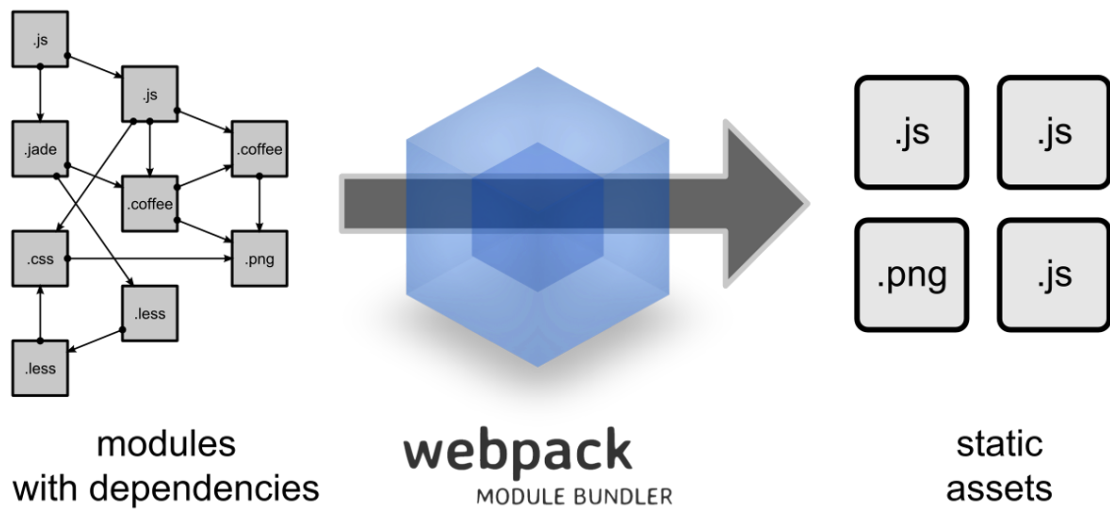
Aquest és un diagrama senzill que representa els diferents components que participen a la nostra aplicació web:



Aquest és el diagrama del flux quan es carrega un recurs en la nostra aplicació web:



## Diagrama del funcionament de Webpack:



## Diagrama resum dels components mencionats:

### Symfony + Webpack Encore

