

# Unitat 5. Frameworks. Laravel

2n DAW - IES María Enríquez

**Part 1**

# Resultats d'aprenentatge i criteris d'avaluació

## RA 5. Desenvolupa aplicacions Web identificant i aplicant mecanismes per a separar el codi de presentació de la lògica de negoci.

Criteris d'avaluació:

- a) S'han identificat els avantatges de separar la **lògica de negoci** dels aspectes de presentació de l'aplicació.
- b) S'han analitzat **tecnologies i mecanismes** que permeten realitzar aquesta separació i les seues característiques principals.
- c) S'han utilitzat **objectes i controls en el servidor** per a generar l'aspecte visual de l'aplicació web en el client.
- d) S'han utilitzat **formularis generats de manera** dinàmica per a respondre als esdeveniments de l'aplicació Web.
- e) S'han identificat i aplicat els **paràmetres relatius a la configuració** de l'aplicació Web.
- f) S'han escrit aplicacions Web amb **manteniment d'estat i separació de la lògica** de negoci.
- g) S'han aplicat els principis de la programació **orientada a objectes**.
- h) S'ha provat i **documentat** el codi.

# Resultats d'aprenentatge i criteris d'avaluació

**RA8. Genera pàgines web dinàmiques analitzant i utilitzant tecnologies i frameworks del servidor web que afigen codi al llenguatge de marques.**

Criteris d'avaluació:

- a) S'han identificat les diferències entre l'**execució de codi en el servidor i en el client web**.
- b) S'han reconegut els avantatges d'**unir totes dues tecnologies** en el procés de desenvolupament de programes.
- c) S'han identificat les **tecnologies i frameworks relacionades amb la generació per part del servidor** de pàgines web amb guions embeguts.
- d) S'han utilitzat aquestes **tecnologies i frameworks per a generar pàgines web** que incloguen **interacció** amb l'**usuari**.
- e) S'han utilitzat aquestes tecnologies i frameworks, per a generar pàgines web que incloguen **verificació** de formularis.
- f) S'han utilitzat aquestes tecnologies i frameworks per a generar pàgines web que incloguen **modificació dinàmica del seu contingut** i la seua estructura.
- g) S'han aplicat aquestes tecnologies i frameworks en la **programació d'aplicacions web**.

# Unitat 5. Frameworks. Laravel

## 1.Introducció

1.1. Frameworks en PHP

1.2. Preparació del programari

1.3. Primers passos amb Laravel



Laravel

# **1. Introducció**

# 1.1. Frameworks en PHP

Un **framework** és una **eina** que permet desenvolupar software proporcionant una sèrie de **components** que agilitzen el procés de creació d'una aplicació web:

- Model de dades
- Connexió a la base de dades
- Gestió de rutes
- Seguretat
- Autenticació

# 1.1. Frameworks en PHP

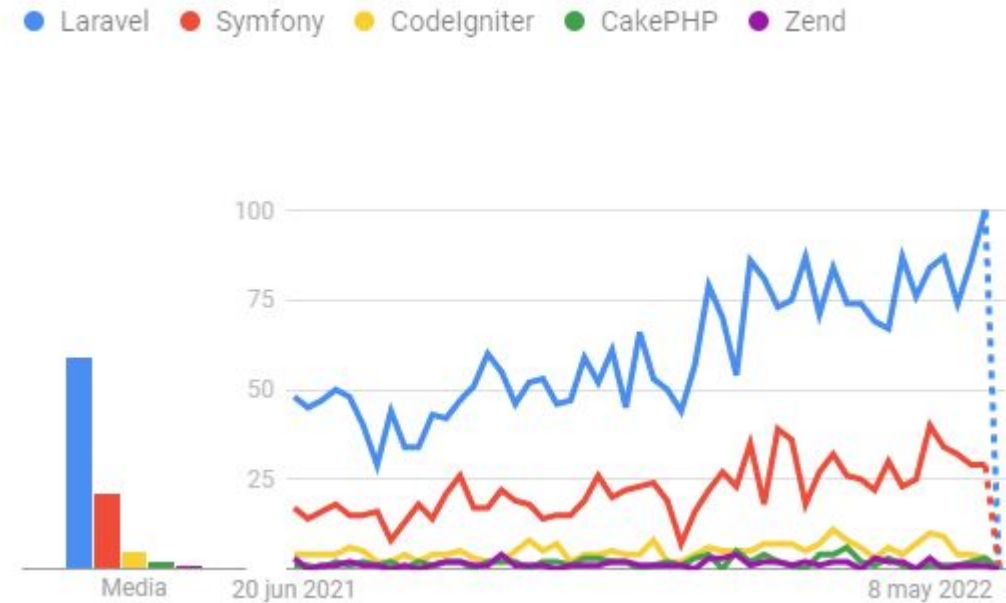
Alguns **exemples** de frameworks per a PHP són:

- Laravel (2011)
- Symfony (2005)
- Code Ignater
- Cake PHP
- Zend
- Phalcon (2012)

# 1.1. Frameworks en PHP

**Quin triar?** Podem tindre en compte:

- Popularitat
- Corva d'aprenentatge
- Demanda al món laboral
- Suport / Documentació
- Extensions / Compatibilitat



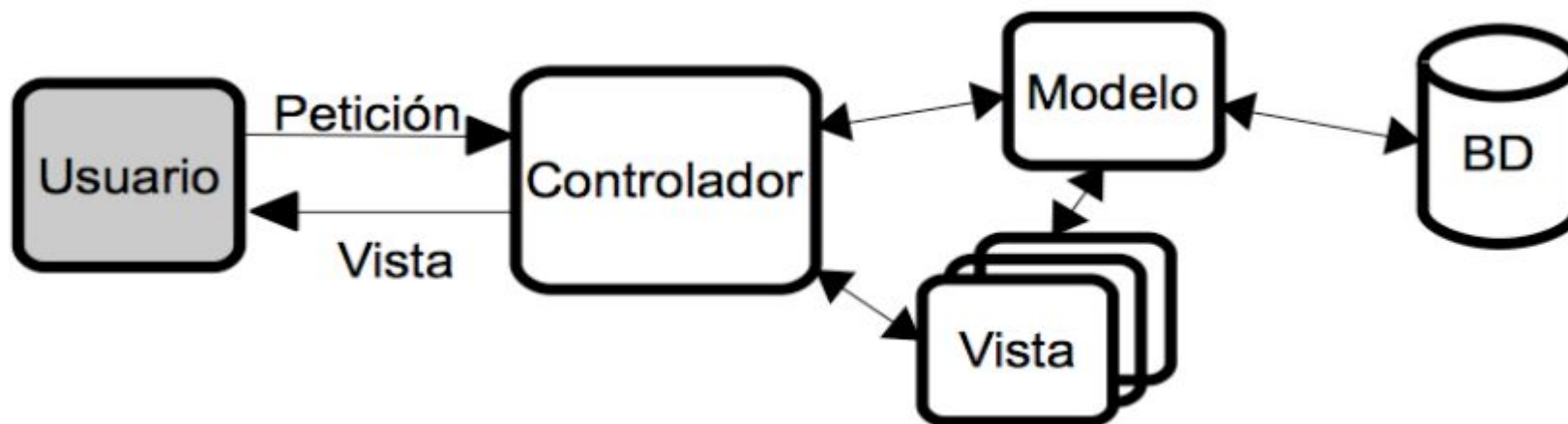
Framework	Ofertas encontradas
Laravel	107
Symfony	76
CodeIgniter	15
Zend	4
CakePHP	3



# 1.1. Frameworks en PHP

La majoria dels frameworks es basen en l'arquitectura **MVC**:

- **Model**: conjunt de dades que manipula l'aplicació.
- **Vista**: el que es mostra a l'usuari. Interfície gràfica.
- **Controlador**: lògica/operacions



## 1.2. Software necessari

- ✓ Visual Studio Code + Laravel Extension Pack
- ✓ Apache
- ✓ PHP. Versió 8.2 o superior.
- ✓ MySQL / MariaDB
- ✓ Composer: dependències servidor
- ✓ Npm: dependències client
- ✓ Test accés API (més endavant)

# 1.2. Software necessari.

## Instal·lació de PHP 8.3:

```
> sudo add-apt-repository ppa:ondrej/php  
> sudo apt update  
> sudo apt install php8.3  
> sudo update-alternatives --set php /usr/bin/php8.3  
> php -v
```

# 1.2. Software necessari.

## Instal·lació de Composer:

```
> sudo apt update
```

```
> sudo apt install php-cli unzip curl
```

```
> curl -sS https://getcomposer.org/installer -o  
composer-setup.php
```

```
> sudo php composer-setup.php
```

```
--install-dir=/usr/local/bin --filename=composer
```

```
> composer -v
```

# 1.2. Software necessari.

## Instal·lació de Laravel:

```
> composer self-update
```

```
> composer global require laravel/installer
```

```
> export PATH="$PATH:$HOME/.composer/vendor/bin"
```

```
> export PATH="$PATH:$HOME/.config/composer/vendor/bin"
```

```
> echo 'export
```

```
PATH="$PATH:$HOME/.config/composer/vendor/bin"' >>
```

```
~/ .bashrc
```

```
> source ~/ .bashrc
```

# 1.2. Software necessari.

## Instal·lació de Laravel:

```
> laravel new nombre_proyecto
```

```
> cd nombre_proyecto
```

```
> php artisan --version
```

```
> php artisan serve
```

<http://localhost:8000>

# 1.2. Software necessari.

## Instal·lació de node.js:

```
> sudo apt update
```

```
> sudo apt upgrade
```

```
> curl -fsSL https://deb.nodesource.com/setup_18.x |  
sudo -E bash -
```

```
> sudo apt install nodejs
```

```
> sudo apt install npm
```

```
> node -v
```

```
> npm -v
```

# 1.2. Software necessari.

## Possibles errors

```
> php artisan config:cache
```

```
> php artisan migrate
```



# 1.3. Primers passos amb Laravel

## Creació del projecte:

```
> laravel new nombre_proyecto
```

```
> laravel new biblioteca
```

## Fent ús de **composer**:

```
> composer create-project --prefer-dist laravel/laravel  
biblioteca
```

# 1.3. Primers passos amb Laravel

**Artisan:** és una interfície de la línia de comandaments que ens permetrà configurar i gestionar els nostres projectes Laravel. (Ver > Terminal)

```
> php artisan list
```

```
> php artisan --version
```

# 1.3. Primers passos amb Laravel

## Estructura dels projectes

**app:** conté el **codi font** de l'aplicació. Inicialment, s'inclouen algunes subcarpetes dins:

- **Console:** per definir els nostres propis comandos.
- **Exceptions:** per definir les nostres pròpies excepcions.
- **Http:** conté els **controladors** i el **middleware**.
- **Models:** emmagatzema els models de dades de l'aplicació (classes que s'utilitzaran per gestionar els objectes que intervenen en el sistema).
- **Providers:** conté els proveïdors de serveis de l'aplicació, més els que puguem definir nosaltres.

# 1.3. Primers passos amb Laravel

## Estructura dels projectes

**bootstrap:** conté l'arxiu **app.php**, que és el que posa en marxa l'aplicació. A més, conté la carpeta **cache**, on s'emmagatzemen els arxius ja carregats per Laravel per accelerar-ne l'accés en futures peticions.

**config:** conté els arxius de configuració de l'aplicació, on es tenen variables d'entorn, o si la nostra aplicació està en desenvolupament o producció, o els paràmetres de connexió a la base de dades, entre altres coses.

**database:** emmagatzema els elements de gestió de la base de dades, com ara generadors d'objectes, migracions, etc.

**public:** contingut visible de la web. Conté l'arxiu **index.php**, punt d'entrada de totes les peticions a la web. A més, podem definir o generar carpetes per ubicar-hi el contingut estàtic del client (imatges, fulls d'estil CSS, arxius JavaScript...).

# 1.3. Primers passos amb Laravel

## Estructura dels projectes

**resources:** conté, d'una banda, les **vistes** de la nostra aplicació. D'altra banda, també conté arxius no compilats de **CSS** (arxius **sass**) i **JavaScript** (arxius sense minimitzar ni optimitzar). A més, també emmagatzema els arxius de traducció, en cas que vulguem fer llocs multiidioma.

**routes:** emmagatzema les **rutes** de l'aplicació, tant per accedir a contingut web normal (**web.php**), com per a serveis web (**api.php**), com per a comandos i altres opcions.

**storage:** conté les vistes compilades, i altres arxius generats per Laravel, com els logs o les sessions.

**test:** per emmagatzemar els tests o proves sobre els components de la nostra aplicació.

**vendor:** on s'emmagatzemen les dependències o llibreries addicionals que es requereixen en el nostre projecte Laravel. Aquesta carpeta hauria de ser ignorada per Git i regenerada cada vegada que es clone el repositori remot, per evitar emmagatzemar massa informació innecessària.

# 1.3. Primers passos amb Laravel

## Configuració general del projecte

L'arxiu de configuració del projecte és .env. En aquest podrem indicar nom de l'aplicació, configurar la connexió a la base de dades, el servidor de correu, ...

```
APP_NAME=Biblioteca
...
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=biblioteca
DB_USERNAME=root
DB_PASSWORD=
...
```

# 1.3. Primers passos amb Laravel

## Arquitectura d'un projecte Laravel

**Proveïdors de serveis:** s'encarreguen d'arrancar el projecte amb classes, mètodes, dependències externes... Per exemple, connexió a la base de dades o autenticació.

**Classes del nucli:** Model, Route, ... Pertanyen a l'espai de noms **Illuminate**

```
use Illuminate\Database\Eloquent\Model;
```

**Altres elements:**

**Contracts i Facades:** proporcionen interfícies estàtiques a mètodes que poden ser útils

```
return Cache::get('key');
```

# 1.3. Primers passos amb Laravel

## Prova de projectes

1. Posar en la carpeta que volem o copiar la carpeta dins de `/htdocs`
2. Donar permisos 777 a les carpetes `bootstrap/cache` i `storage`
3. Posar en marxa amb `php artisan serve`



# 1.3. Primers passos amb Laravel

## Importació i exportació

1. Copia de l'arxiu `.env` i configuració
2. `php artisan key:generate`
3. `composer install`
4. `npm install`

# Exercici

## **Creació del projecte blog**

Utilitza el comandament laravel per crear un nou projecte amb el nom blog. Posteriorment posa-lo en marxa i comprova que pots accedir a ell.

# Atribuciones

[Curs de Nacho Iborra](#)