

# 1- Laravel: Framework pel desenvolupament d'aplicacions web utilitzant el patró MVC amb el llenguatge PHP

## 1.1- Framework Laravel: Què és?. Avantatges.

a) Què és Laravel?:

- El patró MVC es pot implementar manualment des de zero però cal estar molt atent i comprovar contínuament que l'estem utilitzant. És fàcil en algun moment apartar-se del patró amb els problemes que això comporta. Afortunadament existeixen eines per assegurar-nos que farem servir aquest patró sempre. Per PHP trobem una sèrie de frameworks que ens permeten realitzar aplicacions basades en aquest patró. Entre d'altres tenim [Laravel](#) i Zend.
- El framework de Zend és molt potent però una part important és de pagament. Laravel d'altra banda no és de pagament, té el suport d'una potent comunitat que manté actualitzat al dia el framework, té una documentació completa i és de codi obert ([llicència MIT](#)).

b) Les característiques de Laravel segons els seus desenvolupadors són:

- Sistema d'encaminament o routing (traducció d'URL externes a internes) fàcil de crear i mantenir.
- Sistema d'abstracció de bases de dades potent i senzill. Permet tractar les dades de les bases de dades com simples objectes.
- Sistemes de notificació de bases de dades
- Abstracció del sistema d'arxius que permet escriure dades en arxius que estan al cloud, al disc dur del servidor amb el mateix codi.
- Gestió de sessions
- Sistemes d'autenticació, amb el tot el necessari per treballar amb claus, logs d'inicis de sessió, etc...
- Sistemes de notificació a usuaris per exemple via correu electrònic
- Sistemes per incrementar el rendiment de les aplicacions.
- Facilitat d'ús.
- Ampla documentació

Els conceptes de Laravel que treballarem dins d'aquest curs són els següents:

- Namespace
- Models i Migrations
- Controllers
- Views
- Blades i Blade Templates
- Routing
- Request & Responses
- Middleware
- Seeders i Factories

## 1.2- Creació d'un projecte PHP amb el framework Laravel

a) Crea dins del teu sistema **físic** una carpeta de nom **projectes**. Accedeix a **projectes** i dins de la carpeta crea una nova carpeta de nom **empresa** i dins d'aquesta carpeta crea dues de nom **codis** i **scripts**. A continuació, copia el fitxer **Vagrantfile** d'aquesta sessió dins de la carpeta **empresa** i el script SQL d'aquesta sessió **empresa.sql** dins de la carpeta **scripts**.

b) Crea i inicia una nova màquina virtual amb el fitxer **Vagrantfile**.

c) Accedeix a la màquina virtual, i amb **composer** instal·larem una eina del sistema de nom **laravel** (*Laravel installer*) que utilitzarem per fàcilment crear un nou projecte amb el framework complet de **Laravel**.

Per instal·lar aquesta eina dins del sistema executa des del directori personal de l'usuari **vagrant** (/home/vagrant que es veu com ~) l'ordre:

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

d) Un cop instal·lada l'eina **laravel**, modifica el **PATH** de l'usuari vagrant. Dins del directori personal de l'usuari **vagrant** (/home/vagrant que es veu com ~) Executa:

```
echo "export PATH=$PATH:~/.config/composer/vendor/laravel/installer/bin" >> ~/.bashrc
exec bash
```

e) Comprova que si executes: `echo $PATH =>` Dins del nou **PATH** s'ha afegit el directori **/home/vagrant/.config/composer/vendor/laravel/installer/bin** que és a on s'instal·la l'eina **laravel**.

f) Executa: `laravel --version =>` Comprova que la versió actual instal·lada és: **Laravel Installer 5.24.6** o posterior.

g) Dins del directori **/var/www/html** de la teva màquina virtual crearem un projecte amb el framework de **Laravel** incorporat de nom **empresa**. Pots crear el projecte de 2 maneres:

- Utilitzant l'eina **laravel** → `laravel new empresa`
- Utilitzant **composer** → `composer create-project laravel/laravel --prefer-dist empresa`

Farem servir el primer cas perquè generalment és més senzill de treballar amb aquesta opció.

**IMPORTANT: NO UTILITZIS `sudo` PERQUÈ `vagrant` JA TÉ ELS PERMÍSOS `rwX` SOBRE EL DIRECTORI `/var/www/html` !!!!!!!!!**

A les preguntes que presenta el procés d'instal·lació contesta:

- Indica que vols començar sense cap starter kit (que sempre es pot instal·lar posteriorment).
- Selecciona l'**opció per defecte** pel **framework de testing**.
- **No** instal·lis **Laravel Boost** (ho farem més endavant si ens interessa).
- El tipus de base de dades serà **MySQL**.
- Indica que **No** vols executar les **migrations** de la base de dades per defecte (ho farem més endavant manualment).
- Indica que **No** vols executar **npm install** i **npm run build**.

h) Comprova que s'ha creat el directori **/var/www/html/empresa** i que dins de la carpeta s'ha incorporat el framework **Laravel**.

i) Comprova que dins del directori **/home/vagrant/scripts** tens un script SQL de nom **empresa.sql** amb aquest codi:

```
use mysql;
alter user 'root'@'localhost' identified by 'FjeClot26@';
create user 'administrador'@'localhost' identified by 'FjeClot26#';
create database empresa;
use empresa;
grant all privileges on empresa.* to 'administrador'@'localhost';
```

Amb aquest script:

- Es crea la base de dades **empresa**
- Es crea l'usuari **administrador** amb tots els privilegis sobre totes les taules de la base de dades **empresa**.
- S'assigna una contrasenya a l'usuari **root** del servidor MySQL.

j) Des del directori **/home/vagrant/scripts** i amb el terminal accedeix al gestor de **MariaDB** i afegeix l'usuari i la base de dades. Executa des del terminal del sistema:

**sudo mysql**

i després d'escriure la contrasenya de l'usuari **root** del servidor **mysql**, des de dins del gestor **MariaDB**, executa:

**source ./empresa.sql**

k) Surt del terminal de gestió de **MySQL** i accedeix al directori **/var/www/html/empresa**. Edita el fitxer **.env** del projecte per configurar la connexió del projecte amb el servidor. Modifica les línies **23** a **28** del fitxer amb les següents dades de connexió:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=empresa
DB_USERNAME=administrador
DB_PASSWORD="FjeClot26#"
```

l) Executa l'ordre necessària per **crear les taules bàsiques inicials i necessàries** de la base de dades de l'aplicació. Executa dins de la carpeta **empresa**:

**php artisan migrate**

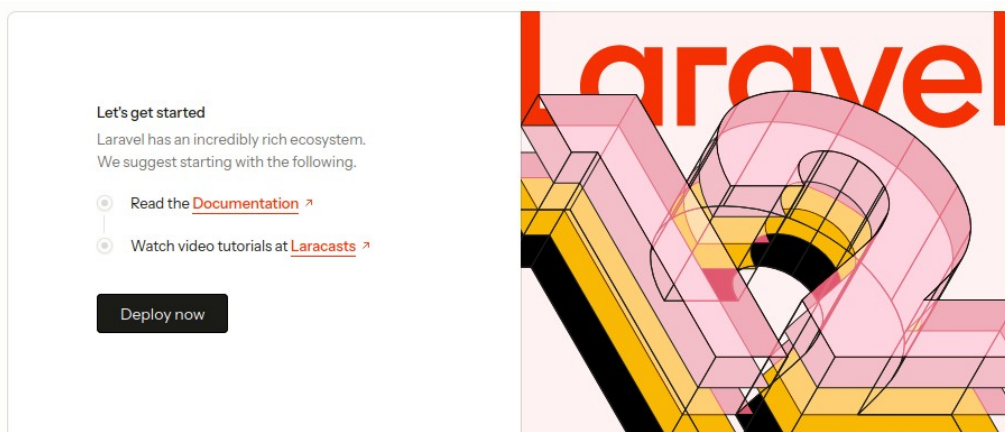
m) Troba l'adreça IP de la màquina virtual executant: **ip a**

n) Ara des de dins del directori **empresa** iniciem el servidor web intern propi del framework Laravel executant:

**php artisan serve --host=0.0.0.0 --port=8000**

i veurem que es posa en marxa el servidor escoltant pel **port 8000** i acceptant connexions des de l'adreça IP **0.0.0.0** (això vol dir que accepta connexions des de qualsevol IP).

o) Finalment, amb el navegador de la màquina **host** comprova que pots accedir a l'aplicació web de la màquina virtual, utilitzant l'adreça IP de la màquina virtual i el port **8000**:



Si això funciona, vol dir que podem establir una connexió amb el servidor i ara ja es poden començar a desenvolupar les diverses parts del projecte.

o) Pots aturar el servidor web simplement amb un **Ctrl + C**

p) Modifica **routes** → **web.php** i comenta la línia **6** i afegeix una nova línia a continuació amb aquest codi:

```
return view('inici');
```

q) Crea un fitxer de nom **inici.blade.php** dins de **resources** → **views**.

**r)** Afegeix el següent codi font a **inici.blade.php**:

```
<strong>Database Connected: </strong>
<?php
    try {
        \DB::connection()->getPDO();
        echo \DB::connection()->getDatabaseName();
    } catch (\Exception $e) {
        echo 'No hi ha connexió';
    }
?>
```

**t)** Comprova que si fem una connexió a l'aplicació ara sortirà una pàgina indicant-nos la base de dades amb la qual ens hem connectat.