LARAVEL

Es un framework desarrollado sb php.

Es una aplicación.

Se lo puede llamar tb librería. Xq hace uso de muchas librerías.

Para crear un proyecto de laravel, escribir en la consola:

**composer create-project laravel/laravel nombreDelProyecto**

PETICIONES: puede ser GET o POST

Crearemos rutas y controladores y ya no tantas carpetas

NAMESPACES es como un alias.

Usaremos USE en lugar de INCLUDE.

DIRECTORIOS MÁS IMPORTANTES:

* En la carpeta APP:

En CONTROLLERS va a estar nuestro trabajo de desarrollo.

* PUBLIC. Carpeta accesible desde el navegador. Si el archivo existe en esta carpeta, puedo acceder, x ej los CSS
* ROUTES: toda petición que viene del navegador pasa por aquí.

Debemos definir las peticiones, aparte de GET y POST están:

* PUT/PATCH ( ACTUALIZA DATOS)
* DELETE

Route::get(‘home’, function (){

Return ‘Hello World’;};

Lo que no esté escrito aquí, me arrojará el cartel 404

Utilizando las **llaves** defino una **variable**.

Route::get(‘home/{name}’, function ($name){

Return ‘Hello’. $name;

};

Si coloco el signo de interrogación,…

Con el arroba apunto a la función, pero previo estará el controlador.

CONFIG

Las URL las definimos así:

# EJERCICIOS:

**PARA SABER SI ES PAR O IMPAR:**

Route::get('/resultado/{numero}', function ($numero){

echo ($numero % 2 == 0) ? "Es par" : "Es impar";

});

**EJ. 3 PARA INGRESAR UN OPCIONAL**

Route::get('/resultado/{numero}/{opcional}', function ($numero, $opcional){

return $numero \* $opcional;

# así me funcionó, pero agregando un cero después de la barra.

Route::get('/resultado/{numero}/{opcional}', function ($numero, $opcional){

if ($numero and $opcional) {

return $numero \* $opcional;

}

else {

if (empty ($opcional)) {

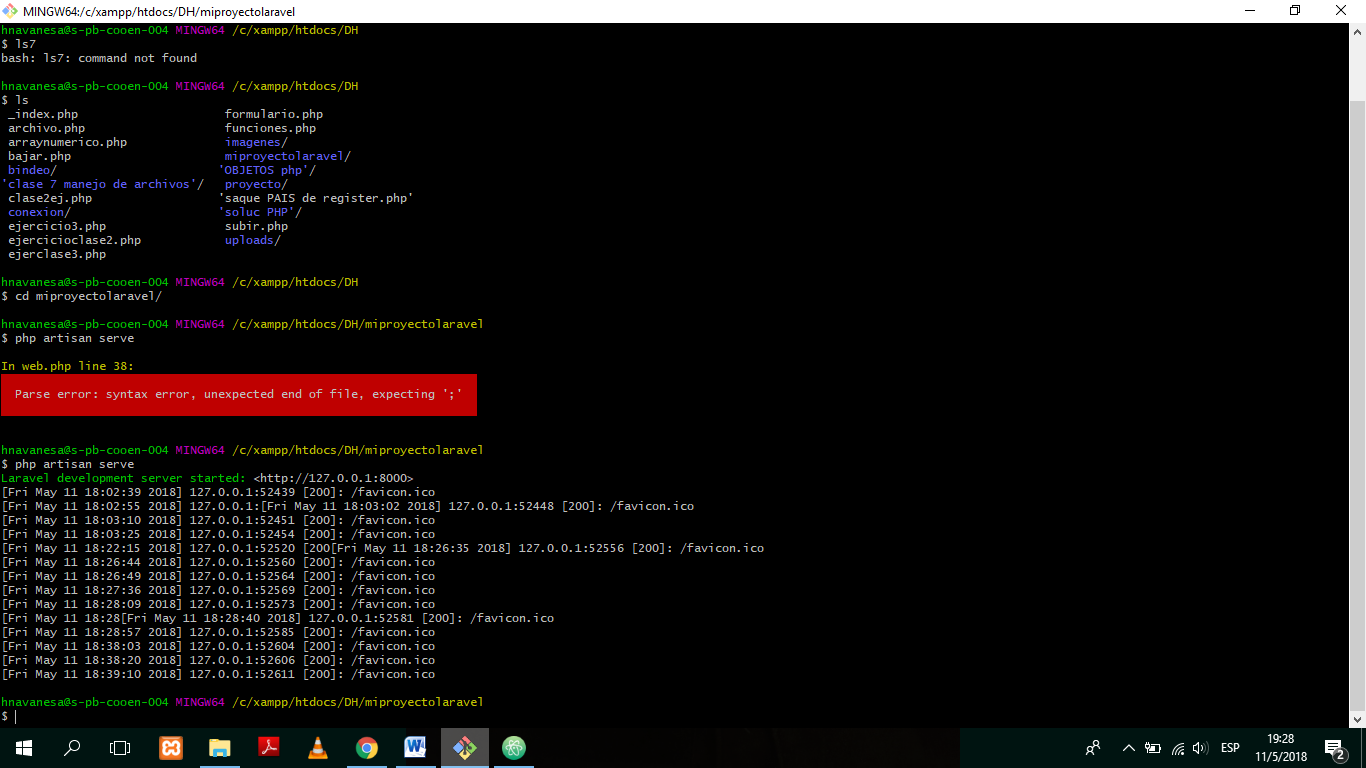
return ($numero % 2 == 0) ? "Es par" : "Es impar";

}

}

});

## en la CONSOLA, para poder ingresar:



VIERNES 11 de may. de 18

## Patrón MVC

Componentes:

1. Controlador
2. modelos
3. vistas,

Las VIEW las tenemos DENTRO DE resources Y VAN dentro de la carpeta VIEW …

Se guardan … .**BLADE.PHP**

## STARK OVERFLOW!!!! Es la web que te da muchas soluciones

## CONTROLLERS

 Su función es la de definir el código a ejecutar como comportamiento frente a una acción solicitada dentro de la aplicación.

Crear un controlador voy a la consola y allí escribo:

php artisan make:controller

por ejemplo: crear el controlador PELICULAS CONTROLLER, escribo:

php artisan make:controller PeliculasController

**APP luego HTTP luego CONTROLLERS**

## NAMESPACE

Con la primera parte digo …

Luego “use”

## COMANDOS BLADE

Usa el@

*Directivas:*

Para que te tire las vistas.

{{….}} es un “Echo” o helper

## Proceso para crear:

1.

2.

3.

@ yield

Funciona para…

Para llamar a mi YIELD escribo @ section ( ) y no debo olvidarme de cerrarlo con @endsection.