

Abstract geometric shapes in the top-left corner, including a green parallelogram, a light blue parallelogram, a brown parallelogram, and a large purple parallelogram.

LARAVEL

CLASE 01

Abstract geometric shapes in the top-right corner, including a green parallelogram, a light blue parallelogram, a purple parallelogram, and a red parallelogram.



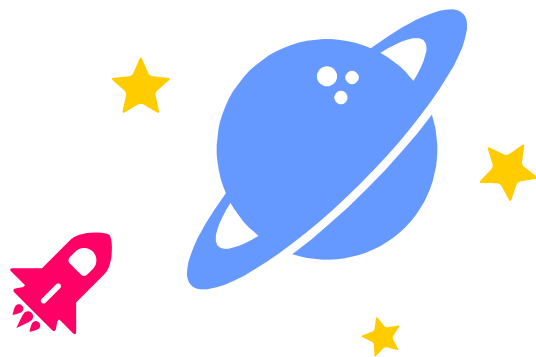
FRAMEWORK

Un framework nos provee un entorno de desarrollo con el cual podemos resolver problemas comunes más rápidamente.

¿Por qué usar un framework?

- › Nos permite organizar el código y nos brinda un sistema de archivos
- › Nos permite desarrollar más rápido y escribir menos código
- › Nos permite trabajar en equipo fácilmente utilizando los mismos criterios
- › Nos permite tener un lenguaje en común independiente del desarrollador





BIBLIOTECA

Una biblioteca (o librería) nos provee código para resolver un problema específico sin necesidad de mucho contexto.

COMPOSER

<https://getcomposer.org>

COMPOSER es un **gestor de dependencias** para proyectos **programados en PHP**. Esto quiere decir que nos permite instalar y mantener actualizados los paquetes de software necesarios para nuestro proyecto.



Accede a la **GUÍA DE INSTALACIÓN** de Composer:



- › **Instalación en LINUX**
- › **Instalación en MAC**
- › **Instalación en WINDOWS**

LARAVEL

<https://laravel.com>

LARAVEL es un `framework` de código abierto para desarrollar aplicaciones y servicios web de un modo mucho más rápido, fácil, y seguro.

Para poder comenzar a trabajar, es necesario crear un proyecto nuevo a través de composer.

INSTALACIÓN de Laravel:



- **Instalación en LINUX / MAC / WINDOWS**

**ES MOMENTO DE
PRACTICAR !**



FILE SYSTEM

PHP

Para cada sitio de nuestra web solemos crear un archivo nuevo, es decir, otro punto de entrada.

LARAVEL

Con Laravel esto ya no es así, contamos con un punto de entrada único a la aplicación y distintas Clases que toman el pedido según la ruta.



FILE SYSTEM

Es decir, debemos quitar la idea de que podemos identificar el archivo que controla el sitio a través de lo que vemos en la URL y en el file system.





ESTRUCTURA DE DIRECTORIOS

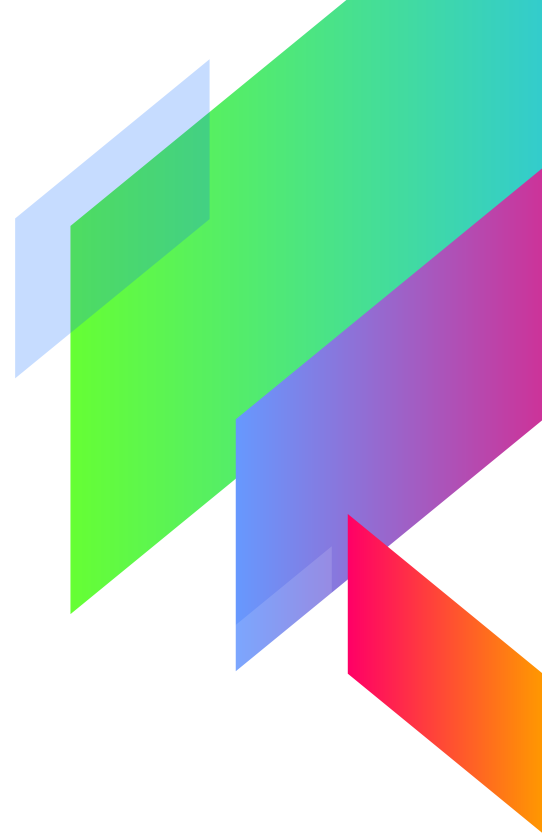
Conozcamos el la estructura de
directorios de Laravel





ROUTING

Conozcamos el sistema
de ruteo de Laravel



METODOS HTTP

MÉTODO

FUNCIÓN

GET

Se utiliza para **consultar** elementos en la base de datos.

POST

Se utiliza para **crear** elementos en la base de datos.

PUT / PATCH

Se utiliza para **modificar** elementos en la base de datos.

DELETE

Se utiliza para **eliminar** elementos en la base de datos.

```
> routes/web.php
```

```
<?php
```

```
/*  
|-----  
| Rutas SIMPLES  
|-----  
|  
| Here is where you can register all of the routes for an application.  
| It's a breeze. Simply tell Laravel the URIs it should respond to  
| and give it the controller to call when that URI is requested.  
|  
*/
```

```
Route::get('home', function () {  
  
    return 'Hello World';  
  
});
```

```
<?php
```

```
/*  
|-----  
| Rutas con PARAMETROS  
|-----  
|  
| Here is where you can register all of the routes for an application.  
| It's a breeze. Simply tell Laravel the URIs it should respond to  
| and give it the controller to call when that URI is requested.  
|  
*/
```

```
Route::get('home/{name}', function ($name) {  
    return 'Hello ' . $name;  
});
```



```
<?php
```

```
/*
```

```
-----  
Rutas con PARÁMETROS OPCIONALES  
-----
```

```
Here is where you can register all of the routes for an application.  
It's a breeze. Simply tell Laravel the URIs it should respond to  
and give it the controller to call when that URI is requested.
```

```
*/
```

```
Route::get('post/{post_id}/comments/{comment_id?}', function ($post_id, $comment_id =  
null) {
```

```
    if ($comment_id !== null) {  
        return $comment_id;  
    } else {  
        return 'No hay comentarios';  
    }  
}
```

```
});
```

> routes/web.php

```
<?php
```

```
/*
```

```
-----  
| Rutas Hacia CONTROLADORES  
|-----  
|
```

```
| Here is where you can register all of the routes for an application.  
| It's a breeze. Simply tell Laravel the URIs it should respond to  
| and give it the controller to call when that URI is requested.  
|
```

```
*/
```

```
Route::get('/peliculas', 'PeliculasController@listado');    // Consulta (lista)
```

```
Route::get('/peliculas/{id}', 'PeliculasController@ver');    // Consulta (item)
```

```
Route::post('/actores', 'ActoresController@crear');    // Alta
```

```
Route::patch('/peliculas/{id}', 'PeliculasController@editar');    // Modificación
```

```
Route::delete('/actores/{id}', 'ActoresController@eliminar');    // Baja
```

**ES MOMENTO DE
PRACTICAR !**



The image features abstract geometric shapes in the corners. On the left, there are overlapping shapes in shades of green, blue, orange, and purple. On the right, there are overlapping shapes in shades of green, blue, purple, and orange. The central text is in a bold, black, sans-serif font.

**¡ HASTA LA
PROXIMA !**