

```
blic function update(Request $request, Receta $receta)
```

```
// Revisar el policy  
$this->authorize('update', $receta);
```

```
// validación  
$data = $request->validate([  
    'titulo' => 'required|min:6',  
    'preparacion' => 'required',  
    'ingredientes' => 'required'
```

MVC

Model View Controller

```
$img->save();
```

```
// Asignar al objeto  
$receta->imagen = $ruta_imagen;
```

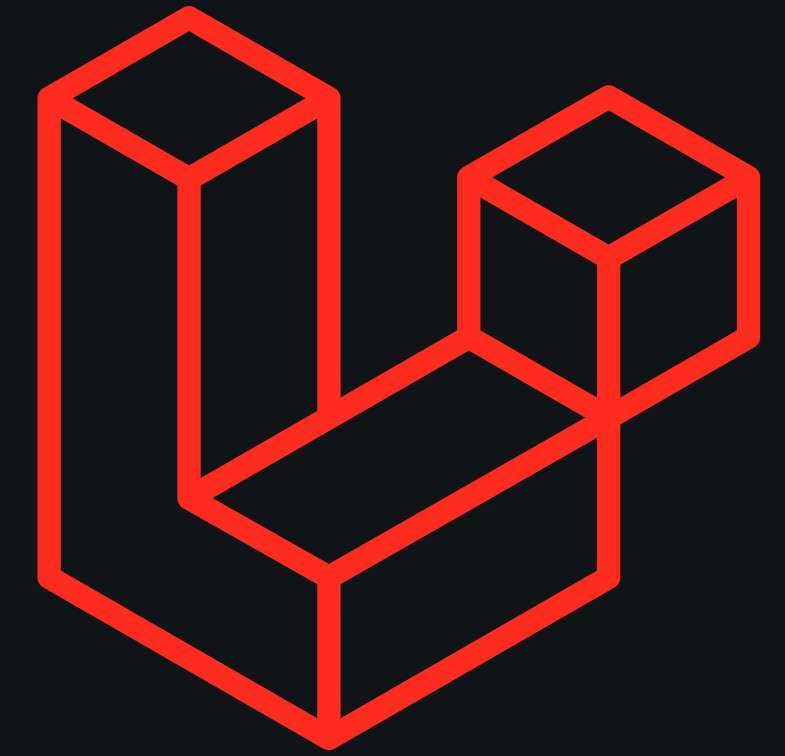
Juan Pablo De la torre Valdez

www.codigoconjuan.com

CÓDIGO
CON
JUAN

```
// redireccionar  
return redirect()->action('RecetaController@index');
```

¿Qué es MVC?

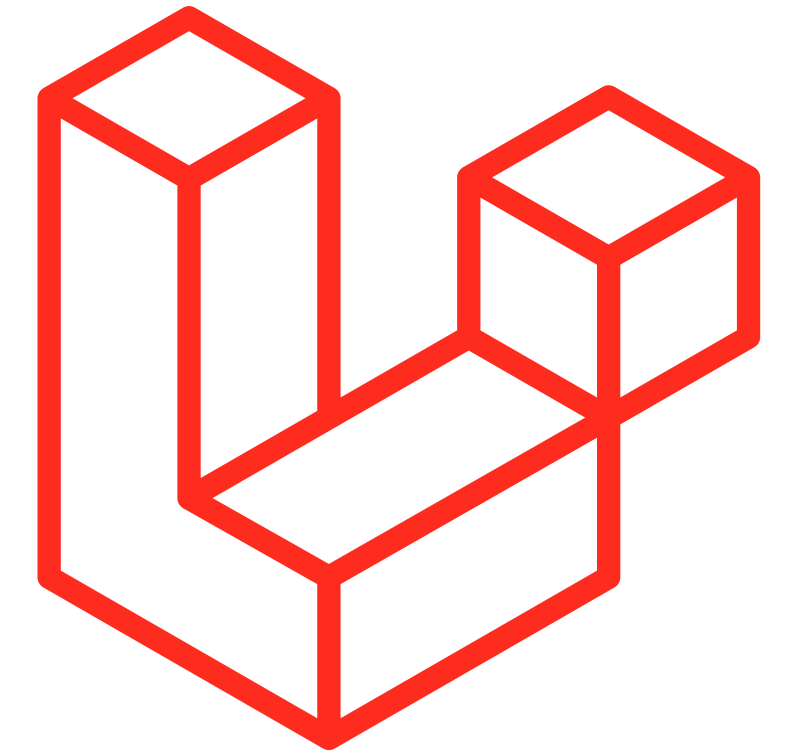


MVC son las iniciales de Model View Controller

Es un patrón de arquitectura de software que permite la separación de obligaciones de cada pieza de tu código.

Enfatiza la separación de la lógica de programación con la presentación

Ventajas de MVC



MVC no mejora el performance del código, tampoco da seguridad; **pero tu código tendrá un mejor orden y será fácil de mantener.**

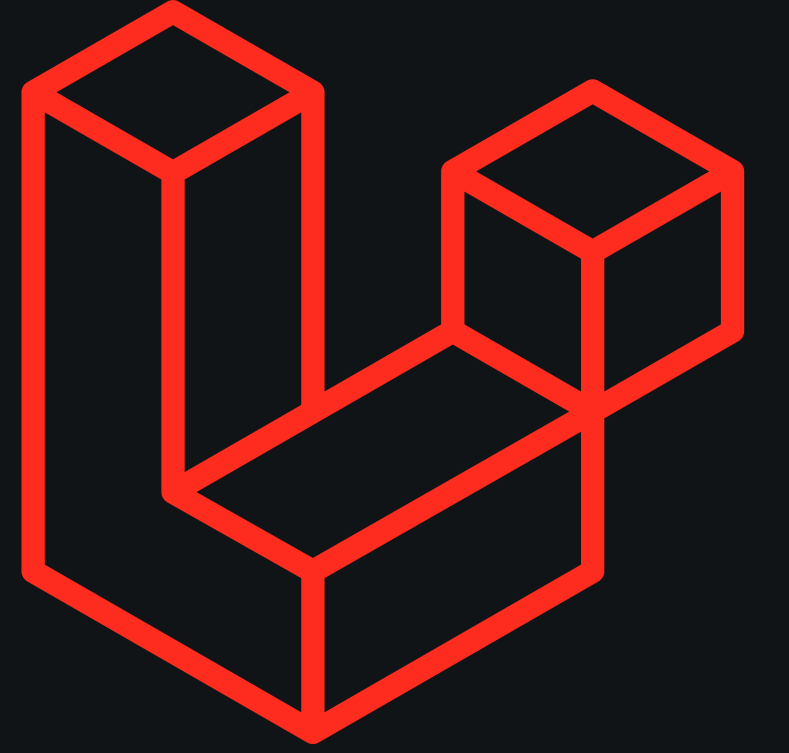
En un grupo de trabajo, el tener el código ordenado permite que más de una persona pueda entender que es lo que hace cada parte de él.

Aprender MVC, te hará que otras tecnologías como Nest, Rails, Django, Net Core, Spring Boot te serán más sencillas de aprender.



Piezas de

MVC



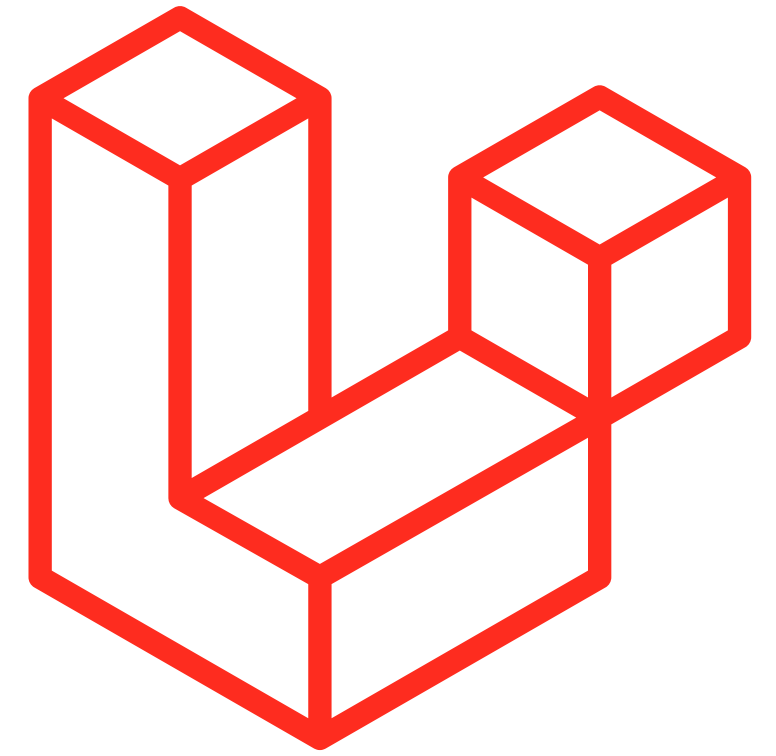
M = Model o Modelo

V = View o Vista

C = Controller o Controlador

Modelo en

MVC

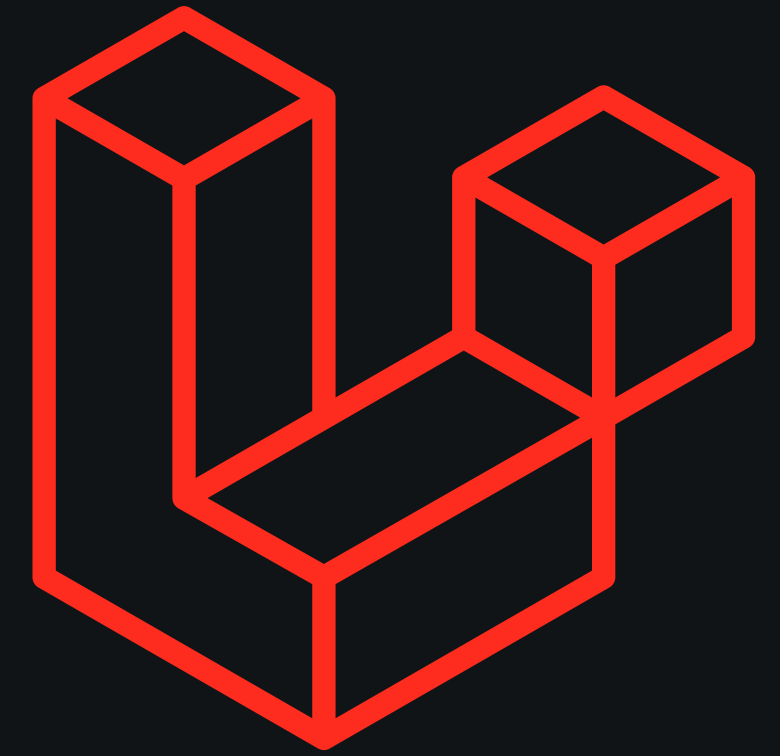


Encargado de todas las interacciones en la base de datos, obtener datos, actualizarlos y eliminar.

El Modelo se encarga de consultar una base de datos, obtiene la información pero no la muestra, eso es trabajo de la vista.

El Modelo tampoco se encarga de actualizar la información directamente; es el Controlador quien decide cuándo llamarlo.

Vista en MVC



Se encarga de todo lo que se ve en pantalla (HTML)

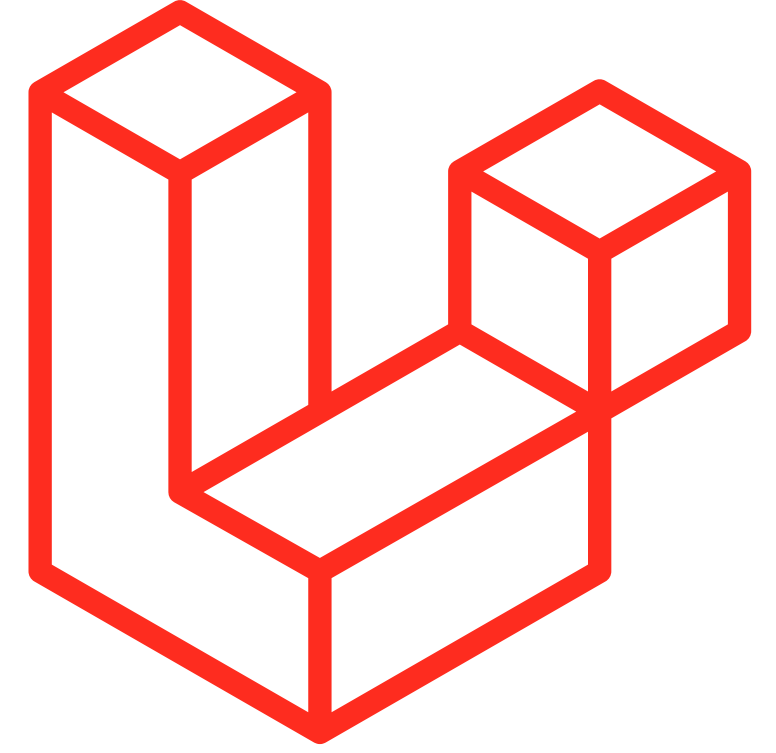
Laravel tiene un Template Engine llamado Blade para mostrar los datos.

Si utilizas React, Vue, Angular, Svelte, estos serían tu vista.

El Modelo consulta la base de datos, pero es por medio del Controlador que se decide que Vista hay que llamar y que datos presentar.

Controlador en

MVC



Es el que comunica modelo y vista; antes de que el Modelo consulte la base de datos el Controlador es el encargado de llamar un Modelo en específico.

Una vez consultado el Modelo, el controlador recibe esa información, manda llamar a la vista y le pasa la información.

El controlador es el que mandar llamar la vista y modelos, que se requieren en cada parte de tu aplicación.

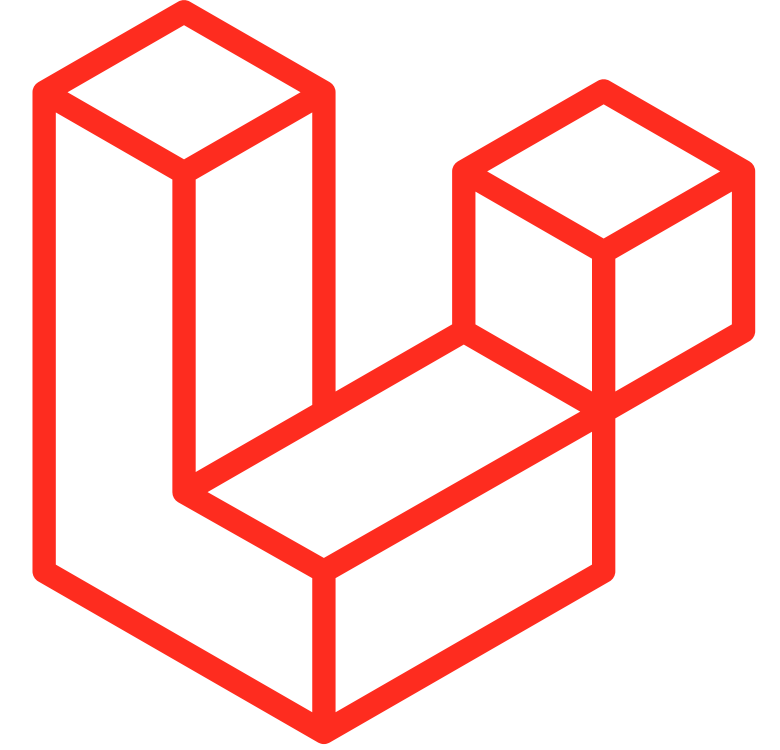
Router en MVC



Es el encargado de registrar todas las URL's o Endpoints que va a soportar nuestra aplicación.

Ejemplo: Si el Usuario accede a /clientes el router ya tiene registrada esa ruta y un controlador con una función que sabe que Modelo debe llamar y que vista mostrar cuando el usuario visita esa URL.

/clientes



controlador

modelo

vista