**Identificación:** Anexo05-Controladores  
**Versión:** 1.0

Tabla de contenido

[Controladores – Laravel 2](#_Toc63108883)

[Patron MVC 2](#_Toc63108884)

[Modelo 2](#_Toc63108885)

[Vista 2](#_Toc63108886)

[Controlador 2](#_Toc63108887)

[MVC en Laravel 3](#_Toc63108888)

[Controladores con Recursos 3](#_Toc63108889)

[Taller 7](#_Toc63108890)

[Bibliografia 7](#_Toc63108891)

# **Controladores – Laravel**

## Patron MVC

MVC (modelo, vista, controlador) es un patrón arquitectónico de software que separa una aplicación en tres capas descritas como su acrónimo lo indica. Laravel, así como la mayoría de frameworks en PHP implementan este patrón de diseño en donde cada capa maneja un aspecto de la aplicación. Pero antes de ver cómo Laravel está diseñado para implementar este patrón de software, vamos a tratar de dejar este concepto un poco más claro definiendo cada una de sus partes.

### Modelo

Hace referencia a la estructura de datos de la aplicación. Los datos pueden ser transferidos desde la base de datos, una clase, un servicio, u otros, directamente a la vista o ser transformados en el controlador para ser actualizados nuevamente al origen.

### Vista

Es la representación de la información en una interfaz de usuario. Por lo general en interfaces no estáticas se representan los datos que vienen directamente del modelo o estos son transformados en un proceso intermedio en el controlador. En vistas estáticas por lo general no hace falta que las vistas sean renderizadas con datos enviados del controlador.

### Controlador

Es el lugar en donde se implementa la lógica de la aplicación, los procedimientos, algoritmos y rutinas que hacen que funcione el software. Actúa como interfaz entre los componentes de modelo y vista aplicando las transformaciones y lógica necesarias.

Diagrama

Descripción generada automáticamente

# MVC en Laravel

En laravel se puede crear un controlador de forma rápida y sencilla

php artisan make:controller <nombreControlador>

Ejemplo:

php artisan make:controller nuevo

<?php

namespace App\\Http\\Controllers;

use Illuminate\\Http\\Request;

class nuevo extends Controller

{

//

}

## Controladores con Recursos

El enrutamiento de recursos de Laravel asigna las típicas rutas "CRUD" a un controlador con una sola línea de código. Por ejemplo, es posible que desee crear un controlador que maneje todas las solicitudes HTTP de "fotos" almacenadas por su aplicación. Usando el comando make: controller Artisan, podemos crear rápidamente dicho controlador:

php artisan make:controller PhotoController –resource

php artisan make:controller PhotoController –r

<?php

namespace App\\Http\\Controllers;

use Illuminate\\Http\\Request;

class PhotoController extends Controller

{

/\*\*

\* Display a listing of the resource.

\*

\* @return \\Illuminate\\Http\\Response

\*/

public function index()

{

//

}

/\*\*

\* Show the form for creating a new resource.

\*

\* @return \\Illuminate\\Http\\Response

\*/

public function create()

{

//

}

/\*\*

\* Store a newly created resource in storage.

\*

\* @param \\Illuminate\\Http\\Request $request

\* @return \\Illuminate\\Http\\Response

\*/

public function store(Request $request)

{

//

}

/\*\*

\* Display the specified resource.

\*

\* @param int $id

\* @return \\Illuminate\\Http\\Response

\*/

public function show($id)

{

//

}

/\*\*

\* Show the form for editing the specified resource.

\*

\* @param int $id

\* @return \\Illuminate\\Http\\Response

\*/

public function edit($id)

{

//

}

/\*\*

\* Update the specified resource in storage.

\*

\* @param \\Illuminate\\Http\\Request $request

\* @param int $id

\* @return \\Illuminate\\Http\\Response

\*/

public function update(Request $request, $id)

{

//

}

/\*\*

\* Remove the specified resource from storage.

\*

\* @param int $id

\* @return \\Illuminate\\Http\\Response

\*/

public function destroy($id)

{

//

}

}

**Especificar el modelo de recursos**

Si está utilizando el enlace del modelo de ruta y le gustaría que los métodos del controlador de recursos escribieran una sugerencia de una instancia de modelo, puede usar la opción --model al generar el controlador:

php artisan make:controller PhotoController --resource --model=Photo

**Pasos a seguir**

C:\\Users\\wramirez\\Documents\\projects\\laravel8>php artisan make:controller PhotoController --resource --model=Photo

A App\\Models\\Photo model does not exist. Do you want to generate it? (yes/no) [yes]:

> yes

**Resultado**

<?php

namespace App\\Http\\Controllers;

use App\\Models\\Photo;

use Illuminate\\Http\\Request;

class PhotoController extends Controller

{

/\*\*

\* Display a listing of the resource.

\*

\* @return \\Illuminate\\Http\\Response

\*/

public function index()

{

//

}

/\*\*

\* Show the form for creating a new resource.

\*

\* @return \\Illuminate\\Http\\Response

\*/

public function create()

{

//

}

/\*\*

\* Store a newly created resource in storage.

\*

\* @param \\Illuminate\\Http\\Request $request

\* @return \\Illuminate\\Http\\Response

\*/

public function store(Request $request)

{

//

}

/\*\*

\* Display the specified resource.

\*

\* @param \\App\\Models\\Photo $photo

\* @return \\Illuminate\\Http\\Response

\*/

public function show(Photo $photo)

{

//

}

/\*\*

\* Show the form for editing the specified resource.

\*

\* @param \\App\\Models\\Photo $photo

\* @return \\Illuminate\\Http\\Response

\*/

public function edit(Photo $photo)

{

//

}

/\*\*

\* Update the specified resource in storage.

\*

\* @param \\Illuminate\\Http\\Request $request

\* @param \\App\\Models\\Photo $photo

\* @return \\Illuminate\\Http\\Response

\*/

public function update(Request $request, Photo $photo)

{

//

}

/\*\*

\* Remove the specified resource from storage.

\*

\* @param \\App\\Models\\Photo $photo

\* @return \\Illuminate\\Http\\Response

\*/

public function destroy(Photo $photo)

{

//

}

}

# **Taller**

En su proyecto actual van a crear un controlador por para todas las rutas, cada metodo del controlador va a responder a cada ruta para asi poder mostrar la vista correspondiente.

**Entregable:** Subir un documento en formato pdf que contenga el pantallazo de las 4 vistas renderizadas y el comportamiento del servidor que muestra en la consola cada ruta visita.

“Enlace denominado Controladores\_y\_Rutas”

Usar las vistas previamente desarrolladas en las actividades anteriores

# **Bibliografia**

https://laravel.com/docs/8.x