

Resumen laravel

Las peticiones llegan a /public/index.php (el único archivo php disponible al public (front-controller))

Kernel: Parte del programa que recibe la request y genera la response, después de esa response el front-controller se la manda al servidor web.

El kernel comprueba a través de la url si hay una ruta que encaja (mira las rutas) dichas rutas están en la carpeta routes/web.php

routes = verbo + url → clausura (función anónima) o acción de un controlador

Ejemplo atención de petición por clausura:

```
Route::get('/', function() {  
    return view('welcome')  
})
```

Aquí devuelve una vista de welcome

Atención de petición por controlador:

En esta petición lo que hace es que la respuesta la recibe el controlador y realiza la acción de editar

```
Route::middleware('auth')->group(function () {  
    Route::get('/profile', [ProfileController::class, 'edit'])->name('profile.edit');  
});
```

Dentro de las acciones de un controlador tenemos:

Datos (Modelos) (/app/models):

- QueryBuilder.
- Eloquent (nombre que le da laravel a un patrón que se llama Active Record): Vincula tablas con clases, columnas con propiedades de las clases y filas con instancias de las clases (ORM mapeado objeto-relacional)

Visualización (vistas) (/resources/views)

En laravel esto se hace usando las plantillas blade.

Estas hacen uso de una serie de marcadores con arroba como @if, @foreach, @guest, @auth también tiene el tema de la {{}} (doble llave) esto bien usado evita que tengas que abrir y cerrar las etiquetas php, salvo en casos puntuales.

Estas van dentro de los componentes que se encuentran en /resources/views/components

Componentes (/resources/views/components):

Son vistas parciales, con estas puedes construir interfaces de usuario componiendo componentes unos con otros. En resumen, son trocitos de paginas que puedes usar en cualquier lado, porque están pensados para su reutilización.

Hay ciertos componentes especiales, que son los layouts (/resources/views/layouts): Contiene el esqueleto básico de la pagina, en medio es donde se renderiza las vistas.

Las tablas se crean con migraciones (como una maquina del tiempo que permite ir hacia cualquier punto que hayas guardado de la base de datos)

Ejemplo:

tabla 'articulos' → modelo 'articulo' (nombre completo de la clase (\App\Models\Articulo))

articulo::all → Devuelve todos los articulos

articulo::find(5) → Devuelve el articulo 5

<x-app-layout>

x → es el componentes

app → es el nombre

layout → es la carpeta donde busca

Livewire: Permite escribir aplicaciones asíncrona sin tener que echar mucha mano a javascript.