# Vistas de administración

## Introducción

En esta sección veremos cómo implementar las vistas, controladores y rutas para mantener nuestras **categorías, subcategorías y usuarios**. Al final del capítulo, también estudiaremos cómo crear un sistema de roles y permisos para nuestros **usuarios**.

## Categorías

A continuación, vamos a crear la vista, controlador y ruta necesarios para implementar nuestro mantenimiento de **categorías**.

### Creación del controlador

Todos los controladores para la administración los crearemos en una carpeta llamada **Admin** dentro de la carpeta **app\* para mantener cierto orden y control. De esta manera, separaremos los controladores del** Frontend\*\* de los controladores del panel de administración.

Para crear el controlador, teclee el siguiente comando:

php artisan make:controller Admin\CategoriaController

Una vez creado el controlador, diríjase a él e introduzca el siguiente código:

public function index()  
{  
 $categorias = Categoria::paginate(4);  
  
 return view('admin.categorias.index', compact('categorias'));  
}

El método **paginate** indica a Eloquent que devuelva el resultado en páginas de 4 categorías por página. Volveremos a este tema en la implementación de la vista.

### Creación de la vista

Todas las vistas asociadas con nuestro **panel de administración** estarán contenidas en una carpeta denominada **admin** dentro de la carpeta \*\*resources\*. En el caso de las categorías, incluiremos todas las vistas asociadas en \*\*resources\*.

Empecemos por la vista **index.blade.php**. Esta vista nos mostrará un listado paginado con todas las categorías disponibles.

@extends('adminlte::page')  
  
@section('title', 'Categorías')  
  
@section('content\_header')  
 <h1>Lista de Categorías</h1>  
@stop  
  
@section('content')  
 {{-- Comprobación de sesión --}}  
 @if (session('success'))  
 <div class="alert alert-success alert-dismissible fade show" role="alert">  
 {{ session('success') }}  
 <button type="button" class="close" data-dismiss="alert" aria-label="Cerrar">  
 <span aria-hidden="true">&times;</span>  
 </button>  
 </div>  
 @endif  
  
 <div class="card col-lg-12 col-md-4">  
 <div class="card-header mb-4">  
 <table>  
 <td>  
 <a href="{{ route('admin.categoria.create') }}" class="btn btn-primary btn-sm">  
 Agregar categoría  
 </a>  
 </td>  
 <!-- El botón "Agregar categoría" redirige a la ruta "admin.categoria.create" -->  
 </table>  
 </div>  
 <div class="card-body mt-2">  
 <table class="table">  
 <thead>  
 <tr>  
 <th scope="col">#</th>  
 <th scope="col">Nombre</th>  
 <th scope="col">Descripción</th>  
 <th scope="col">Imagen</th>  
 <  
  
th scope="col" colspan="2">Acción</th>  
 </tr>  
 </thead>  
 <tbody>  
 {{-- Iteración a través de las categorías --}}  
 @foreach ($categorias as $categoria)  
 <tr>  
 <th scope="row">{{ $categoria->id }}</th>  
 <td><b>{{ $categoria->nombre }}</b></td>  
 <td>{{ strip\_tags($categoria->descripcion) }}</td>  
 <td><img src="{{ $categoria->imagen }}" width="125px"></td>  
 {{-- Botón de edición --}}  
 <td width="10px">  
 <a href="{{ route('admin.categoria.edit', ['id' => $categoria->id]) }}" class="btn btn-primary btn-sm">Editar</a>  
 </td>  
 {{-- Botón de borrado --}}  
 <td width="10px">  
 <form action="{{ route('admin.categoria.delete', ['id' => $categoria->id]) }}" method="POST">  
 @csrf  
 @method('DELETE')  
 <button type="submit" class="btn btn-danger btn-sm" onclick="return confirm('Si borra esta categoría, borrará todas las subcategorías y anuncios asociados.\n¿Estás seguro de que deseas eliminar esta categoría?')">Eliminar</button>  
 </form>  
 </td>  
 </tr>  
 @endforeach  
 </tbody>  
 </table>  
 </div>  
 </div>  
 {{ $categorias->links('pagination::bootstrap-4') }}  
@stop

:::info ¿Qué hace? Claro, aquí tienes una explicación fácil y sencilla de lo que hace la vista:

* @extends('adminlte::page'): Esta línea indica que la vista extiende el layout proporcionado por AdminLTE, lo cual significa que heredará estilos y funcionalidades predefinidas del tema.
* @section('title', 'Categorías'): Aquí se define el título de la página, que en este caso será “Categorías”.
* @section('content\_header'): Esta sección define el contenido que se mostrará en el encabezado de la página. En este caso, se muestra el título “Lista de Categorías”.
* @section('content'): Aquí comienza la sección principal de contenido de la página. A partir de este punto, se empieza a definir el contenido que se mostrará en el cuerpo de la página.
* @if (session('success')): En esta línea se comprueba si hay un mensaje de éxito almacenado en la sesión. Si existe, se muestra un cuadro de alerta con el mensaje.
* Luego se crea una tarjeta (<div class="card">) que contiene una tabla con las categorías. En esta tabla se muestra información como el id, nombre, descripción e imagen de cada categoría.
* Dentro de la iteración @foreach ($categorias as $categoria) se muestran los datos de cada categoría en filas de la tabla. También se incluyen botones de edición y eliminación para cada categoría.
* Al final de la vista, se muestra la paginación utilizando el componente de paginación de Bootstrap 4 ({{ $categorias->links('pagination::bootstrap-4') }}).

En resumen, esta vista muestra una lista de categorías con la posibilidad de agregar, editar y eliminar categorías. Utiliza el diseño y los estilos proporcionados por AdminLTE para lograr una apariencia agradable y coherente. :::

Si después de implementar el código y especificar la ruta desea probar la vista, sustituya las rutas especificadas por el símbolo “#” para evitar errores. Una vez probado, le recomendamos volver a las rutas originales.

A continuación, vamos a definir la ruta asociada a la vista. Diríjase al archivo de rutas **routes.php** y escriba el siguiente código:

use App\Http\Controllers\Admin\CategoriaController;  
  
Route::get('/categorias', [CategoriaController::class, 'index'])->name('admin.categoria.index');

No olvide incluir la referencia al controlador **CategoriaController** al principio del archivo:

<?php  
  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\Route;  
use App\Http\Controllers\Admin\HomeController;  
use App\Http\Controllers\Admin\CategoriaController;  
use App\Http\Controllers\Admin\SubCategoriaController;  
use App\Http\Controllers\Admin\UserController;

Ahora si desde la barra de direcciones tecla la siguiente dirección 'http://localhost:8000/admin/categorias', obtendrá la siguiente vista.

|  |
| --- |
| Vista categorias |

Vista categorias

Como esta forma de acceder a la vista no es de lo más cómodo para el usuario, vamos a modificar el **sidebar** de nuestro **panel de administración**. Para ello editaremos el archivo **config.php**, busque la siguiente sección y notifíquela.

['header' => 'ADMINISTRACIÓN'],  
[  
 'text' => 'Categorías',  
 'route' => 'admin.categoria.index',  
 'icon' => 'fas fa-fw fa-layer-group',  
],

### Agregar categorías

En este apartado vamos a ver como habilitar el botón de **Agregar** de nuestra vista index. Para ello deberemos crear un nuevo método en nuestro controlador **CategoriaController**.

public function create()  
 {  
 return view('admin.categorias.create');  
 }

Como puede observar el código del método **create** es bastante simplemente, llamamos a la vista admin.categorias.create. Diríjase la carpeta resources\views\admin\categorias y cree el archivo create.blade.php.

@extends('adminlte::page')  
  
@section('title', 'Agregar Categoría')  
  
@section('content\_header')  
 <h1>Agregar Categoría</h1>  
@stop  
  
@section('content')  
 <div class="container">  
 <div class="card col-lg-12 p-4">  
 {!! Form::open(['route' => 'admin.categoria.store', 'files' => true, 'method' => 'POST']) !!}  
  
 <div class="form-group">  
 {!! Form::label('nombre', 'Nombre:') !!}  
 {!! Form::text('nombre', null, ['class' => 'form-control'.($errors->has('nombre') ? ' is-invalid' : ''), 'required']) !!}  
 @if ($errors->has('nombre'))  
 <div class="invalid-feedback">  
 {{ $errors->first('nombre') }}  
 </div>  
 @endif  
 </div>  
  
 <div class="form-group">  
 {!! Form::label('descripcion', 'Descripción:') !!}  
 {!! Form::textarea('descripcion', null, ['class' => 'form-control'.($errors->has('descripcion') ? ' is-invalid' : ''), 'required']) !!}  
 @if ($errors->has('descripcion'))  
 <div class="invalid-feedback">  
 {{ $errors->first('descripcion') }}  
 </div>  
 @endif  
 </div>  
  
 <div class="form-group">  
 {!! Form::label('imagen', 'Imagen:') !!}  
 {!! Form::file('imagen', ['class' => 'form-control-file'.($errors->has('imagen') ? ' is-invalid' : ''), 'accept' => 'image/\*', 'onchange' => 'previewImage(event)']) !!}  
 @if ($errors->has('imagen'))  
 <div class="invalid-feedback">  
 {{ $errors->first('imagen') }}  
 </div>  
 @endif  
 <img id="preview" src="#" alt="Preview de la imagen" style="display: none; max-width: 200px; margin-top: 10px;">  
 </div>  
  
 {!! Form::submit('Agregar', ['class' => 'btn btn-primary']) !!}  
  
 {!! Form::close() !!}  
 </div>  
 </div>  
  
 <script>  
 function previewImage(event) {  
 var input = event.target;  
 var preview = document.getElementById('preview');  
  
 if (input.files && input.files[0]) {  
 var reader = new FileReader();  
  
 reader.onload = function (e) {  
 preview.src = e.target.result;  
 preview.style.display = 'block';  
 }  
  
 reader.readAsDataURL(input.files[0]);  
 }  
 }  
 </script>  
@stop  
@extends('adminlte::page')  
  
@section('title', 'Agregar Categoría')  
  
@section('content\_header')  
 <h1>Agregar Categoría</h1>  
@stop

:::info ¿Qué hace?

* @extends('adminlte::page'): Esta línea indica que la vista extiende el layout proporcionado por AdminLTE, lo cual significa que heredará estilos y funcionalidades predefinidas del tema.
* @section('title', 'Agregar Categoría'): Aquí se define el título de la página, que en este caso será “Agregar Categoría”.
* @section('content\_header'): Esta sección define el contenido que se mostrará en el encabezado de la página. En este caso, se muestra el título “Agregar Categoría”.
* @section('content'): Aquí comienza la sección principal de contenido de la página. A partir de este punto, se empieza a definir el contenido que se mostrará en el cuerpo de la página.
* Dentro del div con clase “container” se crea una tarjeta (<div class="card">) que contendrá un formulario para agregar una categoría.
* El formulario se crea utilizando el método Form::open, que especifica la ruta de destino del formulario, los atributos files y method, que permiten el envío de archivos y el método HTTP POST respectivamente.
* Dentro del formulario, hay tres campos de entrada: nombre, descripción e imagen. Cada campo se crea utilizando el método Form::text o Form::textarea, y se le asignan atributos como clase y validación de errores. Si hay errores de validación, se muestra un mensaje de error debajo del campo correspondiente.
* El campo de imagen utiliza el método Form::file y tiene un evento onchange que llama a la función previewImage para mostrar una vista previa de la imagen seleccionada. La vista previa se muestra en un elemento img con id “preview”.
* El botón “Agregar” se crea utilizando el método Form::submit y tiene la clase “btn btn-primary”.
* Al final de la vista, se encuentra un script JavaScript que define la función previewImage. Esta función lee el archivo de imagen seleccionado por el usuario y muestra una vista previa de la imagen en el elemento img con id “preview”.

En definitiva, esta vista muestra un formulario para agregar una categoría, que incluye campos para el nombre, descripción e imagen de la categoría. También permite la visualización de una vista previa de la imagen seleccionada antes de enviar el formulario. ```