

## Explicación del módulo Categoría

**Resumen:** El módulo gestiona las categorías de productos (CRUD completo): listado, creación, edición y eliminación. Archivos principales: [app/Models/Category.php](#), [app/Http/Controllers/CategoryController.php](#), [database/migrations/2025\\_12\\_20\\_221333\\_create\\_categories\\_table.php](#) y vistas en [resources/views/categories](#).

### 1) `Category` (Modelo) — `app/Models/Category.php`

**HasFactory:** permite usar factories para pruebas y seeders.

**\$fillable:** `['name', 'description']` lista de atributos asignables masivamente; protege contra Mass Assignment.

**products():** relación `hasMany(Product::class)`. Una categoría puede tener muchos productos; permite acceder con `$category->products` o contar con `$category->products()->count()`.

### 2) Migración —

`database/migrations/2025_12_20_221333_create_categories_table.php`

**Schema::create('categories')** define la tabla:

- `$table->id()` — clave primaria auto-incremental.
- `$table->string('name')` — nombre (requerido).
- `$table->text('description')->nullable()` — descripción opcional.
- `$table->timestamps()` — `created_at` y `updated_at`.

**down()** elimina la tabla con `Schema::dropIfExists('categories')`. Para aplicar: `php artisan migrate`.

### 3) Controlador — `app/Http/Controllers/CategoryController.php`

**\_\_construct():** aplica `middleware('auth')` para proteger rutas (solo usuarios autenticados).

**index():** obtiene todas las categorías con `Category::all()` y retorna la vista `categories.index` con `categories`.

**create():** muestra el formulario de creación (`categories.create`).

**store(Request \$request):** valida ( 'name' => 'required', 'description' => 'nullable' ) y crea la categoría con `Category::create($request->all())`. Redirige con mensaje de éxito.

**edit(Category \$category):** usa route-model binding y retorna `categories.edit` con la instancia.

**update(Request \$request, Category \$category):** valida y actualiza con `$category->update($request->all())`.

**destroy(Category \$category):** elimina con `$category->delete()` y redirige.

La validación mínima y `$fillable` protegen contra asignaciones no deseadas.

#### 4) Vistas — `resources/views/categories`

**index.blade.php:**

- Muestra tabla con *ID, Nombre, Descripción, Productos*.
- Usa `Str::limit($category->description, 50)` para acortar texto.
- Muestra conteo de productos con `$category->products_count ?? $category->products()->count()` (usa `withCount` si está cargado).
- Formulario de eliminación incluye `@csrf` y `@method('DELETE')` con confirmación JS.

**create.blade.php** y **edit.blade.php:** formularios con `@csrf`, validación de errores mostrada con `$errors->all()`, y en `edit` se usa `@method('PUT')` y se rellenan los campos con los valores actuales.

#### 5) Puntos de integración y mejoras recomendadas

- Rutas: usar `Route::resource('categories', CategoryController::class)` en `routes/web.php`.
- Eager loading: cambiar `index()` a `Category::withCount('products')->get()` para evitar N+1 al contar productos.
- Validación ampliada: reglas de longitud, formatos y sanitización.
- Autorización: usar Políticas si no todos los usuarios pueden modificar categorías.
- Soft deletes: agregar `SoftDeletes` y columna `softDeletes()` si se desea recuperación.
- Tests: crear tests de feature para `store/update/destroy` y acceso protegido por `auth`.

Generado automáticamente: breve explicación del módulo categoría para presentar el ejercicio.