

Catálogo de Productos

Esta guía describe los **endpoints públicos de catálogo** disponibles en la API de **Laravel**.
Permiten listar, filtrar y consultar productos con distintos criterios (búsqueda, categoría, subcategoría, precios, etc.).

Formato: todos los endpoints son **GET** y retornan **JSON**.
No requieren autenticación (a menos que tu política de acceso lo indique).

Resumen de Endpoints

Acción	Método	Ruta	Params / Query	Auth
Listar productos (general)	GET	/catalog	✔ Filtros disponibles	✗
Detalle de producto	GET	/catalog/{id}	—	✗
Productos por categoría	GET	/catalog/category/{id}	✔ Filtros básicos	✗
Productos por subcategoría	GET	/catalog/subcategory/{id}	✔ Filtros básicos	✗

1 Listado general — GET /catalog

Lista todos los productos con posibilidad de aplicar múltiples **filtros**.

Parámetros (query)

Nombre	Tipo	Descripción
q	string	Texto de búsqueda (nombre, descripción, etc.)
category_id	integer	ID de la categoría
subcategory_id	string	ID(s) de subcategoría separados por coma (ej: 1,3,7)
min_price	integer	Precio mínimo
max_price	integer	Precio máximo
available_only	boolean	Solo productos disponibles (true / false)
sort	string	Orden: price_asc , price_desc , newest , oldest
per_page	integer	Paginación (1–100, opcional)

Ejemplo de uso

```
fetch(`${BASE_URL}/catalog?q=camisa&min_price=100&max_price=500&sort=price_asc`)
  .then(r => r.json())
  .then(console.log)
  .catch(console.error);
```

Ejemplo de respuesta (éxito)

```
{
  "success": true,
  "message": "Listado de productos",
  "data": {
    "current_page": 1,
    "data": [
      {
        "id": 15,
        "name": "Camisa Azul",
        "price": 199,
        "category": "Ropa",
        "available": true
      }
    ],
    "total": 32,
    "last_page": 4
  }
}
```

2 Detalle de producto — GET /catalog/{id}

Devuelve la información completa de un producto específico, incluyendo precios y descuentos activos.

Ejemplo de uso

```
fetch(`${BASE_URL}/catalog/15`)
  .then(r => r.json())
  .then(console.log)
  .catch(console.error);
```

Ejemplo de respuesta (éxito)

```
{
  "success": true,
  "message": "Detalle de producto",
  "data": {
    "id": 15,
    "name": "Camisa Azul",
    "description": "Camisa de algodón premium",
    "price": 199,
    "discount": {
      "percent": 10,
      "price_after_discount": 179
    },
    "category": "Ropa",
    "subcategory": "Camisas",
    "available": true
  }
}
```

Error — producto no encontrado

```
{
  "message": "The given data was invalid.",
  "errors": {
    "product": ["Producto no encontrado o no disponible."]
  }
}
```

3 Productos por categoría — GET /catalog/category/{id}

Filtra productos por **ID de categoría**.
Admite los mismos filtros básicos que el listado general (excepto `category_id`).

Parámetros (query)

Nombre	Tipo	Descripción
q	string	Texto de búsqueda
min_price	integer	Precio mínimo
max_price	integer	Precio máximo
available_only	boolean	Solo disponibles
sort	string	Orden (price_asc , price_desc , newest , oldest)
per_page	integer	Tamaño de página

Ejemplo de uso

```
fetch(`${BASE_URL}/catalog/category/3?available_only=true&sort=newest`)
  .then(r => r.json())
  .then(console.log)
  .catch(console.error);
```

Ejemplo de respuesta

```
{
  "success": true,
  "message": "Productos por categoría",
  "data": {
    "data": [
      { "id": 21, "name": "Pantalón negro", "price": 299 }
    ]
  }
}
```

4 Productos por subcategoría — GET /catalog/subcategory/{id}

Filtra productos por **ID de subcategoría**.

Permite los mismos parámetros de búsqueda que /category/{id} .

Ejemplo de uso

```
fetch(`${BASE_URL}/catalog/subcategory/8?min_price=50&max_price=200`)
  .then(r => r.json())
  .then(console.log)
  .catch(console.error);
```

Ejemplo de respuesta

```
{
  "success": true,
  "message": "Productos por subcategoría",
  "data": {
    "data": [
      { "id": 52, "name": "Gorra roja", "price": 149 }
    ]
  }
}
```

Posibles respuestas de error

Tipo	Código	Ejemplo
Validación de parámetros	422	<pre>{ "message": "The given data was invalid.", "errors": { "sort": ["El valor no es válido."] } }</pre>
Producto no encontrado	422	<pre>{ "errors": { "product": ["Producto no encontrado o no disponible."] } }</pre>

Notas adicionales

- Todos los endpoints retornan el campo "success": true/false .
- Se usa paginación estándar de Laravel (data , current_page , last_page , etc.).
- El parámetro subcategory_id admite **múltiples valores separados por coma**.
- Si no se envían filtros, se listan todos los productos activos.