

Almacen API

Este proyecto es una API construida con ****.NET 8.0**** que permite gestionar un sistema de inventario de almacén. Incluye operaciones CRUD (Crear, Leer, Actualizar y Eliminar) para productos y categorías, con una relación de claves foráneas entre ambas entidades.

Tabla de Contenidos

- [Requisitos](#requisitos)
- [Dependencias](#dependencias)
- [Instalación](#instalación)
- [Ejecución](#ejecución)
- [Pruebas](#pruebas)
- [EndPoints Principales](#endpoints-principales)
- [Contribución](#contribución)
- [Licencia](#licencia)

Requisitos

Asegúrate de tener los siguientes requisitos antes de comenzar:

- ****.NET 8.0 SDK**** o superior
- ****SQL Server**** (u otro motor de base de datos compatible)
- ****Entity Framework Core**** para la gestión de datos
- Un editor de texto o entorno de desarrollo como Visual Studio o Visual Studio Code

Dependencias

El proyecto usa las siguientes dependencias principales:

- ****Microsoft.EntityFrameworkCore.SqlServer****: Para interactuar con SQL Server.
- ****Microsoft.EntityFrameworkCore.Tools****: Proporciona las herramientas necesarias para las migraciones de la base de datos.
- ****Microsoft.Extensions.DependencyInjection****: Utilizado para la inyección de dependencias.
- ****Microsoft.AspNetCore.Mvc****: Para construir controladores de API REST.
- ****xUnit****: Framework de pruebas unitarias.
- ****Moq****: Para crear mocks en las pruebas unitarias.

Instalación

Clonación del Repositorio

Primero, clona el repositorio desde GitHub:

```
https://github.com/No-Haiku/Ejercicios.git  
cd AlmacenAPI
```

Configuración de la Base de Datos

Abre el archivo appsettings.json y edita la cadena de conexión para que apunte a tu servidor de base de datos SQL Server.

```
"ConnectionStrings": {  
  "DefaultConnection":  
    "Server=DESKTOP-EKGHLPG\MSSQLSERVER01;Database=AlmacenDB;Trusted_Connection=True;MultipleActiveResultSets=true;TrustServerCertificate=True"  
}
```

Aplica las migraciones de la base de datos para crear las tablas necesarias ejecutando el siguiente comando:

```
dotnet ef database update
```

Esto creará las tablas Productos y Categorías en la base de datos.

Ejecución

Para iniciar la API, ejecuta el siguiente comando:

```
dotnet run
```

Variables de entorno

Puedes configurar las variables de entorno específicas para las pruebas o el entorno de producción editando el archivo launchSettings.json o estableciéndolas en tu entorno local.

Pruebas

Este proyecto incluye pruebas unitarias para garantizar el correcto funcionamiento de los repositorios y los controladores.

Ejecutar Pruebas

Para ejecutar todas las pruebas, utiliza el siguiente comando:

```
dotnet test
```

Ejemplo de Test Unitario

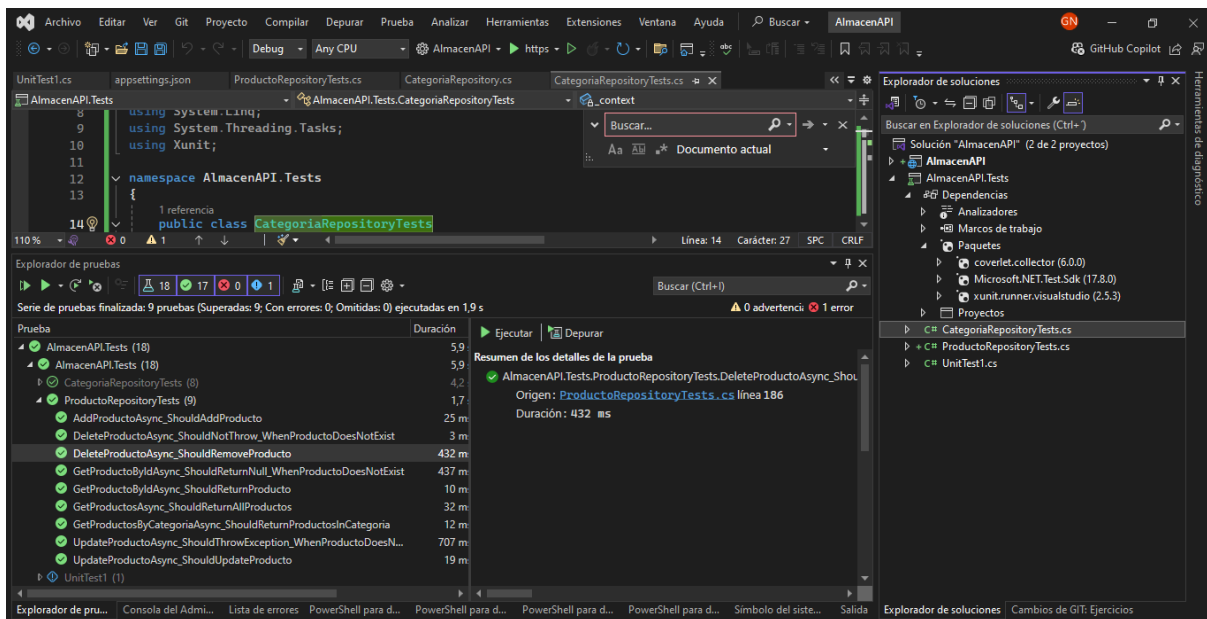
A continuación, se muestra un ejemplo de una prueba unitaria que valida la eliminación de un producto:

```
[Fact]  
public async Task DeleteProductoAsync_ShouldRemoveProducto()  
{  
    // Arrange  
    var producto = new Producto { Nombre = "ProductoTest", Descripcion = "DescripcionTest"  
};  
    await _context.Productos.AddAsync(producto);  
    await _context.SaveChangesAsync();
```

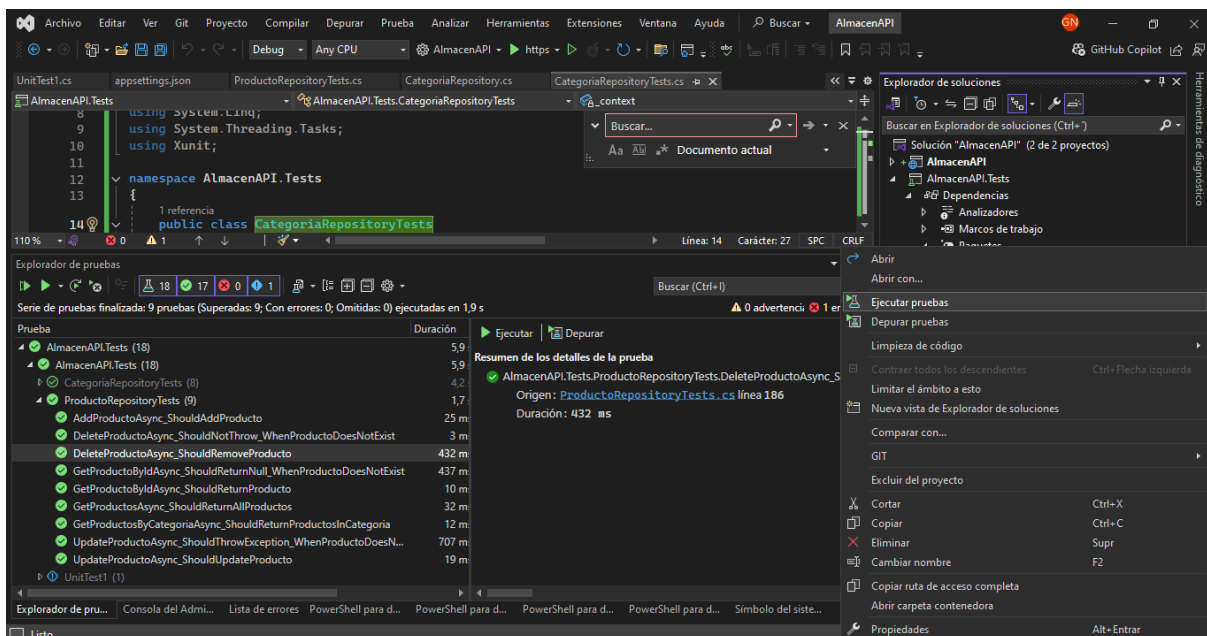
```
// Act
await _repository.DeleteProductoAsync(producto.Id);
await _context.SaveChangesAsync();

// Assert
var result = await _context.Productos.FindAsync(producto.Id);
Assert.Null(result);
}
```

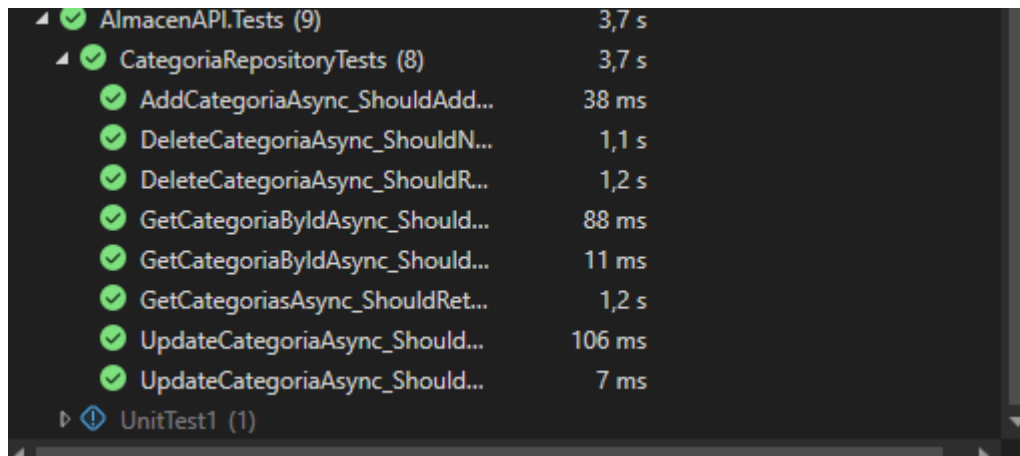
Como usar el Test ; SELECCIONAR TEST



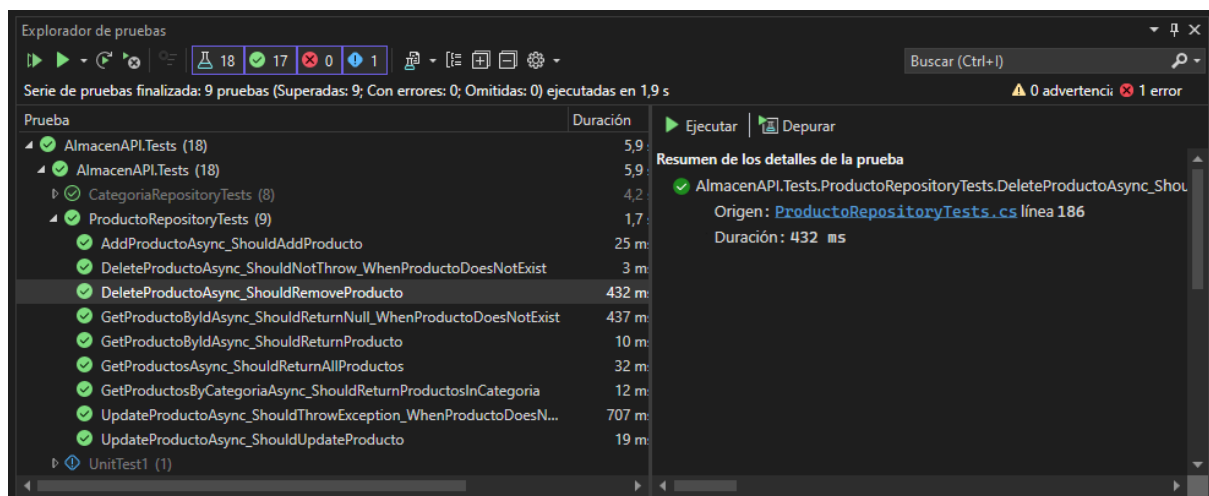
CLICK DERECHO EJECUTAR PRUEBAS



EJEMPLOS EXITOSOS CATEGORIA:



PRODUCTO:



EndPoints Principales

La API expone los siguientes endpoints para gestionar productos y categorías:

Productos

GET /api/productos

Devuelve todos los productos.

GET /api/productos/{id}

Devuelve un producto específico por su ID.

POST /api/productos

Crea un nuevo producto.

PUT /api/productos/{id}

Actualiza un producto existente.

DELETE /api/productos/{id}

Elimina un producto.

Categorías

GET /api/categorias

Devuelve todas las categorías.

GET /api/categorias/{id}

Devuelve una categoría específica por su ID.

POST /api/categorias

Crea una nueva categoría.

PUT /api/categorias/{id}

Actualiza una categoría existente.

DELETE /api/categorias/{id}

Elimina una categoría.

Uso de Swagger :

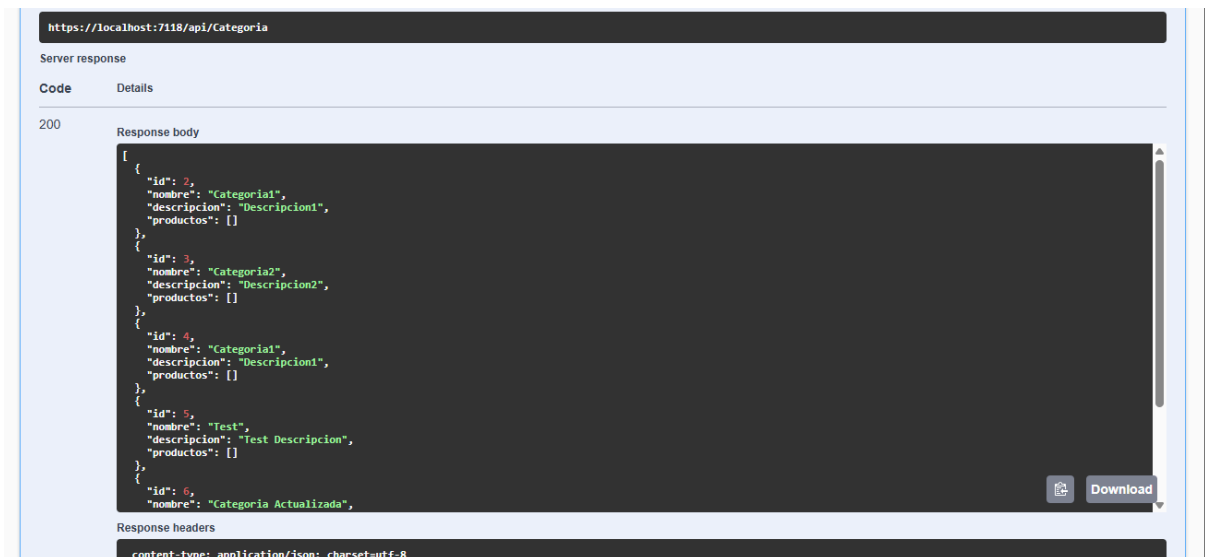
GET CATEGORIA : devuelve todas las categorías

The image shows a Swagger UI interface for an API endpoint named 'Categoria'. The interface is divided into several sections:

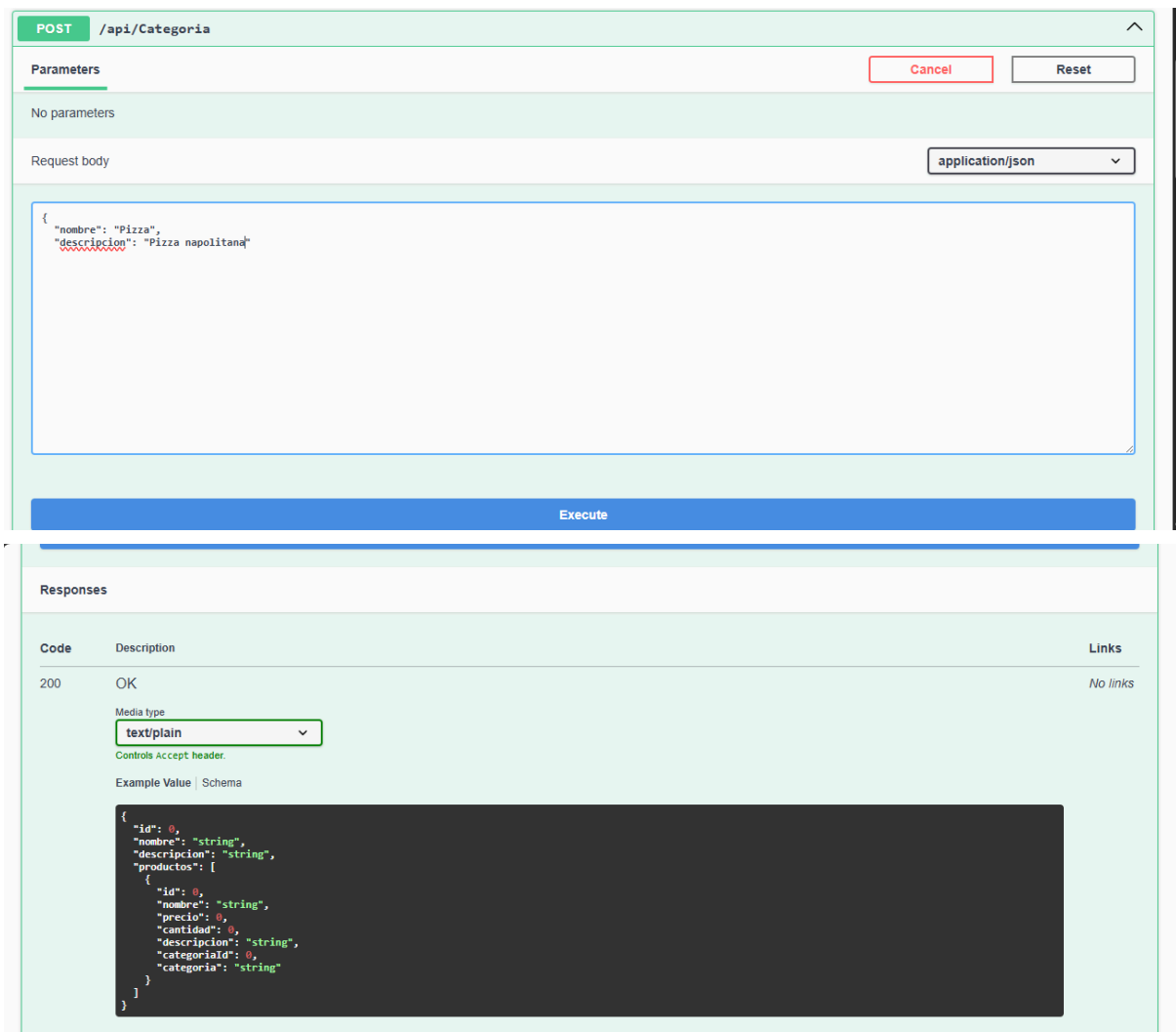
- Header:** Displays the endpoint name 'Categoria' and a 'Cancel' button.
- Method and Path:** Shows the HTTP method 'GET' and the path '/api/Categoria'.
- Parameters:** A section labeled 'Parameters' with a 'No parameters' message and an 'Execute' button.
- Responses:** A section labeled 'Responses' with a 'Curl' button.
- Curl:** A text area containing the command:

```
curl -X 'GET' \
  'https://localhost:7118/api/Categoria' \
  -H 'accept: text/plain'
```
- Request URL:** A text area containing the URL:

```
https://localhost:7118/api/Categoria
```
- Server response:** A section with a 'Code' and 'Details' tab.



POST CATEGORIA : hace un post de una categoria



GET CATEGORIA ID: devuelve una categoria por Id

GET /api/Categoria/{id}

Parameters Cancel

Name	Description
id * required	
integer(\$int32)	2
(path)	

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7118/api/Categoria/2' \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7118/api/Categoria/2

Server response

Code Details

Request URL

https://localhost:7118/api/Categoria/2

Server response

Code Details

200

Response body

```
{
  "id": 2,
  "nombre": "Categoría1",
  "descripcion": "Descripción1",
  "productos": []
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Thu, 10 Oct 2024 06:07:04 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	OK	No links

Media type

text/plain

Controls Accept header.

Example Value | Schema

```
{
  "id": 0,
```

PUT CATEGORIA ID : actualiza una categoria segun su id

PUT /api/Categoria/{id}

Parameters Cancel Reset

Name	Description
id * required	
integer(\$int32)	2
(path)	

Request body

application/json

```
{
  "id": 2,
  "nombre": "Actualizacion",
  "descripcion": "Se Actualizo"
}
```

Curl

```
curl -X 'PUT' \
  'https://localhost:7118/api/Categoria/2' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 2,
    "nombre": "Actualizacion",
    "descripcion": "Se Actualizo"
  }'
```

Request URL

<https://localhost:7118/api/Categoria/2>

Server response

Code	Details
200	<p>Response body</p> <pre>{ "id": 2, "nombre": "Actualizacion", "descripcion": "Se Actualizo", "productos": [] }</pre> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Thu, 10 Oct 2024 06:09:12 GMT server: Kestrel</pre>

Responses

Code	Description	Links
------	-------------	-------

DELETE CATEGORIA ID: elimina una categoria por su Id

DELETE /api/Categoria/{id}

Parameters

Name	Description
id * required	
integer(\$int32)	
(path)	

Execute Clear

Responses

Curl

```
curl -X 'DELETE' \
  'https://localhost:7118/api/Categoria/2' \
  -H 'accept: */*'
```

Request URL

<https://localhost:7118/api/Categoria/2>

Server response

Code	Details
204 <i>Undocumented</i>	<p>Response headers</p> <pre>date: Thu, 10 Oct 2024 06:10:52 GMT server: Kestrel</pre>

Responses

Code	Description	Links
200	OK	No links

POST PRODUCTO: crea un producto

POST

/api/Productos

Parameters

Cancel

Reset

No parameters

Request body

application/json

```
{
  "nombre": "Bruno",
  "precio": 20,
  "cantidad": 440,
  "descripcion": "testing",
  "categoriaId": 4
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7118/api/Productos' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
    "nombre": "Bruno",
    "precio": 20,
    "cantidad": 440,
    "descripcion": "testing",
    "categoriaId": 4
  }'
```

Request URL

https://localhost:7118/api/Productos

Server response

Code

Details

201

Undocumented

Response body

```
{
  "id": 116,
  "nombre": "Bruno",
  "precio": 20,
  "cantidad": 440,
  "descripcion": "testing",
  "categoriaId": 4,
  "categoria": null
}
```

Download

GET PRODUCTOS : devuelve todos los productos

GET

/api/Productos

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7118/api/Productos' \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7118/api/Productos

Server response

Code

Details

Request URL

https://localhost:7118/api/Productos

Server response

Code	Details
200	<p>Response body</p> <pre>[{"id": 116, "nombre": "Bruno", "precio": 20, "cantidad": 440, "descripcion": "testing", "categoriaId": 4, "categoria": null}, {"id": 117, "nombre": "Bruno2", "precio": 20, "cantidad": 440, "descripcion": "testing", "categoriaId": 4, "categoria": null}]</pre> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Thu, 10 Oct 2024 06:18:45 GMT server: Kestrel</pre>

Responses

Code	Description	Links
------	-------------	-------

GET PRODUCTOS ID : devuelve un producto segun su Id

GET /api/Productos/{id}

Parameters

Name	Description
id * required	
integer(\$int32)	116
(path)	

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7118/api/Productos/116' \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7118/api/Productos/116

Server response

Curl

```
curl -X 'GET' \
  'https://localhost:7118/api/Productos/116' \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7118/api/Productos/116

Server response

Code	Details
200	<p>Response body</p> <pre>{ "id": 116, "nombre": "Bruno", "precio": 20, "cantidad": 440, "descripcion": "testing", "categoriaId": 4, "categoria": null }</pre> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Thu, 10 Oct 2024 06:17:19 GMT server: Kestrel</pre>

Responses

Code	Description	Links
------	-------------	-------

PUT PRODUCTO ID : actualiza un producto segun su id

PUT

/api/Productos/{id}

Parameters

Cancel

Reset

Name	Description
id * required	
integer(\$int32)	117
(path)	

Request body

application/json

```
{  "id": 117,  "nombre": "Juan",  "precio": 101,  "categoriaId": 4,  "cantidad": 20,  "descripcion": "mishi"}
```

Curl

```
curl -X 'PUT' \
  'https://localhost:7118/api/Productos/117' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 117,
    "nombre": "Juan",
    "precio": 101,
    "categoriaId": 4,
    "cantidad": 20,
    "descripcion": "mishi"
  }'
```

Request URL

https://localhost:7118/api/Productos/117

Server response

Code	Details
200	<div>Response body</div> <div><pre>{ "id": 117, "nombre": "Juan", "precio": 101, "categoriaId": 4, "cantidad": 20, "descripcion": "mishi", "categoria": null}</pre></div> <div>Response headers</div> <div>content-type: application/json; charset=utf-8 date: Thu, 10 Oct 2024 06:21:02 GMT server: Kestrel</div>

Download

DELETE PRODUCTO ID: elimina un producto segun su Id

DELETE

/api/Productos/{id}

Parameters

Cancel

Name	Description
id * required	
integer(\$int32)	116
(path)	

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
  'https://localhost:7118/api/Productos/116' \
  -H 'accept: */*' \
```

Request URL

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
  'https://localhost:7118/api/Productos/116' \
  -H 'accept: */*'
```

Request URL

https://localhost:7118/api/Productos/116

Server response

Code

Details

204
Undocumented

Response headers

```
date: Thu, 10 Oct 2024 06:22:25 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	OK	No links

GET PRODUCTO POR CATEGORIA : devuelve todos los productos de una categoria asociada

GET

/api/Productos/categoria/{categoriaId}

Cancel

Parameters

Name	Description
categoriaId * required integer(\$int32) (path)	<input type="text" value="3"/>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7118/api/Productos/categoria/3' \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7118/api/Productos/categoria/3

Server response

Code

Details

200

Response body

```
[
  {
    "id": 120,
    "nombre": "juan1",
    "precio": 20,
    "cantidad": 440,
    "descripcion": "testing",
    "categoriaId": 3,
    "categoria": null
  },
  {
    "id": 121,
    "nombre": "juan21",
    "precio": 20,
    "cantidad": 440,
    "descripcion": "testing",
    "categoriaId": 3,
    "categoria": null
  }
]
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Thu, 10 Oct 2024 06:24:29 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	OK	No links