

Documentación del API RESTFUL – Gestión de Inventario de Productos

Base URL: <http://localhost:8080>

Formato de datos: JSON

Autenticación: No requerida

Endpoints:

1.GET /product

Descripción: obtiene la lista completa de productos

Parámetros de entrada:

Ninguno

Para respuesta HTTP 200 (exitosa):

```
[  
 {  
   "idProducto": 1,  
   "nombreProducto": "Chai",  
   "idProveedor": 1,  
   "idCategoria": 1,  
   "unidad": "10 boxes x 20 bags",  
   "precio": 18  
 },  
 {  
   "idProducto": 14,  
   "nombreProducto": "Tofu",  
   "idProveedor": 6,  
   "idCategoria": 7,  
   "unidad": "40 – 100 g pkgs.",  
   "precio": 23  
 }]  
 ]
```

Estados:

200 OK: se obtiene lista de forma exitosa

500 Internal Server Error: Error del servidor

2. GET /producto/{id}

Descripción: Se llama un producto por su ID

Parámetros de entrada:

id (path parameter): ID del producto (pedido)

Respuesta exitosa (HTTP 200):

```
[  
 {  
   "idProducto": 1,  
   "nombreProducto": "Chai",  
   "idProveedor": 1,  
   "idCategoria": 1,  
   "unidad": "10 boxes x 20 bags",  
   "precio": 18  
 },  
 ]
```

Respuesta de error HTTP 400:

```
{  
   "mensaje": "Producto no encontrado con ID: 99999"  
 }
```

Estados:

- 200 OK: producto encontrado y devuelto

- 400 Bad Request: ID no valido, producto no encontrado

- 500 Internal Server Error: error del servidor

3. POST /product

Descripción: Crea un nuevo producto en la tabla

Parámetros de entrada:

Body (JSON): datos del nuevo producto

```
{  
  "nombreProducto": "Nuevo-Producto",  
  "idProveedor": 31,  
  "idCategoria": 9,  
  "unidad": "24 unidades",  
  "precio": 25  
}
```

Respuesta exitosa HTTP 201:

```
{  
  "mensaje": "Producto creado exitosamente",  
  "id": "78"  
}
```

Respuesta al error HTTP 400:

```
{  
  "mensaje": "El nombre del producto es requerido"  
}
```

Estado:

201 Created: Producto creado de forma exitosa

400 Bad Request: Datos no validos para la solicitud

500 Internal Server Error: Error del servidor

4. PUT /product

Descripción: Actualizar un producto que ya existe

Parámetros de entrada:

Body JSON: datos reemplazados con los actualizados

```
{  
    "idProducto": 1,  
    "nombreProducto": "Chai-Actualizado",  
    "idProveedor": 1,  
    "idCategoria": 1,  
    "unidad": "12 boxes x 20 bags",  
    "precio": 21  
}
```

Respuesta de forma exitosa HTTP 200:

```
{  
    "mensaje": "Producto actualizado exitosamente"  
}
```

Respuesta al error HTTP 400:

```
{  
    "mensaje": "Producto no encontrado con ID: 99999"  
}
```

Estados:

200 OK: Producto actualizado de forma exitosa

400 Bad Request: idProducto no especificado, producto no encontrado o con datos no validos.

500 Internal Server Error: Error del servidor

5. DELETE /producto/{id}

Descripción: se elimina un producto según su ID

Parámetros de entrada:

Id: ID del producto requerido a eliminar

Respuesta de forma exitosa HTTP 200:

```
{  
    "mensaje": "Producto eliminado exitosamente"  
}
```

Respuesta al error de forma exitosa HTTP 400:

```
{  
    "mensaje": "Producto no encontrado con ID: 9999"  
}
```

Estados:

200 OK: Producto eliminado de forma exitosa

400 Bad Request: ID no valido o no encontrado

500 Internal Server Error: Error del servidor

Estructura:

Campo	Tipo	Descripción	Se requiere
IdProducto	Integer	Identificador único	En PUT
nombreProducto	String	Nombre del producto	Si
idProveedor	Integer	ID de él proveedor	Si
idCategoria	Integer	ID de la categoría	Si
unidad	String	Unidades	Si
precio	BigDecimal	Precio	Si

Consideraciones:

- Validaciones:

Todos los IDs deben ser números positivos enteros

Campo requerido: nombreProducto, el precio debe ser un valor positivo

- Manejo de errores:
 - Errores de validación devuelven HTTP 400 con un mensaje descriptivo
 - Errores del servidor devuelven HTTP 500
 - Todas las respuestas son en JSON
- Base de datos:
 - URL: jdbc:mysql://lewisrp.dev:3306/Northwind
 - Tabla: Products
 - (Credenciales a solicitar)

Habilitado para todos los origins*