



# Documentación de la API

# Microservicios

Antony Pesantes  
Damian Seminario

# 1. Auth Service (/api/auth)

## a. POST /api/auth/register

- **Descripción:** Registra un nuevo usuario.

- **Request Body:**

```
json  
{ "username": "string", "password": "string" }
```

- **Response:**

```
json  
{ "message": "User created successfully" }
```

- **Errores:**

- 400 Bad Request: Si faltan campos obligatorios.
- 500 Internal Server Error: Si ocurre un error en la base de datos.

## b. POST /api/auth/login

- **Descripción:** Inicia sesión y devuelve un token JWT.

- **Request Body:**

json  
{ "username": "string", "password": "string" }

- **Response:**

json  
{ "token": "string" }

- **Errores:**

- 400 Bad Request: Si el usuario no existe o la contraseña es incorrecta.
- 500 Internal Server Error: Si ocurre un error en la base de datos.

## 2. Product Service (/api/products)

### a. POST /api/products

- **Descripción:** Crea un nuevo producto (requiere autenticación).

- **Headers:**

- Authorization: Bearer <JWT\_TOKEN>

- **Request Body:**

json

```
{ "name": "string", "description": "string", "price": 0 }
```

- **Response:**

json

```
{ "_id": "string", "name": "string", "description": "string", "price": 0 }
```

- **Errores:**

- 400 Bad Request: Si falta información o los datos son incorrectos.
  - 401 Unauthorized: Si el token JWT no es válido o está ausente.
  - 500 Internal Server Error: Si ocurre un error en la base de datos.

## b. GET /api/products

- Descripción:** Obtiene todos los productos.

- Response:**

json

```
[ { "_id": "string", "name": "string", "description": "string", "price": 0 } ]
```

## c. GET /api/products/:id

- **Descripción:** Obtiene un producto por su ID.

- **Response:**

json

```
{ "_id": "string", "name": "string", "description": "string", "price": 0 }
```

- **Errores:**

- 404 Not Found: Si el producto no existe.

## d. PUT /api/products/:id

- Descripción:** Actualiza un producto por su ID (requiere autenticación).

- Headers:**

  - Authorization: Bearer <JWT\_TOKEN>

- Request Body:**

json

```
{ "name": "string", "description": "string", "price": 0 }
```

- Response:**

json

```
{ "_id": "string", "name": "string", "description": "string", "price": 0 }
```

- Errores:**

  - 400 Bad Request: Si faltan campos obligatorios.

  - 401 Unauthorized: Si el token JWT no es válido o está ausente.

  - 404 Not Found: Si el producto no existe.

## e. DELETE /api/products/:id

- Descripción:** Elimina un producto por su ID (requiere autenticación).

- Headers:**

- Authorization: Bearer <JWT\_TOKEN>

- Response:**

- json

- { "\_id": "string", "name": "string", "description": "string", "price": 0 }

- Errores:**

- 404 Not Found: Si el producto no existe.

- 401 Unauthorized: Si el token JWT no es válido o está ausente.