

# API de Inventario

---

Esta API permite gestionar un inventario, ofreciendo endpoints para consultar productos, agregar nuevos productos y actualizar el stock de productos existentes.

## Requisitos

- Python 3.8 o superior

## Instalación

1. Clona este repositorio o descarga el código fuente:

```
git clone <URL_DEL_REPOSITORIO>
cd api_inventario
```

2. Instala las dependencias:

```
pip install -r requirements.txt
```

## Ejecución

1. Ejecuta el archivo principal para iniciar el servidor:

```
python run.py
```

2. La API estará disponible en <http://127.0.0.1:5000/>.

## Endpoints

1. Consultar un producto

**GET** [/producto/<id\\_producto>](#)

Consulta un producto por su ID.

- **Parámetros:** ID del producto (entero positivo).
- **Respuesta exitosa:**

```
{
  "nombre": "Producto A",
```

```
"cantidad": 10
}
```

- **Errores posibles:**
  - **400 Bad Request:** ID inválido.
  - **404 Not Found:** Producto no encontrado.

## 2. Agregar un producto

### POST /producto

Agrega un nuevo producto al inventario.

- **Cuerpo de la solicitud (JSON):**

```
{
  "id_producto": 3,
  "cantidad": 15
}
```

- **Respuesta exitosa:**

```
{
  "mensaje": "Producto agregado exitosamente."
}
```

- **Errores posibles:**
  - **400 Bad Request:** ID o cantidad inválidos, o producto ya existente.

## 3. Actualizar el stock de un producto

### PUT /producto/<id\_producto>

Actualiza el stock de un producto existente.

- **Parámetros:** ID del producto (entero positivo).
- **Cuerpo de la solicitud (JSON):**

```
{
  "nueva_cantidad": 25
}
```

- **Respuesta exitosa:**

```
{
  "mensaje": "Stock actualizado exitosamente."
}
```

- **Errores posibles:**

- **400 Bad Request:** ID o cantidad inválidos.
- **404 Not Found:** Producto no encontrado.

## Pruebas

### Pruebas Manuales

1. Probar en **Postman** o **cURL** para probar los endpoints:

- **Ejemplo de consulta con cURL:**

```
curl -X GET http://127.0.0.1:5000/producto/1
```

- **Ejemplo de agregar un producto con cURL:**

```
curl -X POST http://127.0.0.1:5000/producto -H "Content-Type: application/json" -d '{"id_producto": 3, "cantidad": 15}'
```

### Pruebas Automatizadas

1. Ejecutar las pruebas unitarias:

```
pytest tests/
```

## Notas

- Este proyecto utiliza una base de datos simulada (en memoria). Los datos no se guardan de manera persistente.
- Para un entorno de producción, se recomienda usar una base de datos real y configurar correctamente la seguridad del servidor.