



StockFlow

Inventory Management

Chara, Trillini, Roldan y Targa



01 CONTEXTO

02 PROBLEMAS

03 SOLUCIONES

04 CÓDIGO: DB, API & CRUD, WEB

CONTEXTO



En el ámbito de las empresas, **la gestión de inventarios es un proceso crucial que impacta directamente en la eficiencia operativa y la rentabilidad.** Estas empresas necesitan mantener un equilibrio preciso entre el exceso de inventario o la falta de stock.

La digitalización y la automatización de estos procesos son esenciales para **optimizar la cadena de suministro y mejorar la toma de decisiones basada en datos.**

PROBLEMAS

- **Desbalance en el Inventario**
 - Sobre-stock
 - Desabastecimiento
- **Errores en el Registro de Datos**
- **Falta de Visibilidad en Tiempo Real**
- **Dificultad en la Gestión de Compras y Proveedores**
- **Ineficiencia Operativa**



SOLUCIONES

03

INVENTARIO EN TIEMPO REAL // PRECIO PROMEDIO DE COMPRA



StockFlow

[Inicio](#) [Añadir Producto](#) [Compras](#) [Ventas](#)

Inventario de Productos

ID	Nombre	Cantidad	Precio	Acciones	
1	Café	75 kg	\$33,066.67	Editar	Eliminar
3	Té	60 unidades	\$220.00	Editar	Eliminar
4	Yerba	30 kg	\$3,000.00	Editar	Eliminar
5	Jugo	50 unidades	\$220.00	Editar	Eliminar

[Añadir Producto](#)

[Carga de Compras](#)



CARGA DE COMPRAS // HISTORIAL DE COMPRAS

03

Carga de Compras

Fecha:

27/06/2024

Producto:

Selecciona un producto...

Cantidad:

Precio de Compra:

Cargar Compra

Historial de Compras

Nº de Orden	Fecha	Producto	Cantidad	Precio	Total
1	2024-06-01	Café	20	\$30,000.00	\$600,000.00

Carga de Venta

03

Fecha:

27/06/2024



Nombre del Cliente:

Carlos S. M.

Producto:

Café (Stock: 75)



Cantidad:

1

Precio de Venta:

6000

Descuento:

300

Cargo por Envío:

100

Cargar Orden

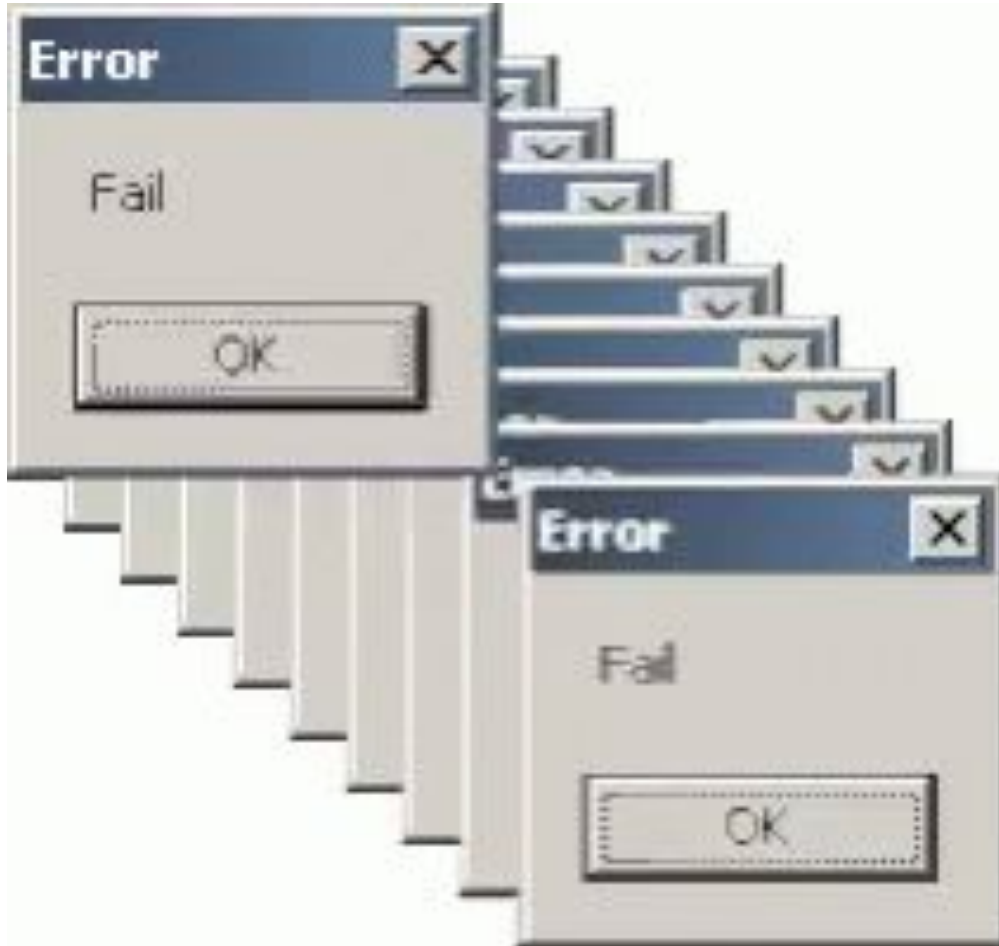




HISTORIAL DE VENTAS

Últimas Transacciones de Ventas

N° de Orden	Fecha	Cliente	Producto	Cantidad	Precio de Venta	Subtotal	Descuento	Cargo por Envío	Total
1	2024-06-27	Carlos S. M.	Café	1	\$6000.00	\$6000.00	\$300.00	\$100.00	\$5800.00
2	2024-06-27	Néstor K.	Jugo	1	\$200.00	\$200.00	\$0.00	\$0.00	\$200.00
3	2024-06-27	Mauricio M.	Yerba	10	\$5000.00	\$50000.00	\$1000.00	\$0.00	\$49000.00



CÓDIGO

PYTHON, SQL





DATABASE

Donde almacenamos toda la información

La creamos utilizando Python, operamos sobre ella con SQL



CONEXIÓN DB

```
def get_db():  
    if 'db' not in g:  
        g.db = sqlite3.connect('inventario.db')  
        g.db.row_factory = sqlite3.Row  
    return g.db
```

Con esta función, conectamos con la base de datos “inventario.db”, y si no existe, se crea una nueva.

CREACIÓN TABLAS

04

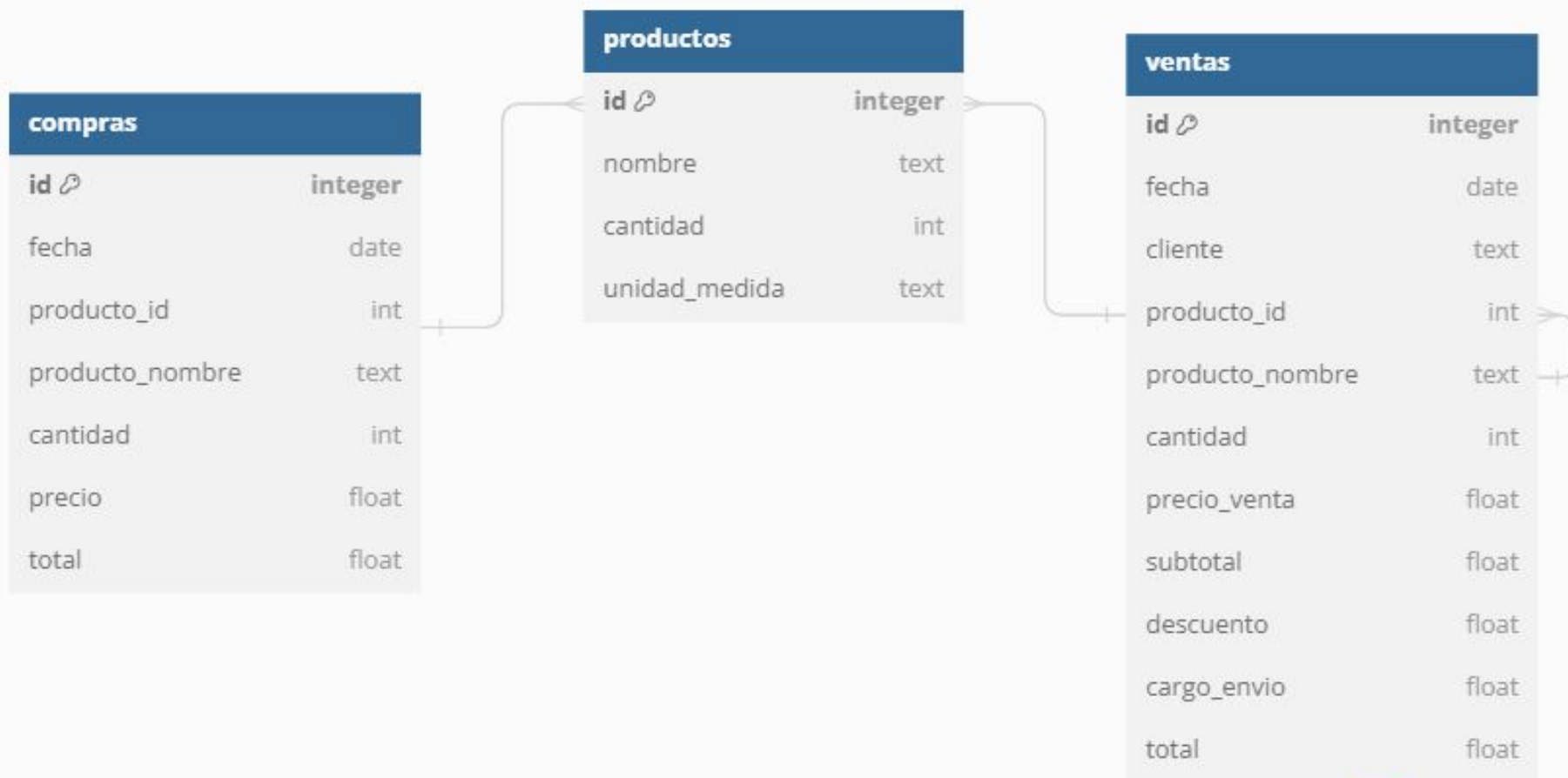
```
def create_productos_table():
    db = get_db()
    c = db.cursor()
    c.execute('''
CREATE TABLE IF NOT EXISTS productos (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    nombre TEXT,
    cantidad INTEGER,
    precio REAL,
    unidad_medida TEXT DEFAULT 'unidades'
)
''')
    db.commit()
```

```
def create_sales_table():
    db = get_db()
    c = db.cursor()
    c.execute('''
CREATE TABLE IF NOT EXISTS ventas (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    fecha TEXT,
    cliente TEXT,
    producto_id INTEGER,
    producto_nombre TEXT,
    cantidad INTEGER,
    precio_venta REAL,
    subtotal REAL,
    descuento REAL,
    cargo_envio REAL,
    total REAL,
    FOREIGN KEY (producto_id) REFERENCES productos (id)
)
''')
    db.commit()
```

Creamos las columnas y les definimos su formato a cada una.

TABLAS DEL DB

04





DATA ANALYTICS

¿Cómo podemos aprovechar la información?

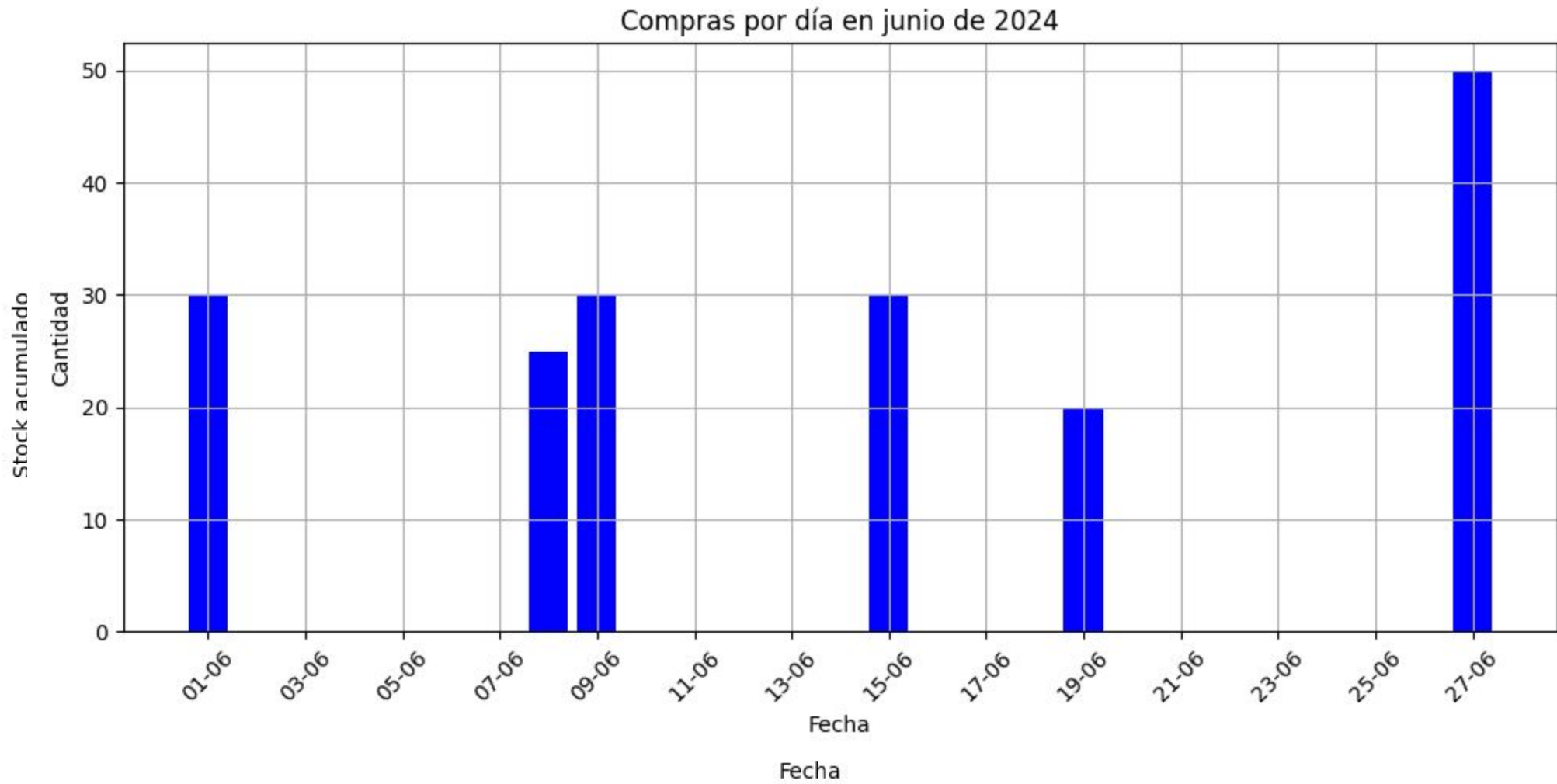
Análisis utilizando la librería Pandas



DASHBOARDS

04





OTROS USOS

- **Alertas de nivel mínimo de stock**
- **Análisis de transacciones y frecuencias**
- **Resultados del mes**
- **Previsionar demanda**

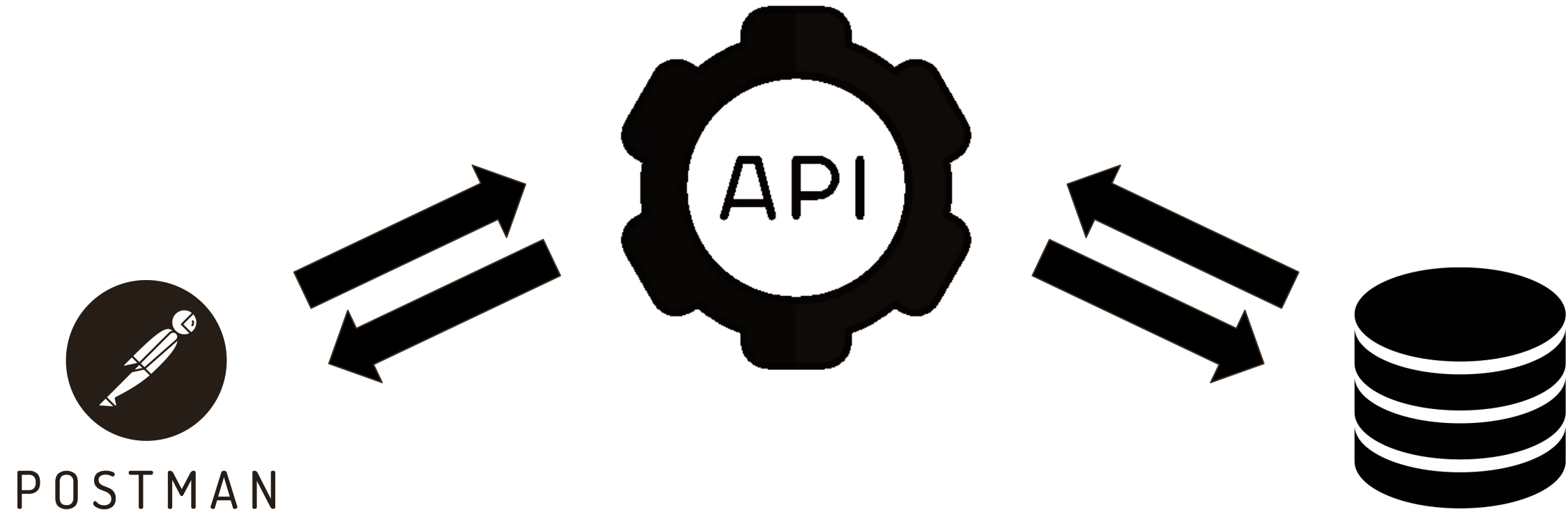


API

04

POST, GET, PUT, DELETE

Creamos los endpoints utilizando Python,
para poder conectarnos remotamente a la base de datos.



```
@app.route('/api/productos', methods=['GET'])
def get_productos_endpoint():
    productos = get_productos()
    return jsonify(productos)
```

```
▼ def get_productos():
    db = get_db()
    c = db.cursor()
    c.execute("SELECT * FROM productos")
    rows = c.fetchall()
    productos = [dict(row) for row in rows]
    return productos
```

**CREACIÓN DEL ENDPOINT
PARA LA API**

**FUNCIÓN DE SQL LLAMADA
PARA LEER LA TABLA Y
OBTENER LOS PRODUCTOS**

POST

04

```
@app.route('/api/productos', methods=['POST'])
def add_producto_endpoint():
    data = request.json
    if 'nombre' not in data or 'cantidad' not in data or 'precio' not in data or 'unidad_medida' not in data:
        return jsonify({'error': 'Datos incompletos'}), 400
    add_producto(data['nombre'], data['cantidad'], data['precio'], data['unidad_medida'])
    return jsonify({'message': 'Producto añadido'}), 201
```

```
def add_producto(nombre, cantidad, precio, unidad_medida):
    db = get_db()
    c = db.cursor()
    c.execute("INSERT INTO productos (nombre, cantidad, precio, unidad_medida) VALUES (?, ?, ?, ?)",
              (nombre, cantidad, precio, unidad_medida))
    db.commit()
```

PUT

04

```
@app.route('/api/productos/<int:id>', methods=['PUT'])
def update_producto_endpoint(id):
    data = request.json
    if 'nombre' not in data or 'cantidad' not in data or 'precio' not in data or 'unidad_medida' not in data:
        return jsonify({'error': 'Datos incompletos'}), 400
    update_producto(id, data['nombre'], data['cantidad'], data['precio'], data['unidad_medida'])
    return jsonify({'message': 'Producto actualizado'})
```

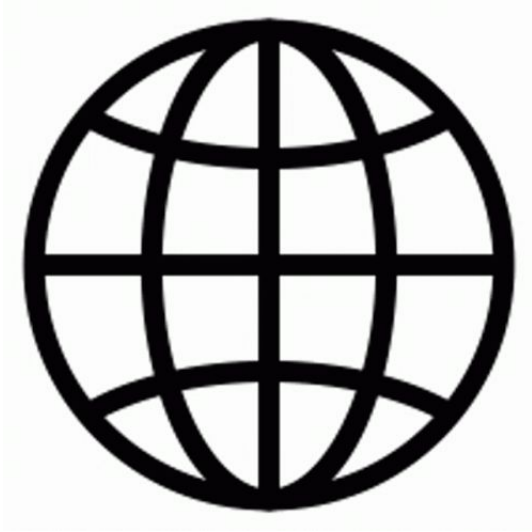
```
def update_producto(id, nombre, cantidad, precio, unidad_medida):
    db = get_db()
    c = db.cursor()
    c.execute("UPDATE productos SET nombre = ?, cantidad = ?, precio = ?, unidad_medida = ? WHERE id = ?",
              (nombre, cantidad, precio, unidad_medida, id))
    db.commit()
```

DELETE

04

```
@app.route('/api/productos/<int:id>', methods=['DELETE'])
def delete_producto_endpoint(id):
    delete_producto(id)
    return jsonify({'message': 'Producto eliminado'}), 200
```

```
def delete_producto(id):
    db = get_db()
    c = db.cursor()
    c.execute("DELETE FROM productos WHERE id = ?", (id,))
    db.commit()
```



FRONTEND WEB

HTML



¿?



¡GRACIAS! :)

ROLDAN, CHARA, TRILLINI y TARGA

