

UNIVERSIDAD DA VINCI DE GUATEMALA
FACULTAD DE INGENIERÍA, INDUSTRIA Y TECNOLOGÍA
CARRERA: LICENCIATURA EN INGENIERÍA EN SISTEMAS



Documentación Inventario SGA
Ingeniería de Software II

Ashley Dayane Alfaro Aguilar

Carné 202301982

Byron Rodolfo Maldonado Palacios

Carné 202300076

Guatemala, 2025.

Creamos entorno virutal y lo activamos

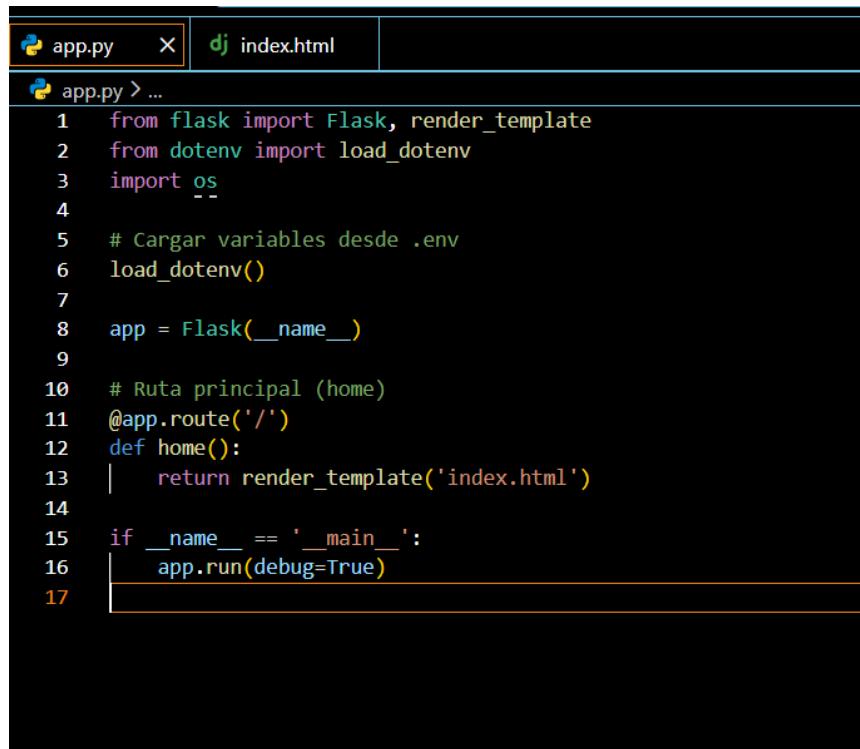
```
Windows PowerShell

PS C:\Users\Lenovo\documents\u\semestre6\inventario_sga> python -m venv venv
PS C:\Users\Lenovo\documents\u\semestre6\inventario_sga> venv\Scripts\activate
(venv) PS C:\Users\Lenovo\documents\u\semestre6\inventario_sga>
```

Instalamos flask y python-dotenv

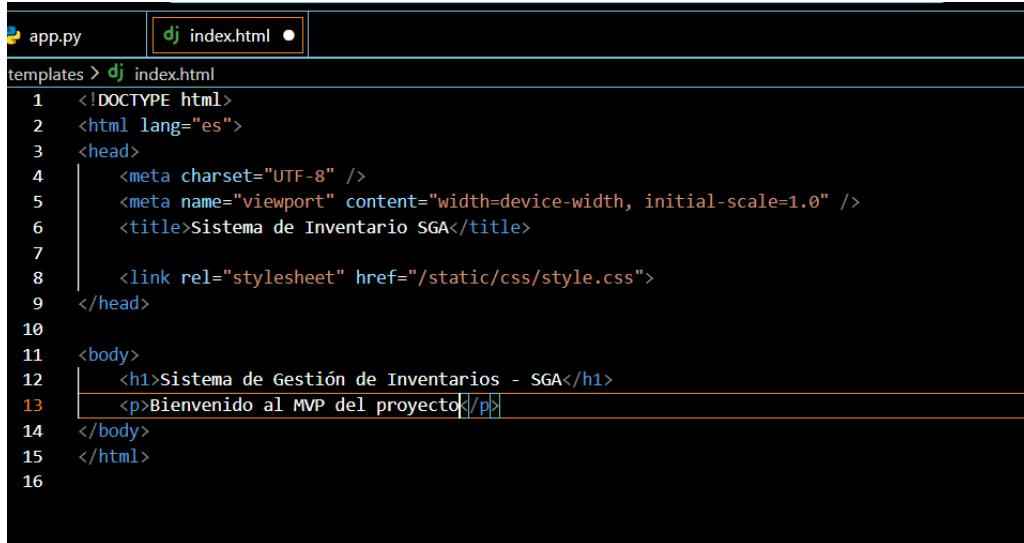
```
(venv) PS C:\Users\Lenovo\documents\u\semestre6\inventario_sga> pip install flask python-dotenv
Collecting flask
  Downloading flask-3.1.2-py3-none-any.whl.metadata (3.2 kB)
Collecting python-dotenv
  Downloading python_dotenv-1.2.1-py3-none-any.whl.metadata (25 kB)
Collecting blinker>=1.9.0 (from flask)
  Downloading blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting click<=8.1.3 (from flask)
  Downloading click-8.3.1-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.2.0 (from flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting jinja2>=3.1.2 (from flask)
  Downloading jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
Collecting markupsafe>=2.1.1 (from flask)
  Downloading markupsafe-3.0.3-cp313-cp313-win_amd64.whl.metadata (2.8 kB)
Collecting werkzeug>=3.1.0 (from flask)
  Downloading werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Collecting colorama (from click>=8.1.3->flask)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Downloading flask-3.1.2-py3-none-any.whl (103 kB)
Downloading python_dotenv-1.2.1-py3-none-any.whl (21 kB)
Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
Downloading click-8.3.1-py3-none-any.whl (108 kB)
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
Downloading markupsafe-3.0.3-cp313-cp313-win_amd64.whl (15 kB)
Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: python-dotenv, markupsafe, itsdangerous, colorama, blinker, werkzeug, jinja2, click, flask
Successfully installed blinker-1.9.0 click-8.3.1 colorama-0.4.6 flask-3.1.2 itsdangerous-2.2.0 jinja2-3.1.6 markupsafe-3.0.3 python-dotenv-1.2.1 werkzeug-3.1.3
[notice] A new release of pip is available: 24.3.1 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Creamos el app.py



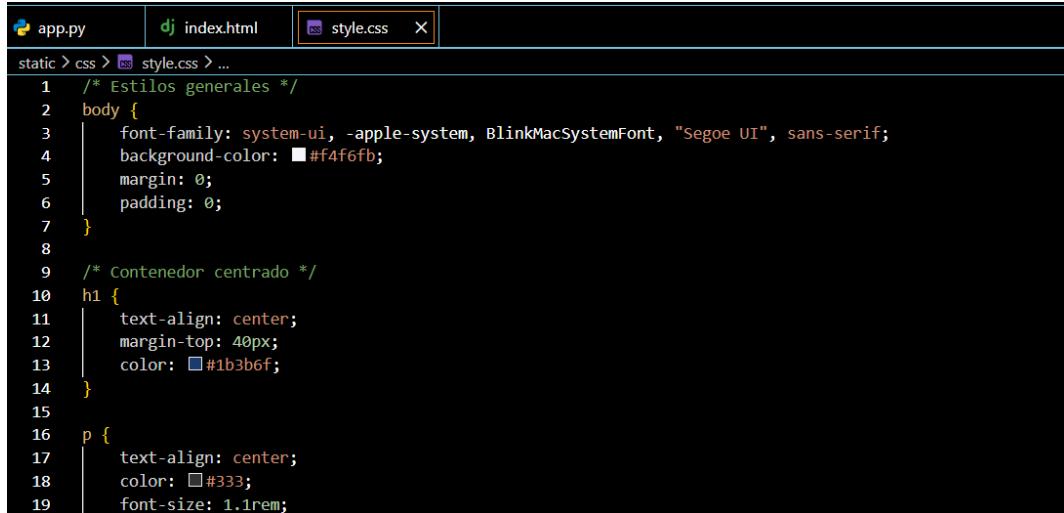
```
app.py > ...
1  from flask import Flask, render_template
2  from dotenv import load_dotenv
3  import os
4
5  # Cargar variables desde .env
6  load_dotenv()
7
8  app = Flask(__name__)
9
10 # Ruta principal (home)
11 @app.route('/')
12 def home():
13     return render_template('index.html')
14
15 if __name__ == '__main__':
16     app.run(debug=True)
17
```

Creamos el index.html



```
app.py | index.html ●
templates > index.html
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>Sistema de Inventario SGA</title>
7
8      <link rel="stylesheet" href="/static/css/style.css">
9  </head>
10
11 <body>
12     <h1>Sistema de Gestión de Inventarios - SGA</h1>
13     <p>Bienvenido al MVP del proyecto</p>
14 </body>
15 </html>
16
```

Creamos Styles.css



```
app.py | index.html | style.css X
static > css > style.css > ...
1  /* Estilos generales */
2  body {
3      font-family: system-ui, -apple-system, BlinkMacSystemFont, "Segoe UI", sans-serif;
4      background-color: #f4f6fb;
5      margin: 0;
6      padding: 0;
7  }
8
9  /* Contenedor centrado */
10 h1 {
11     text-align: center;
12     margin-top: 40px;
13     color: #1b3b6f;
14 }
15
16 p {
17     text-align: center;
18     color: #333;
19     font-size: 1.1rem;
```

Intentamos ejecutar el primer app.py



```

21
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python + v

● PS C:\Users\Lenovo\Documents\U\SEMESTRE6\inventario_sga> & C:/Users/Lenovo/Documents/U/SEMESTRE6/inventario_sga/venv/Scripts/Activate.ps1
○ (venv) PS C:\Users\Lenovo\Documents\U\SEMESTRE6\inventario_sga> python app.py
  * Serving Flask app 'app'
  * Debug mode: on
  WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
  * Running on http://127.0.0.1:5000
Press CTRL+C to quit
  * Restarting with stat
  * Debugger is active!
  * Debugger PIN: 229-602-718
127.0.0.1 - - [24/Nov/2025 23:22:53] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [24/Nov/2025 23:22:53] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [24/Nov/2025 23:22:53] "GET /static/css/style.css HTTP/1.1" 200 -
127.0.0.1 - - [24/Nov/2025 23:22:55] "GET /static/css/style.css HTTP/1.1" 304 -
127.0.0.1 - - [24/Nov/2025 23:22:55] "GET /favicon.ico HTTP/1.1" 404 -

```

Se crea y se prueba conexión a la base de datos

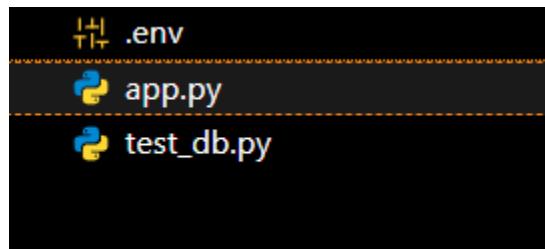
```

● (venv) PS C:\Users\Lenovo\Documents\U\SEMESTRE6\inventario_sga> python test_db.py
Conexión exitosa con SQL Server
○ (venv) PS C:\Users\Lenovo\Documents\U\SEMESTRE6\inventario_sga>

```

+ ↗ Spanish
+ ↗ **InventarioDB**
+ ↗ Security

Con los siguientes archivos hicimos la conexión a la base de datos .env y test_db.py



Creamos la tabla productos en la bd inventariodb

The screenshot shows the SSMS interface with the Object Explorer on the left and a query editor window on the right.

Object Explorer:

- Connected to: LAPTOP-LKNA3HJB (SQL Server 16)
- Databases:
 - System Databases
 - Database Snapshots
 - BancoVivienda
 - Spanish
 - InventarioDB** (selected)
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Products
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Query Store
 - Service Broker
 - Storage
 - Security

SQL Query Editor:

```

USE InventarioDB;
GO

CREATE TABLE Productos (
    IdProducto INT IDENTITY(1,1) PRIMARY KEY,
    Codigo VARCHAR(50) NOT NULL,
    Nombre VARCHAR(150) NOT NULL,
    Categoría VARCHAR(100) NULL,
    Proveedor VARCHAR(150) NULL,
    PrecioCompra DECIMAL(18,2) NOT NULL,
    PrecioVenta DECIMAL(18,2) NOT NULL,
    StockActual INT NOT NULL DEFAULT 0,
    StockMínimo INT NOT NULL DEFAULT 0,
    Activo BIT NOT NULL DEFAULT 1,
    FechaCreación DATETIME NOT NULL DEFAULT GETDATE()
);

```

Messages:

```

Commands completed successfully.
Completion time: 2025-11-24T23:49:52.7113119-06:00

```

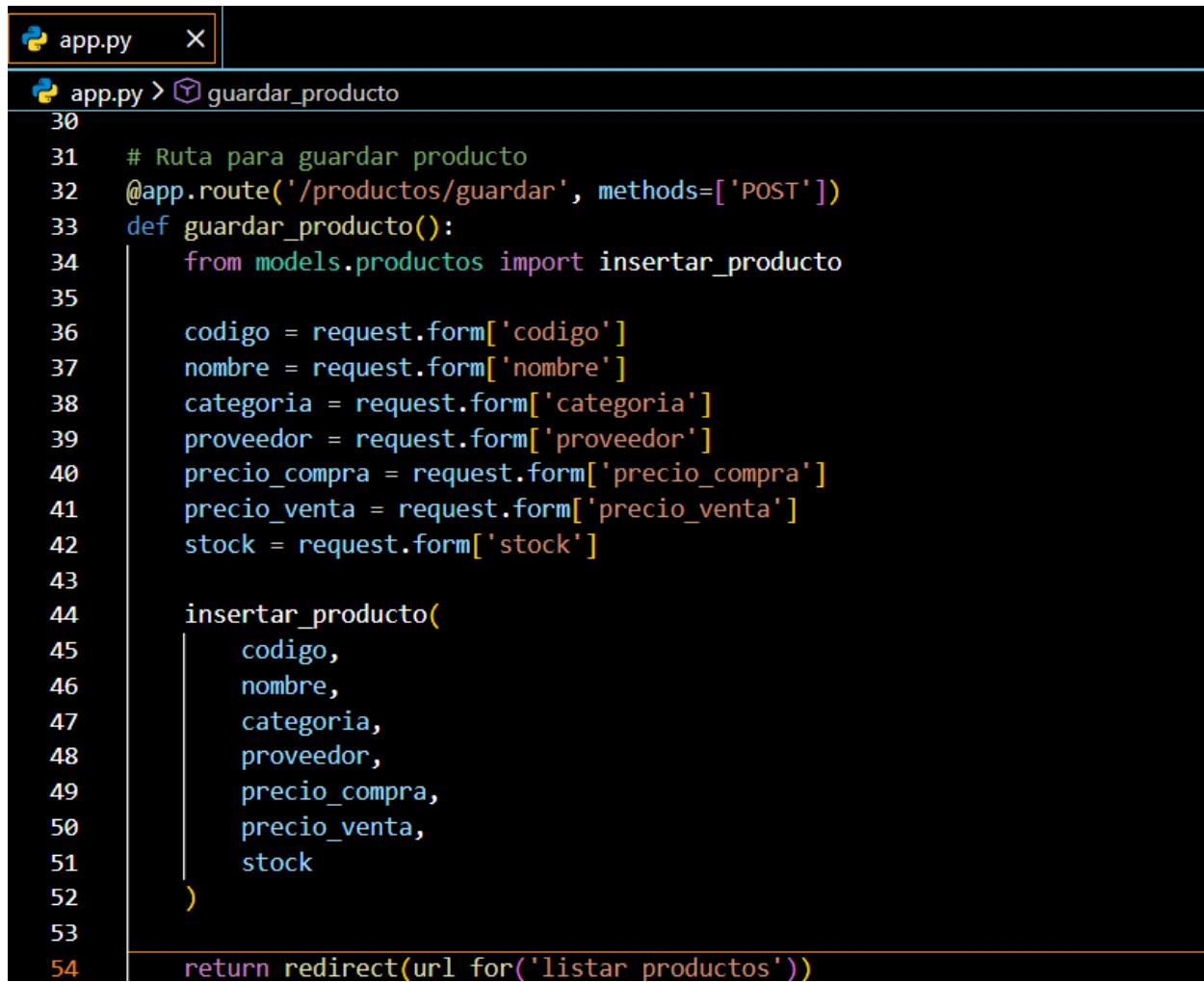
Creamos productos.py que lee los productos de sql

```
models > 🐍 productos.py > ...
1  from .db import get_connection
2
3  def obtener.todos.los.productos():
4      """
5          Devuelve una lista de diccionarios con todos los productos.
6      """
7      productos = []
8
9      conn = get_connection()
10     if not conn:
11         return productos # lista vacía si no hay conexión
12
13     try:
14         cursor = conn.cursor()
15         cursor.execute("""
16             SELECT IdProducto, Código, Nombre, Categoría,
17                 Proveedor, PrecioCompra, PrecioVenta,
18                 StockActual, StockMínimo, Activo, FechaCreación
19             FROM Productos
20             ORDER BY Nombre
21         """)
22
23         columnas = [col[0] for col in cursor.description]
24
25         for fila in cursor.fetchall():
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
Successfully installed pip-25.3
● (venv) PS C:\Users\Lenovo\Documents\U\SEMESTRE6\inventario_sga> python test_db.py
Error conectando a SQL Server: ('28000', '[Microsoft][ODBC Driver 17 for SQL Server][SQL Server]')
```

Se actualiza app.py para que sepa que existe un apartado que se llama productos

```
terminal   Help   ← →   inventario_sga
... 🐍 app.py > ...
app.py > ...
1  from flask import Flask, render_template
2  from dotenv import load_dotenv
3  import os
4
5  # Importar la función para traer productos
6  from models.productos import obtener.todos.los.productos
7
8  # Cargar variables desde .env
9  load_dotenv()
10
11 app = Flask(__name__)
12
13 # Ruta principal (home)
14 @app.route('/')
15 def home():
16     return render_template('index.html')
17
18 # Nueva ruta: listado de productos
19 @app.route('/productos')
20 def listar_productos():
21     productos = obtener.todos.los.productos()
22     return render_template('productos.html', productos=productos)
23
24 if __name__ == '__main__':
25     app.run(debug=True)
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
```

Se actualizan las rutas



```
app.py | guardar_producto
30
31     # Ruta para guardar producto
32     @app.route('/productos/guardar', methods=['POST'])
33     def guardar_producto():
34         from models.productos import insertar_producto
35
36         codigo = request.form['codigo']
37         nombre = request.form['nombre']
38         categoria = request.form['categoria']
39         proveedor = request.form['proveedor']
40         precio_compra = request.form['precio_compra']
41         precio_venta = request.form['precio_venta']
42         stock = request.form['stock']
43
44         insertar_producto(
45             codigo,
46             nombre,
47             categoria,
48             proveedor,
49             precio_compra,
50             precio_venta,
51             stock
52         )
53
54     return redirect(url_for('listar_productos'))
```

Creamos la página de productos



Agregamos la función de insertar producto

app.py productos.py X

models > app.py > productos.py > insertar_producto

```
35
36     def insertar_producto(codigo, nombre, categoria, proveedor,
37     |   |   |   |   | precio_compra, precio_venta, stock):
38     |   |
39     |   Inserta un nuevo producto en la tabla Productos.
40     |   |
41     |   conn = get_connection()
42     |   if not conn:
43     |       print("No hay conexión a la base de datos.")
44     |       return
45
46     try:
47         cursor = conn.cursor()
48         cursor.execute("""
49             INSERT INTO Productos
50                 (Codigo, Nombre, Categoria, Proveedor,
51                  | PrecioCompra, PrecioVenta, StockActual, StockMinimo, Activo)
52             VALUES
53                 (?, ?, ?, ?, ?, ?, ?, ?, 0, 1)
54             """, [
55                 | codigo,
56                 | nombre,
57                 | categoria,
58                 | proveedor,
59                 | float(precio_compra),
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Creamos el html para agregar un nuevo producto

The screenshot shows a web browser window with the URL `127.0.0.1:5000/productos/nuevo`. The page title is "Nuevo Producto". The form contains the following fields:

- Código:
- Nombre:
- Categoría:
- Proveedor:
- Precio de compra:
- Precio de venta:
- Stock inicial:

At the bottom of the form are two buttons: "Guardar producto" (in a dark blue bar) and "[Volver al listado](#)".

Probamos agregar un nuevo producto y ver si ya esta reflejado en la tabla de productos

Nuevo Producto

Código

120132

Nombre

Aire Comprimido con Trigger

Categoría

Limpieza

Proveedor

etouch

Precio de compra

40

Precio de venta

55

Stock inicial

20

Guardar producto[← Volver al listado](#)

Listado de Productos

ID	Código	Nombre	Categoría	Proveedor	Precio Compra	Precio Venta	Stock	Estado
1	120132	Aire Comprimido con Trigger	Limpieza	etouch	Q 40.00	Q 55.00	20	Activo


```
use InventoryDB;
go
select * from dbo.Productos;
```

Results

IdProducto	Código	Nombre	Categoría	Proveedor	PrecioCompra	PrecioVenta	StockActual	StockMinimo	Activo	FechaCreacion
1	120132	Aire Comprimido con Trigger	Limpieza	etouch	40.00	55.00	20	0	1	2025-11-26 18:13:00.283

Producto insertado correctamente.

```
127.0.0.1 - - [26/Nov/2025 18:24:48] "POST /productos/guardar HTTP/1.1" 302 -
127.0.0.1 - - [26/Nov/2025 18:24:48] "GET /productos HTTP/1.1" 200 -
127.0.0.1 - - [26/Nov/2025 18:24:48] "GET /static/css/style.css HTTP/1.1" 304 -
```

Se agrega la función de editar un producto

```
app.py | productos.py X | productos.html | nuevo_producto.htm

models > productos.py > ...
106  def actualizar_producto(id_producto, codigo, nombre, categoria,
107      precio_compra, precio_venta, stock):
108      """
109          Actualiza los datos de un producto existente.
110      """
111      conn = get_connection()
112      if not conn:
113          print("No hay conexión a la base de datos.")
114          return
115
116      try:
117          cursor = conn.cursor()
118          cursor.execute("""
119              UPDATE Productos
120                  SET Código = ?,
121                      Nombre = ?,
122                      Categoría = ?,
123                      Proveedor = ?,
124                      PrecioCompra = ?,
125                      PrecioVenta = ?,
126                      StockActual = ?
127              WHERE IdProducto = ?
128          """, (

```

Y se agregan las rutas

```
app.py > ...
62     # Ruta para mostrar el formulario de edición
63     @app.route('/productos/editar/<int:id_producto>', methods=['GET'])
64     def editar_producto(id_producto):
65         producto = obtener_producto_por_id(id_producto)
66         if not producto:
67             # Si no existe, vuelve al listado
68             return redirect(url_for('listar_productos'))
69         return render_template('editar_producto.html', producto=producto)
70
71     # Ruta para actualizar los datos en la BD
72     @app.route('/productos/actualizar', methods=['POST'])
73     def actualizar_producto_route():
74         id_producto = request.form['id_producto']
75         codigo = request.form['codigo']
76         nombre = request.form['nombre']
77         categoria = request.form['categoria']
78         proveedor = request.form['proveedor']
79         precio_compra = request.form['precio_compra']
80         precio_venta = request.form['precio_venta']
81         stock = request.form['stock']
82
83         actualizar_producto(
84             id_producto,
85             codigo,
```

Agregamos pagina de editar producto

```
templates > dj editar_producto.html
65 <body>
66     <div class="form-container">
67         <h2>Editar Producto</h2>
68
69         <form action="/productos/actualizar" method="POST">
70             <!-- ID oculto -->
71             <input type="hidden" name="id_producto" value="{{ producto.IdProducto }}>
72
73             <label for="codigo">Código</label>
74             <input type="text" id="codigo" name="codigo" value="{{ producto.Codigo }}" required>
75
76             <label for="nombre">Nombre</label>
77             <input type="text" id="nombre" name="nombre" value="{{ producto.Nombre }}" required>
78
79             <label for="categoria">Categoria</label>
80             <input type="text" id="categoria" name="categoria" value="{{ producto.Categoria }}>
81
82             <label for="proveedor">Proveedor</label>
83             <input type="text" id="proveedor" name="proveedor" value="{{ producto.Proveedor }}>
84
85             <label for="precio_compra">Precio de compra</label>
86             <input type="number" step="0.01" id="precio_compra" name="precio_compra"
87                 value="{{ producto.PrecioCompra }}" required>
88
89         </form>
```

Agregamos el botón de editar

Sistema de Inventario SGA | Listado de Productos | 127.0.0.1:5000/productos

Listado de Productos

ID	Código	Nombre	Categoría	Proveedor	Precio Compra	Precio Venta	Stock	Estado	Acciones
1	120132	Aire Compresido con Trigger	Limpieza	etouch	Q 40.00	Q 55.00	20	Activo	<button>Editar</button>
2	258794	Contact Cleaner	Limpieza	GX industry	Q 40.00	Q 55.00	22	Activo	<button>Editar</button>
3	1234567	Kit de reparación	Herramientas	TRUPER	Q 90.00	Q 115.00	36	Activo	<button>Editar</button>
4	654893	Lenovo LEGION	Computadoras	Lenovo	Q 12000.00	Q 15000.00	2	Activo	<button>Editar</button>

ID	Código	Nombre	Categoría	Proveedor	Precio Compra	Precio Venta	Stock	Estado	Acciones
1	120132	Aire Comprimido con Trigger	Limpieza	etouch	Q 40.00	Q 55.00	10	Activo	<button>Editar</button>
2	258794	Contact Cleaner	Limpieza	GX industry	Q 40.00	Q 55.00	22	Activo	<button>Editar</button>
3	1234567	Kit de reparación	Herramientas	TRUPER	Q 90.00	Q 115.00	36	Activo	<button>Editar</button>

Results Messages

IdProducto	Código	Nombre	Categoría	Proveedor	PrecioCompra	PrecioVenta	StockActual	StockMinimo	Activo	FechaCreacion
1	120132	Aire Comprimido con Trigger	Limpieza	etouch	40.00	55.00	10	0	1	2025-11-26 18:13:00.283

```
127.0.0.1 - - [26/Nov/2025 18:59:59] "GET /static/css/style.css HTTP/1.1" 304 -
Producto actualizado correctamente.
127.0.0.1 - - [26/Nov/2025 19:00:04] "POST /productos/actualizar HTTP/1.1" 302 -
127.0.0.1 - - [26/Nov/2025 19:00:04] "GET /productos HTTP/1.1" 200 -
127.0.0.1 - - [26/Nov/2025 19:00:04] "GET /static/css/style.css HTTP/1.1" 304 -
```

Agregamos función y la ruta de eliminar producto

```
models > productos.py > eliminar_producto
145     conn.close()
146
147
148 def eliminar_producto(id_producto):
149     """
150         Elimina (o marca como inactivo) un producto por su IdProducto.
151     """
152     conn = get_connection()
153     if not conn:
154         print("No hay conexión a la base de datos.")
155         return
156
157     try:
158         cursor = conn.cursor()
159
160         # Borrado lógico
161         cursor.execute("""
162             UPDATE Productos
163             SET Activo = 0
164             WHERE IdProducto = ?
165             """, (int(id_producto),))
166
167         conn.commit()
168         print("Producto eliminado (marcado como inactivo).")
169     
```

```

@app.route('/productos/eliminar/<int:id_producto>', methods=['GET'])
def eliminar_producto_route(id_producto):
    eliminar_producto(id_producto)
    return redirect(url_for('listar_productos'))

```

Agregamos botón de eliminar

```

        </a>
        <a href="/productos/eliminar/{{ p.IdProducto }}"
            onclick="return confirm('¿Seguro que deseas eliminar este producto?');"
            style="padding: 4px 8px; background-color: #dc3545; color: white; text-decoration: none; border-radius: 4px;">Eliminar
    </a>
</td>

```

+ Nuevo producto									
ID	Código	Nombre	Categoría	Proveedor	Precio Compra	Precio Venta	Stock	Estado	Acciones
1	120132	Aire Comprimido con Trigger	Limpieza	etouch	Q 40.00	Q 55.00	10	Activo	<button>Editar</button> <button>Eliminar</button>
2	258794	Contact Cleaner	Limpieza	GX industry	Q 40.00	Q 55.00	22	Activo	<button>Editar</button> <button>Eliminar</button>
3	1234567	Kit de reparación	Herramientas	TRUPER	Q 90.00	Q 115.00	36	Activo	<button>Editar</button> <button>Eliminar</button>
4	654893	Lenovo LEGION	Computadoras	Lenovo	Q 12000.00	Q 15000.00	2	Activo	<button>Editar</button> <button>Eliminar</button>

probamos su función

127.0.0.1:5000 dice

¿Seguro que deseas eliminar este producto?

Aceptar
Cancelar

Categoría	Proveedor	Precio Compra	Precio Venta	Stock	Estado
Limpieza	etouch	Q 40.00	Q 55.00	10	Activo

IdProducto Código Nombre Categoría Proveedor PrecioCompra PrecioVenta StockActual StockMinimo Activo FechaCreacion

1	120132	Aire Comprimido con Trigger	Limpieza	etouch	40.00	55.00	10	0	1	2025-11-26 18:13:00.283
2	258794	Contact Cleaner	Limpieza	GX industry	40.00	55.00	22	0	1	2025-11-26 18:19:46.303
3	1234567	Kit de reparación	Herramientas	TRUPER	90.00	115.00	36	0	1	2025-11-26 18:21:09.777
4	654893	Lenovo LEGION	Computadoras	Lenovo	12000.00	15000.00	2	0	0	2025-11-26 18:24:48.810

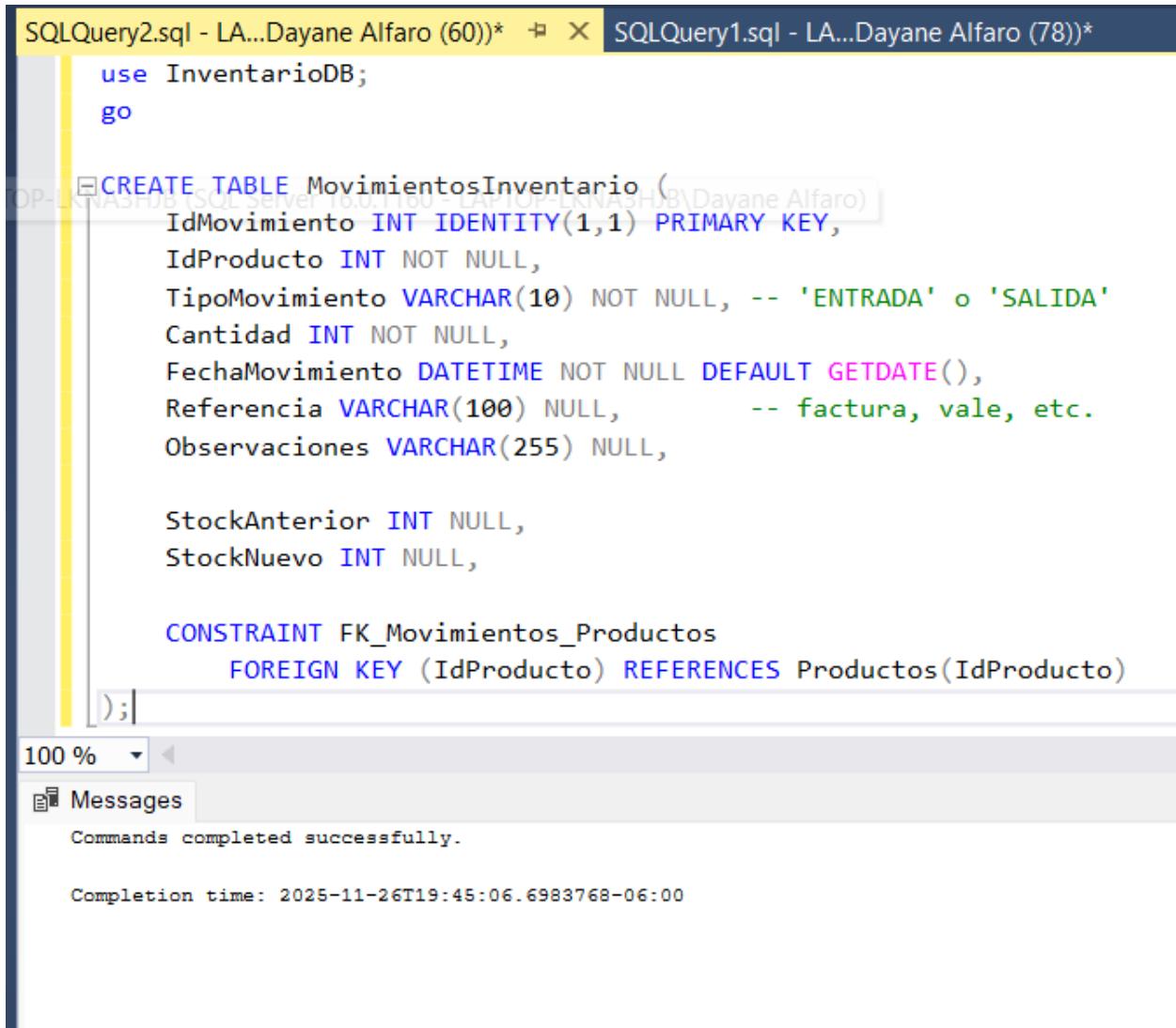
```

[26/Nov/2025 19:41:07] "GET /productos/eliminar/4 HTTP/1.1" 302 -
[26/Nov/2025 19:41:07] "GET /productos HTTP/1.1" 200 -

```

Listado de Productos									
+ Nuevo producto									
ID	Código	Nombre	Categoría	Proveedor	Precio Compra	Precio Venta	Stock	Estado	Acciones
1	120132	Aire Comprimido con Trigger	Limpieza	etouch	Q 40.00	Q 55.00	10	Activo	Editar Eliminar
2	258794	Contact Cleaner	Limpieza	GX industry	Q 40.00	Q 55.00	22	Activo	Editar Eliminar
3	1234567	Kit de reparación	Herramientas	TRUPER	Q 90.00	Q 115.00	36	Activo	Editar Eliminar
4	654893	Lenovo LEGION	Computadoras	Lenovo	Q 12000.00	Q 15000.00	2	Inactivo	Editar Eliminar

Creamos la tabla para los movimientos del inventario



```

SQLQuery2.sql - LA...Dayane Alfaro (60))*
SQLQuery1.sql - LA...Dayane Alfaro (78)*

use InventarioDB;
go

CREATE TABLE MovimientosInventario (
    IdMovimiento INT IDENTITY(1,1) PRIMARY KEY,
    IdProducto INT NOT NULL,
    TipoMovimiento VARCHAR(10) NOT NULL, -- 'ENTRADA' o 'SALIDA'
    Cantidad INT NOT NULL,
    FechaMovimiento DATETIME NOT NULL DEFAULT GETDATE(),
    Referencia VARCHAR(100) NULL, -- factura, vale, etc.
    Observaciones VARCHAR(255) NULL,

    StockAnterior INT NULL,
    StockNuevo INT NULL,

    CONSTRAINT FK_Movimientos_Productos
        FOREIGN KEY (IdProducto) REFERENCES Productos(IdProducto)
);

```

100 %

Messages

Commands completed successfully.

Completion time: 2025-11-26T19:45:06.6983768-06:00

Creamos el módulo de movimientos del inventario y agregamos la ruta de movimientos a app.py

```
app.py          movimientos.py X
models > movimientos.py > registrar_movimiento
1  from .db import get_connection
2
3  def registrar_movimiento(id_producto, tipo_movimiento, cantidad,
4  |  |  |  |  |  | referencia=None, observaciones=None):
5      """
6          Registra un movimiento de inventario (ENTRADA o SALIDA) y actualiza el stock del producto.
7      """
8
9      conn = get_connection()
10     if not conn:
11         print("No hay conexión a la base de datos.")
12         return False
13
14     try:
15         cursor = conn.cursor()
16
17         # 1. Obtener stock actual del producto
18         cursor.execute("""
19             SELECT StockActual
20             FROM Productos
21             WHERE IdProducto = ?
22         """, (int(id_producto),))
23
24         fila = cursor.fetchone()
25         if not fila:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Creamos el html de movimientos

```
app.py          nuevo_movimiento.html ●  movimientos.py
templates > nuevo_movimiento.html
80  <body>
81
82  <div class="form-container">
83      <h2>Nuevo Movimiento</h2>
84
85      <form action="/movimientos/guardar" method="POST">
86
87          <label for="id_producto">Producto</label>
88          <select id="id_producto" name="id_producto" required>
89              <option value="">-- Seleccione un producto --</option>
90              {% for p in productos %}
91                  <option value="{{ p.IdProducto }}>{{ p.Codigo }} - {{ p.Nombre }}

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS


```

Ario SGA | Listado de Productos | Nuevo Movimiento | +

127.0.0.1:5000/movimientos/nuevo

Nuevo Movimiento

Producto
-- Seleccione un producto --

Tipo de movimiento
 Entrada
 Salida

Cantidad

Referencia (factura, vale, etc.)

Observaciones

Registrar movimiento

[← Volver al listado de productos](#)

Nuevo Movimiento

Producto
120132 - Aire Comprimido con Trigger

Tipo de movimiento
 Entrada
 Salida

Cantidad
 5

Referencia (factura, vale, etc.)
 Factura

Observaciones
 esto es una prueba

Registrar movimiento

ID	Código	Nombre	Categoría	Proveedor	Precio Compra	Precio Venta	Stock	Estado	Acciones
1	120132	Aire Comprimido con Trigger	Limpieza	etouch	Q 40.00	Q 55.00	15	Activo	<button>Editar</button> <button>Eliminar</button>
2	258794	Contact Cleaner	Limpieza	GX industry	Q 40.00	Q 55.00	22	Activo	<button>Editar</button> <button>Eliminar</button>
3	1234567	Kit de reparación	Herramientas	TRUPER	Q 90.00	Q 115.00	36	Activo	<button>Editar</button> <button>Eliminar</button>
4	654893	Lenovo LEGION	Computadoras	Lenovo	Q 12000.00	Q 15000.00	2	Inactivo	<button>Editar</button> <button>Eliminar</button>

Results Messages											
IdProducto	Codigo	Nombre	Categoría	Proveedor	PrecioCompra	PrecioVenta	StockActual	StockMinimo	Activo	FechaCreacion	
1	1	120132	Aire Comprimido con Trigger	Limpieza	etouch	40.00	55.00	15	0	1	2025-11-26 18:13:00.283
2	2	258794	Contact Cleaner	Limpieza	GX industry	40.00	55.00	22	0	1	2025-11-26 18:19:46.303

Nuevo Movimiento

Producto

120132 - Aire Comprimido con Trigger

Tipo de movimiento

Entrada

Salida

Cantidad

5

Referencia (factura, vale, etc.)

Factura

Observaciones

esto es una prueba

Registrar movimiento

ID	Código	Nombre	Categoría	Proveedor	Precio Compra	Precio Venta	Stock	Estado	Acciones
1	120132	Aire Comprimido con Trigger	Limpieza	etouch	Q 40.00	Q 55.00	10	Activo	<button>Editar</button> <button>Eliminar</button>

```
Movimiento registrado correctamente.  
127.0.0.1 - - [26/Nov/2025 20:11:08] "POST /movimientos/guardar HTTP/1.1" 302 -  
127.0.0.1 - - [26/Nov/2025 20:11:08] "GET /productos HTTP/1.1" 200 -  
127.0.0.1 - - [26/Nov/2025 20:11:08] "GET /static/css/style.css HTTP/1.1" 304 -
```

Agregamos botón de movimiento

The screenshot shows a table with four rows of movement data. Each row contains three buttons: 'Editar' (Edit), 'Movimiento' (Movement), and 'Eliminar' (Delete). The 'Movimiento' button is highlighted in blue, while the others are yellow.

Acciones
Editar Movimiento Eliminar
Editar Movimiento Eliminar
Editar Movimiento Eliminar

Agregamos la función de obtener movimientos de un producto

```
75 |     conn.close()  
76 |  
77 |  
78 |  
79 |  
80 v def obtener_movimientos_por_producto(id_producto):  
81 v     """  
82     Devuelve una lista de movimientos de inventario para un producto específico.  
83     """  
84     conn = get_connection()  
85     movimientos = []  
86  
87 v     if not conn:  
88         print("No hay conexión a la base de datos.")  
89         return movimientos  
90  
91 v     try:  
92         cursor = conn.cursor()  
93 v         cursor.execute("""  
94             SELECT  
95                 m.IdMovimiento,  
96                 m.TipoMovimiento,  
97                 m.Cantidad,  
98                 m.FechaMovimiento,  
99                 m.Referencia,
```

Agregamos la ruta en app.py

```
# Historial de movimientos de un producto
@app.route('/movimientos/producto/<int:id_producto>', methods=['GET'])
def movimientos_por_producto(id_producto):
    producto = obtener_producto_por_id(id_producto)
    if not producto:
        return redirect(url_for('listar_productos'))

    movimientos = obtener_movimientos_por_producto(id_producto)
    return render_template(
        'movimientos_producto.html',
        producto=producto,
        movimientos=movimientos
    )
```

Creamos el html para los movimientos



The screenshot shows a code editor with three tabs at the top: 'app.py', 'movimientos_producto.html' (which is selected and highlighted with a yellow border), and 'movimientos.py'. The main area displays the content of the 'movimientos_producto.html' template.

```
templates > dj movimientos_producto.html
48  <body>
49
50      <h2>Historial de Movimientos</h2>
51
52      <div class="info-producto">
53          <h3>{{ producto.Codigo }} - {{ producto.Nombre }}</h3>
54          <p>Stock actual: <strong>{{ producto.StockActual }}</strong></p>
55      </div>
56
57      <table>
58          <thead>
59              <tr>
60                  <th>ID Mov.</th>
61                  <th>Tipo</th>
62                  <th>Cantidad</th>
63                  <th>Stock Anterior</th>
64                  <th>Stock Nuevo</th>
65                  <th>Fecha</th>
66                  <th>Referencia</th>
67                  <th>Observaciones</th>
68          </tr>
69      </thead>
```

Sistema de Inventario SGA | Listado de Productos | Movimientos de Aire Comprimido

127.0.0.1:5000/movimientos/producto/1

Historial de Movimientos

120132 - Aire Comprimido con Trigger

Stock actual: 15

ID Mov.	Tipo	Cantidad	Stock Anterior	Stock Nuevo	Fecha	Referencia	Observaciones
3	ENTRADA	5	10	15	2025-11-26 20:11:08.553000	Factura	esto es una prueba
2	SALIDA	5	15	10	2025-11-26 20:09:39.720000	Factura	esto es una prueba
1	ENTRADA	5	10	15	2025-11-26 20:08:07.430000	Factura	esto es una prueba

[--> Volver al listado de productos](#)

100 %

Results Messages

IdMovimiento	IdProducto	TipoMovimiento	Cantidad	FechaMovimiento	Referencia	Observaciones	StockAnterior	StockNuevo
1	1	ENTRADA	5	2025-11-26 20:08:07.430	Factura	esto es una prueba	10	15
2	2	SALIDA	5	2025-11-26 20:09:39.720	Factura	esto es una prueba	15	10
3	3	ENTRADA	5	2025-11-26 20:11:08.553	Factura	esto es una prueba	10	15

Agregamos botón de movimientos

Estado **Acciones**

Activo	Editar	Movimiento	Movimientos	Eliminar
Activo	Editar	Movimiento	Movimientos	Eliminar
Activo	Editar	Movimiento	Movimientos	Eliminar
Inactivo	Editar	Movimiento	Movimientos	Eliminar

```
<a href="/movimientos producto/{{ p.IdProducto }}"
style="padding: 4px 8px; background-color: #20c997; color: white; text-decoration: none; border-radius: 4px">
    Movimientos
</a>
```

Creamos el login, pero primero creamos las tablas de usuarios

```

SQLQuery3.sql - LA...Dayane Alfaro (68)*  X  MovimientosInventa...ayane Alfaro (60))      SQLQuery1
use InventarioDB;
go

CREATE TABLE Usuarios (
    IdUsuario INT IDENTITY(1,1) PRIMARY KEY,
    Username VARCHAR(50) NOT NULL UNIQUE,
    Password VARCHAR(255) NOT NULL,
    NombreCompleto VARCHAR(150) NOT NULL,
    Rol VARCHAR(20) NOT NULL, -- ADMIN, OPERADOR, SUPERVISOR
    Activo BIT NOT NULL DEFAULT 1,
    FechaCreacion DATETIME NOT NULL DEFAULT GETDATE()
);

-- Usuario admin inicial (solo para pruebas)
INSERT INTO Usuarios (Username, Password, NombreCompleto, Rol, Activo)
VALUES ('admin', 'admin123', 'Administrador General', 'ADMIN', 1);

```

100 % ▾

Messages

(1 row affected)

Completion time: 2025-11-26T20:56:30.7054464-06:00

select * from apo.usuarios;

100 % ▾

Results Messages

	IdUsuario	Username	Password	NombreCompleto	Rol	Activo	FechaCreacion
1	1	admin	admin123	Administrador General	ADMIN	1	2025-11-26 20:56:30.700

Creamos el modulo para los usuarios

```
app.py | 🐍 usuarios.py X
models > 🐍 usuarios.py > ...
1  from .db import get_connection
2
3  def validar_usuario(username, password):
4      """
5          Valida las credenciales del usuario.
6          Devuelve un diccionario con los datos del usuario si son correctos,
7          o None si no coinciden.
8      """
9      conn = get_connection()
10     if not conn:
11         print("No hay conexión a la base de datos.")
12         return None
13
14    try:
15        cursor = conn.cursor()
16        cursor.execute("""
17            SELECT IdUsuario, Username, NombreCompleto, Rol, Activo
18            FROM Usuarios
19            WHERE Username = ? AND Password = ? AND Activo = 1
20        """, (username, password))
21
22        fila = cursor.fetchone()
23        if fila:
24            [columnas] [selección] [selección]
```

Creamos el login.html

The screenshot shows a web browser window with the following details:

- Header:** Shows the URL "127.0.0.1:5000/login".
- Navigation:** Includes links for "Listado de Productos" and "Iniciar sesión".
- Content:** A modal dialog titled "Sistema de Inventarios" containing a login form.
 - Usuario:** A text input field.
 - Contraseña:** A text input field.
 - Ingresar:** A blue button at the bottom right of the form.



Sistema de Inventarios

Usuario

Contraseña

Ingresar

Usuario o contraseña incorrectos.

Agregamos mensaje de bienvenida y agregamos botones para cerrar sesión y para ver tabla de productos

The screenshot shows a terminal window with two tabs: 'app.py' and 'index.html'. The 'index.html' tab is active, displaying the following code:

```
templates > index.html
59  <body>
61      <div class="home-container">
64          <p>
66              <strong>
67                  | {{ session.get('usuario_nombre', 'Usuario') }}
68              </strong>
69          </p>
71          <p>
72              Rol:
73              <strong>
74                  | {{ session.get('usuario_rol', 'SIN ROL') }}
75              </strong>
76          </p>
77
78          <p style="margin-top: 15px;">
79              | Use las opciones siguientes para gestionar productos y movimientos de inventario.
80          </p>
81
82          <div class="btn-group">
83              <a href="/productos" class="btn-link btn-primary">
84                  | Ver listado de productos
85              </a>
86
87              <a href="/logout" class="btn-link btn-secondary">
```

Below the tabs, there are buttons for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected), and 'PORTS'. On the right side of the terminal window, there is a small icon and the text 'py'.



Creamos nuevos usuarios

	IdUsuario	Username	Password	NombreCompleto	Rol	Activo	FechaCreacion
1	1	admin	admin123	Administrador General	ADMIN	1	2025-11-26 20:56:30.700
2	2	Operador	oper123	Operador	OPERADOR	1	2025-11-26 21:48:41.943
3	3	supervisor	sup123	Supervisor General	SUPERVISOR	1	2025-11-26 21:58:47.053

Aplicamos cambios para que solo supervisor y admin tengan acceso a las modificaciones de los productos

```
7
8     def rol_requerido(*roles_permitidos):
9         def decorador(vista_func):
10            @wraps(vista_func)
11            def wrapper(*args, **kwargs):
12                if 'usuario_id' not in session:
13                    return redirect(url_for('login'))
14
15                rol_usuario = session.get('usuario_rol', '').upper()
16
17                if rol_usuario not in [r.upper() for r in roles_permitidos]:
18                    # Si no tiene rol permitido, lo mandamos al home
19                    return redirect(url_for('home'))
20
21                return vista_func(*args, **kwargs)
22            return wrapper
23        return decorador
24
```

```
3     @app.route('/productos/eliminar/<int:id_producto>', methods=['GET'])
4     @login_requerido
5     @rol_requerido(['ADMIN', 'SUPERVISOR'])
6     def eliminar_producto_route(id_producto):
7         eliminar_producto(id_producto)
8
9         {{ ACTIVO if p.Activo == 1 else inactivo }}</td>
10        <td>
11            {% if session.get('usuario_rol') in ['ADMIN', 'SUPERVISOR'] %}
12            <a href="/productos/editar/{{ p.IdProducto }}"
13                style="padding: 4px 8px; background-color: #ffc107; color: black; text-decoration: none;">Editar
14            </a>
15            {% endif %}
```

Entrando con operador se ve de la siguiente manera

ID	Código	Nombre	Categoría	Proveedor	Precio Compra	Precio Venta	Stock	Estado	Acciones
1	120132	Aire Comprimido con Trigger	Limpieza	etouch	Q 40.00	Q 55.00	15	Activo	<button>Movimiento</button> <button>Movimientos</button>
2	258794	Contact Cleaner	Limpieza	GX industry	Q 40.00	Q 55.00	22	Activo	<button>Movimiento</button> <button>Movimientos</button>
3	1234567	Kit de reparación	Herramientas	TRUPER	Q 90.00	Q 115.00	36	Activo	<button>Movimiento</button> <button>Movimientos</button>
4	654893	Lenovo LEGION	Computadoras	Lenovo	Q 12000.00	Q 15000.00	2	Inactivo	<button>Movimiento</button> <button>Movimientos</button>

Entrando como admin se ve de la siguiente forma

ID	Código	Nombre	Categoría	Proveedor	Precio Compra	Precio Venta	Stock	Estado	Acciones
1	120132	Aire Comprimido con Trigger	Limpieza	etouch	Q 40.00	Q 55.00	15	Activo	<button>Editar</button> <button>Movimiento</button> <button>Movimientos</button> <button>Eliminar</button>
2	258794	Contact Cleaner	Limpieza	GX industry	Q 40.00	Q 55.00	22	Activo	<button>Editar</button> <button>Movimiento</button> <button>Movimientos</button> <button>Eliminar</button>
3	1234567	Kit de reparación	Herramientas	TRUPER	Q 90.00	Q 115.00	36	Activo	<button>Editar</button> <button>Movimiento</button> <button>Movimientos</button> <button>Eliminar</button>
4	654893	Lenovo LEGION	Computadoras	Lenovo	Q 12000.00	Q 15000.00	2	Inactivo	<button>Editar</button> <button>Movimiento</button> <button>Movimientos</button> <button>Eliminar</button>

Reportes productos con stock bajo

Productos con Stock Bajo

No se encontraron productos con stock bajo. El inventario se encuentra dentro de los niveles esperados.

[← Volver al listado de productos](#)

Listado de Productos

[+ Nuevo producto](#) [Ver productos con stock bajo](#)

ID	Código	Nombre	Categoría	Proveedor	Precio Compra	Precio Venta	Stock	Estado	Acciones
1	120132	Aire Comprimido con Trigger	Limpieza	etouch	Q 40.00	Q 55.00	15	Activo	<button>Editar</button> <button>Movimiento</button> <button>Movimientos</button> <button>Eliminar</button>
2	258794	Contact Cleaner	Limpieza	GX industry	Q 40.00	Q 55.00	22	Activo	<button>Editar</button> <button>Movimiento</button> <button>Movimientos</button> <button>Eliminar</button>
3	1234567	Kit de reparación	Herramientas	TRUPER	Q 90.00	Q 115.00	36	Activo	<button>Editar</button> <button>Movimiento</button> <button>Movimientos</button> <button>Eliminar</button>
4	654893	Lenovo LEGION	Computadoras	Lenovo	Q 12000.00	Q 15000.00	2	Inactivo	<button>Editar</button> <button>Movimiento</button> <button>Movimientos</button> <button>Eliminar</button>

Results		Messages									
	IdProducto	Código	Nombre	Categoría	Proveedor	PrecioCompra	PrecioVenta	StockActual	StockMinimo	Activo	FechaCreacion
1	1	120132	Aire Comprimido con Trigger	Limpieza	etouch	40.00	55.00	15	8	1	2025-11-26 18:13:00.283
2	2	258794	Contact Cleaner	Limpieza	GX industry	40.00	55.00	22	10	1	2025-11-26 18:19:46.303
3	3	1234567	Kit de reparación	Herramientas	TRUPER	90.00	115.00	36	5	1	2025-11-26 18:21:09.777
4	4	654893	Lenovo LEGION	Computadoras	Lenovo	12000.00	15000.00	2	2	0	2025-11-26 18:24:48.810