Sistema de Inventario en Django

Alumno: Richard Contreras

Profesora: Carla Bravo

Asignatura: Back End

Fecha: 24-09-2025

# 1. Diseño de Modelos y Relaciones

El sistema de inventario fue diseñado utilizando Django con base de datos PostgreSQL. Se implementaron los siguientes modelos principales:

- Categoría: agrupa productos en el inventario.

- Proveedor: almacena información de proveedores.

- Cliente: permite registrar las ventas a usuarios finales.

- Producto: incluye SKU único, nombre, precio y stock actual.

- Compra: registro de adquisiciones realizadas a proveedores.

- Venta: registro de productos vendidos a clientes.

- MovimientoStock: historial de entradas, salidas y mermas.

# 2. Lógica de Movimientos y Reglas de Stock

La aplicación implementa las siguientes reglas de negocio para la gestión del inventario:

- Entrada: aumenta el stock del producto.

- Salida: disminuye el stock, sin permitir valores negativos.

- Merma: descuenta stock por pérdida o daño.

Reglas adicionales: validación de stock disponible, SKU único, y obligatoriedad de asignar categoría y proveedor a cada producto.

# 3. Pruebas Realizadas y Resultados

Durante las pruebas se verificó lo siguiente:

- Migraciones aplicadas correctamente en PostgreSQL, creando 19 tablas.

- Creación exitosa de superusuario (usuario: richard).

- Acceso correcto al panel de administración de Django.

- CRUD funcionando para Productos, Categorías, Proveedores, Compras y Ventas.

- Reporte de movimientos de stock accesible desde vistas y administración.

- Evidencias de capturas incluidas en la carpeta docs/evidencias.

# 4. Conclusión

El proyecto de Sistema de Inventario en Django fue implementado con éxito, cumpliendo los requerimientos de la asignatura de Back End. Se integró PostgreSQL como motor de base de datos, asegurando persistencia y escalabilidad. El sistema gestiona productos, categorías, proveedores, compras y ventas, además de controlar movimientos de stock (entradas, salidas y mermas). Se comprobó su correcto funcionamiento mediante migraciones, pruebas de CRUD y creación de superusuarios.