

## Agente IA para Gestión de Inventario (Flask + MySQL)

Este proyecto implementa un servicio web que recibe consultas en **lenguaje natural (español)**, las traduce a sentencias **SQL** mediante **Groq**, y las ejecuta en una base de datos **MySQL**.

El sistema incluye una interfaz web moderna construida con **Flask** para la interacción directa con la base de datos de inventario.

### Objetivo del Agente

El agente de IA convierte solicitudes de gestión de inventario en consultas SQL válidas. Se utiliza la API de [link text](#) para el motor de traducción inteligente.

### Interfaz Web

Para interactuar con el agente:

1. Ejecutar python app.py desde el entorno virtual activo.
2. La aplicación se despliega en **http://127.0.0.1:5000**.

La interfaz web permite al usuario ingresar consultas y ver los resultados en tablas estructuradas o mensajes de estado.

### Esquema de Base de Datos

El sistema está orientado a la gestión de inventario (tienda\_inventario) y maneja las siguientes tablas:

Tabla	Descripción	Campos Clave (PK)
<b>productos</b>	Catálogo de artículos con datos de stock y precio.	id, nombre, categoria, stock
<b>ventas</b>	Registro de las transacciones históricas de productos vendidos.	id, producto_id, cantidad, fecha

### Reglas NL a SQL

El servicio guía a la IA (Groq) con reglas estrictas para garantizar la precisión y la seguridad:

- **Especificar columnas** explícitas en SELECT (Ej: SELECT nombre, stock).
- **Usar JOIN** para relacionar productos con ventas.
- **Incluir WHERE** específico en operaciones UPDATE y DELETE.
- **Validar y bloquear** operaciones peligrosas (DROP, TRUNCATE, ALTER).

### Funcionalidades

Categoría	Ejemplos de Consultas (SELECT)	Ejemplos de Operaciones DML
Consultas	"Mostrar productos con precio mayor a 50.00"	"Añadir producto 'Libreta' en Oficina con 50 unidades" (INSERT)
Reportes	"¿Cuál es el precio promedio de la categoría Electrónica?"	"Actualizar stock de 'Laptop Pro' a 20 unidades" (UPDATE)
Análisis	"Mostrar productos con stock menor a 5"	"Eliminar el producto 'Pantalón Jeans'" (DELETE)

**Requisitos Técnicos**

- Python 3.x
- MySQL Server (con el esquema tienda\_inventario)
- API Key de Groq (para el módulo de IA)

**Instalación**

Bash

# Dentro del entorno virtual (venv)

pip install Flask mysql-connector-python groq

**Estructura del Proyecto**

Agente\_SQL\_Final/

├── venv/  
  
├── templates/  
| └── index.html <-- Interfaz (HTML/JS)  
  
├── static/  
| └── style.css <-- Estilos de diseño  
  
└── app.py <-- Núcleo del Agente (Flask/Groq/MySQL)

**Configuración**

Archivo de configuración (implícito en app.py):

Python

GROQ\_API\_KEY = "tu\_clave\_groq\_aqui"

DB\_CONFIG = {  
 "database": "tienda\_inventario",

```
"user": "root",  
  
"password": "1234" # Usa tu contraseña real  
}
```

### Características de Seguridad

- **Validación** de consultas peligrosas (Prevención de DDL).
- Manejo de excepciones en Python para prevenir *crashes* por fallas en MySQL.
- **Prompt Engineering** para forzar salidas seguras si la intención es ambigua.

### Ejemplos de Consultas Comunes

Tipo de Consulta	Ejemplos de Prompt al Agente
Información Básica	"Muestra el nombre, categoría y precio de todos los productos."
Análisis de Venta	"Productos que se han vendido en el último mes."
Inventario Crítico	"Productos que tienen un stock crítico, menor a 5 unidades."