

PRACTICA 3

Sistema de Gestión de Inventario con Django y Flask



PONDERACIÓN: 100

Horas Aproximadas: 10

Universidad San Carlos de Guatemala

Facultad de ingeniería.

Ingeniería en ciencias y sistemas

Índice

Índice.....	1
Competencias.....	2
Objetivos.....	2
General.....	2
Específicos.....	2
Enunciado.....	2
Entregables.....	3
Consideraciones.....	4
Cronograma.....	4
Rúbrica de Calificación.....	4
Valores.....	5
Referencias.....	5

Competencias

Desarrollo de aplicaciones web con Django y Flask, implementación de APIs REST, integración de sistemas mediante consumo de APIs, y manejo de bases de datos para gestión de inventarios.

Objetivos

General

El objetivo de esta práctica es que los estudiantes desarrollen una aplicación web en Django que consuma una API REST creada en Flask. La API en Flask gestionará los productos de un supermercado, mientras que Django proporcionará una interfaz web para interactuar con los datos.

Específicos

- Comprender la integración entre Django y Flask mediante el consumo de una API REST.
- Implementar un CRUD en Flask para gestionar productos de un supermercado.
- Desarrollar una aplicación en Django que consuma la API Flask y muestre los datos en una interfaz web.
- Utilizar Django Templates para mejorar la presentación de los productos.
- Manejar errores y validaciones tanto en Flask como en Django para garantizar la estabilidad del sistema.

Enunciado

En el entorno competitivo de los supermercados, llevar un control adecuado de los productos disponibles es fundamental para garantizar el abastecimiento, evitar pérdidas

y ofrecer una buena experiencia al cliente. Muchos establecimientos buscan soluciones digitales que les permitan actualizar su inventario en tiempo real y de forma eficiente.

Para satisfacer esta necesidad, se desarrollará una aplicación web que permita al personal del supermercado gestionar su inventario de productos de manera sencilla e intuitiva. El sistema estará dividido en dos partes: el backend, desarrollado con Flask, que proveerá una API REST encargada de la gestión de productos, almacenando los datos en un archivo JSON llamado **inventario.json**; y el frontend, desarrollado en Django, que se comunicará con la API y permitirá visualizar y modificar los datos mediante una interfaz web.

El sistema deberá permitir realizar las siguientes operaciones sobre los productos:

- **Crear** nuevos productos con nombre, categoría, descripción, precio, cantidad en stock y fecha de vencimiento (si aplica).
- **Leer** la lista de productos disponibles y consultar los detalles de cada uno.
- **Actualizar** la información de productos existentes.
- **Eliminar** productos discontinuados o agotados.

Django se encargará de recibir las solicitudes del usuario y comunicarse con la API Flask mediante peticiones HTTP utilizando el módulo requests. Los datos se mostrarán en una interfaz amigable creada con Django Templates, permitiendo que los encargados del supermercado interactúen fácilmente con el sistema.

Como desarrollador, deberá implementar tanto la API REST en Flask como la aplicación Django, asegurándose de que ambas funcionen correctamente en conjunto. También es importante validar los datos y manejar errores en ambas aplicaciones para mantener la robustez del sistema.

Entregables

- La entrega será el día viernes 17 de octubre a más tardar a las 11:59 pm.
- La entrega será por medio de la plataforma UEDI.
- Deberá agregar al auxiliar como colaborador al repositorio (Hes-007).

Consideraciones

- El backend deberá ser desarrollado utilizando Flask, implementando una API REST para la gestión de productos.
- El frontend deberá ser desarrollado en Django, el cual consumirá la API de Flask para mostrar y gestionar los datos.
- Cualquier caso de copia parcial o total tendrá una nota de 0 y será reportada.
- Para dudas concernientes a la práctica se utilizarán los foros en UEDI de manera que todos los estudiantes puedan ver las preguntas y las posteriores respuestas.

Cronograma

Tarea	Fecha
Asignación de la práctica / Entrega del enunciado	7 de octubre de 2025
Fecha de entrega	17 de octubre de 2025 (11:59 pm)
Fecha de calificación	Por definir

Rúbrica de Calificación

Criterio	Descripción	Puntos Máximos
Backend (Flask)	Implementación correcta de los endpoints de la API REST (POST, GET, PUT, DELETE)	25
Frontend (Django)	Desarrollo de vistas y templates para la interfaz de usuario	30
Funcionalidad del Sistema	Correcta implementación de todas las funcionalidades requeridas	40

Preguntas	Verificación de comprensión y autoría de la práctica.	5
-----------	---	---

Valores

En el desarrollo de la práctica, se espera que cada estudiante demuestre honestidad académica y profesionalismo. Por lo tanto, se establecen los siguientes principios:

1. **Originalidad del Trabajo**
 - Cada estudiante o equipo debe desarrollar su propio código y/o documentación, aplicando los conocimientos adquiridos en el curso.
2. **Prohibición de Copias y Plagio**
 - Si se detecta la copia total o parcial del código, documentación o cualquier otro entregable, la calificación será de **0 puntos**.
 - Esto incluye la reproducción de código entre compañeros, la reutilización de proyectos de semestres anteriores o el uso de código externo sin la debida referencia.
3. **Uso Responsable de Recursos Externos**
 - El uso de bibliotecas, frameworks y ejemplos de código externos está permitido, siempre y cuando se referencian correctamente y se comprendan plenamente. (Consultar con el catedrático su política)
4. **Revisión y Detección de Plagio**
 - Se podrán utilizar herramientas automatizadas y revisiones manuales para identificar similitudes en los proyectos.
 - En caso de sospecha, el estudiante deberá justificar su código y demostrar su desarrollo individual o en equipo. Si este extremo no es comprobable la calificación será de **0 puntos**.


Al detectarse estos aspectos se informará al catedrático del curso quien realizará las acciones que considere oportunas.

Referencias

Django Documentation: <https://docs.djangoproject.com/>

Flask Documentation: <https://flask.palletsprojects.com/>

RESTful API Design: <https://restfulapi.net/>

 Django REST Framework: <https://www.django-rest-framework.org/>

Flask-RESTful: <https://flask-restful.readthedocs.io/>