Parcial 2: Página Web con Django

Marianne Nicté, Rodríguez Canek, 202000656[1, *]

[1] *Facultad de Ingeniería, Departamento de Electronica, Universidad de San Carlos, Ciudad Universitaria, Zona 12, Guatemala.*

INTRODUCCIÓN

Django es un framework web de alto nivel para Python pensado para crear aplicaciones rápidas, seguras y mantenibles. Su filosofía de "baterías incluidas" significa que trae muchas herramientas listas para usar —autenticación de usuarios, panel de administración, manejo de sesiones, formularios, protección contra ataques comunes y más— de modo que puedas concentrarte en la lógica de tu proyecto en vez de reinventar piezas básicas.

La arquitectura que utiliza se llama MVT (Model–View–Template). El "Modelo" describe tus datos y relaciones en Python y se conecta a la base de datos mediante un ORM que te permite hacer consultas sin escribir SQL crudo. La "Vista" contiene la lógica que responde a cada petición del navegador (qué datos cargar, qué validaciones hacer, qué devolver). El "Template" es la capa de presentación: archivos HTML con etiquetas sencillas para mostrar los datos que preparó la vista.

Un rasgo muy valorado es su panel de administración automático. Con solo registrar tus modelos, obtienes una interfaz para crear, editar y eliminar registros, ideal para gestionar catálogos, usuarios o contenidos sin tener que programar un backoffice desde cero. Además, las migraciones de base de datos te permiten versionar cambios en tus modelos y aplicarlos de forma ordenada, lo que facilita el trabajo en equipo y los despliegues.

I. EXPLICACIÓN DEL CÓDIGO

El código que se describe a continuación está diseñado para interactuar con el sistema de base de datos PostgreSQL, donde se almacenan los datos relacionados con los clientes y los productos.

1. Crear un modelo de base de datos y realizar las operaciones de crear, leer, actualizar y eliminar datos de él.
2. Conectar un proyecto realizado en Python utilizando el framework de Django.
3. Generación de la factura en formato texto y almacenamiento en un archivo de texto.
4. Inserción de los datos en una base de datos PostgreSQL.
5. Consulta de las facturas previas almacenadas en el archivo de texto y la base de datos.

A continuación, se presenta el fragmento de código:

Figura 1: Código de main.html

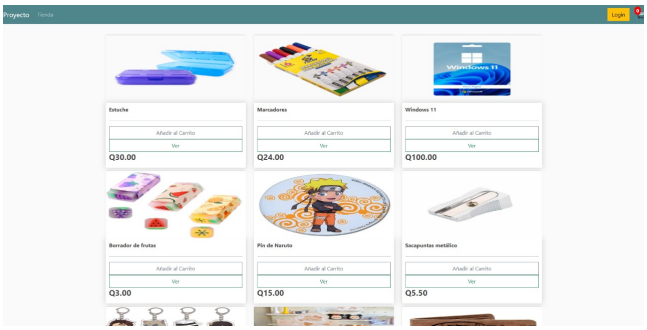

Figura 2: Menu principal de la página web.

El sistema también realiza la inserción de los datos en la base de datos PostgreSQL mediante el uso de la librería psycopg2. El siguiente fragmento muestra cómo se realiza la inserción de los datos:

II. FUNCIONAMIENTO DEL SISTEMA

El sistema permite al usuario ingresar los datos de entrada mediante un menú interactivo. Tras completar los datos necesarios, el sistema calcula el total a pagar , genera el resumen del pedido que se muestra en pantalla con los datos detallados sobre el total a pagar y se guarda en la base de datos.

* e-mail: 3243383091703@ingenieria.usac.edu.gt

III. CONCLUSIONES

- El sistema desarrollado permite gestionar de manera eficiente el despacho de combustible para vehículos, calculando automáticamente el monto total a pagar y generando facturas. La integración con una base de datos PostgreSQL facilita el almacenamiento y consulta de los datos, mientras que el archivo de texto actúa como un respaldo adicional para las facturas generadas. Este sistema mejora la gestión de cobros de combustibles y la experiencia del usuario al automatizar procesos clave como el cálculo de cobro y la generación de facturas.

IV. REPOSITORIO EN GITHUB

El código fuente de este proyecto, junto con ejemplos y otros recursos, está disponible en el siguiente enlace: `https://github.com/Marianne8934/Tareas-y-proyectos` Este repositorio contiene el código original, las mejoras realizadas y las gráficas generadas por los programas descritos en este documento.