

---

## **PROYECTO 2: CRUD**

---

**201801539 – Brenda Paola Gramajo Paniagua**

**202100692- Josue Ricardo Carias Ordoñez**

**201900081 - José Luis Saloj Julajuj**

### **Resumen**

El objetivo principal es capacitar al estudiante en el desarrollo de una aplicación web que funcione como un punto de venta. Se deben implementar operaciones básicas (CRUD) para clientes, productos y facturas, empleando Python con enfoque en POO, HTML y CSS para la interfaz visual, y el framework Django para la implementación del sistema. Se establecen requisitos específicos para los datos de clientes, productos y facturas, así como funcionalidades de búsqueda y ordenamiento. Además, se hace hincapié en utilizar plantillas HTML y CSS para la presentación de datos y almacenar la información utilizando XML como formato de almacenamiento.

### **Palabras clave**

CRUD,MVC,POO,Django,HTML,CSS,XML,Django Models, Views,Búsqueda y ordenamiento

## **Abstract**

*The main objective is to train the student in the development of a web application that works as a point of sale. Basic operations (CRUD) must be implemented for customers, products and invoices, using Python with a focus on OOP, HTML and CSS for the visual interface, and the Django framework for the implementation of the system. Specific requirements are established for customer, product and invoice data, as well as search and sorting functionalities. In addition, emphasis is placed on using HTML and CSS templates for data presentation and storing information using XML as the storage format.*

## **Keywords**

*CRUD,MVC,POO,Django,HTML,CSS,XML,Django Models, Views,Search & Sort*

## **Introducción**

El proyecto trabajado se enfoca en la realización de un CRUD (Create, Read, Update, Delete) de un punto de venta. El CRUD se desarrolló a través del framework llamado Django haciendo uso del modelo MVT (Models, Views, Templates) para su visualización en una página web. El punto de venta creado cuenta con un menú con las siguientes categorías: clientes, productos, ventas (facturas) y reportes. El usuario podrá agregar, visualizar, editar, buscar y eliminar información de cada categoría a través de formularios en HTML, en la categoría de reportes se encuentran las gráficas de polígonos de frecuencias de la información procesado en la categoría de ventas donde se muestran dos tipos de gráficas y un archivo de salida con formato XML. La primera gráfica a visualizar corresponde a la relación de cantidad de productos vendidos vrs nombres de productos, esta gráfica lleva por nombre productos más vendidos y la segunda gráfica corresponde a la relación de cantidad de productos comprados vrs nombre cliente, lleva por nombre cliente con más productos.

## **Desarrollo del tema**

El punto de venta se desarrolló en dos partes que son: el backend y el frontend, en el backend se trabajaron listas simples con sus respectivos métodos, siendo los más relevantes el método de insertar nodo, buscar, eliminar, editar, imprimir. Dichos métodos se utilizaron para crear los controladores de la página web y en ellos almacenar la información recibida de los formularios de clientes, productos y ventas.

En backend también se realizaron los métodos de salida de datos los cuales se almacenaron en formato xml, para las gráficas de productos más vendidos y clientes con más productos se instaló la librería matplotlib.pyplot en el IDE, para poder visualizar las coordenadas en un plano cartesiano y la gráfica de polígono de frecuencias, en la gráfica de productos más vendidos se utilizó la relación nombre del producto vrs cantidad vendido y en la gráfica de clientes con más productos se usó la relación nombre cliente vrs cantidad productos comprados.

En la segunda parte del proyecto se realizó la instalación del framework Django para poder crear y visualizar el entorno web. En esta parte se agregó un archivo HTML llamado index para poder acceder al menú del punto de venta, en el menú se encuentran las siguientes categorías: clientes, productos, ventas, estadística y generar xml.

En la categoría clientes se crearon otras funciones y con ello otros templates los cuales son:

1. Crear cliente:  
Permite el ingreso de datos de un cliente nuevo.
2. Ver clientes:  
Muestra todos los clientes existentes.
3. Eliminar cliente:  
Función que elimina un cliente al ingresar el nombre del cliente.
4. Buscar cliente:  
Función que busca un cliente por nombre.
5. Editar cliente:  
Función que edita a un cliente a través del DPI.

En la categoría productos se realizaron las siguientes funciones:

1. Crear producto:  
Permite el ingreso de datos de un producto nuevo.
2. Ver productos:  
Muestra todos los productos existentes.
3. Eliminar producto:  
Función que elimina un producto por nombre.
4. Buscar producto:  
Función que busca un producto por nombre.
5. Editar producto:  
Función que edita los datos existentes de un producto a través del ID.

En la categoría factura se realizaron las siguientes funciones:

1. Crear factura:  
Permite el ingreso de datos de una nueva factura.
2. Ver facturas:  
Muestra todas las facturas existentes.
3. Eliminar factura:  
Función que elimina una factura por nombre del cliente.
4. Buscar factura:  
Función que busca una factura por ID.

5. Editar factura:

Función que edita los datos existentes de una factura a través del nombre del cliente.

En la categoría estadística se agregaron las gráficas creadas en el backend:

1. Productos más vendidos
2. Clientes con más productos

Todo el programa fue desarrollado con base a la Programación Orientada a Objetos y para las listas se utilizaron tipo de datos abstractos (TDA).

*Figura 1. Menú en consola.*

*Fuente: elaboración propia.*

Todo el programa fue desarrollado en consola y el programa se desarrolló con base de Programación Orientada a Objetos y la aplicación de las Listas Enlazadas.

*Figura 2. Diagrama de Clases.*

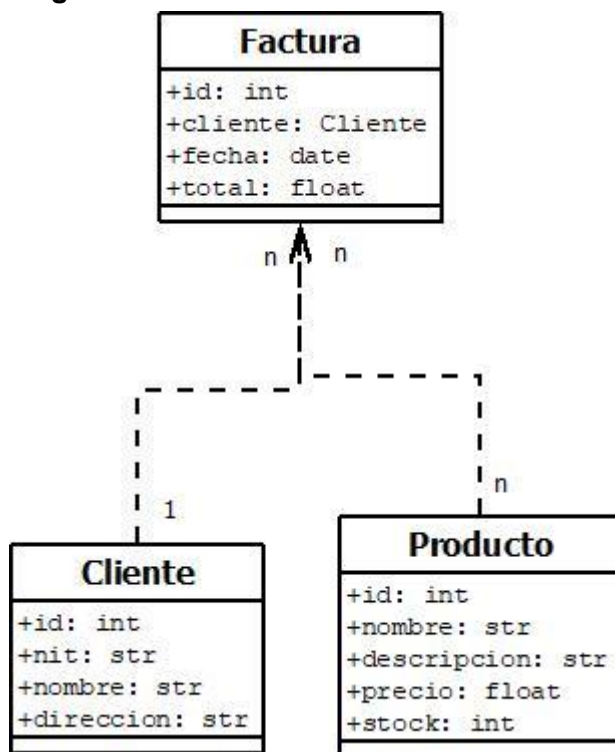
*Fuente: elaboración propia.*

## Conclusiones

- La programación orientada a objetos se basa en el concepto de crear un modelo del problema de destino en sus programas. La programación orientada a objetos disminuye los errores y promueve la reutilización del código. Python es un lenguaje orientado a objetos.
- Una aplicación web en python es necesario el uso de un framework como Django para poder visualizar y conectar los formatos de HTML.
- Es importante el uso del esquema de MVT para tener un mejor orden entre los archivos a trabajar y además proporciona mayor facilidad de manejo de formatos.

## Anexos

### Diagrama entidad relación



## Diagrama de clase

