

---

## PROYECTO 3: BANCA VIRTUAL.

---

202203009 – Mateo Estuardo Diego Noriega

### Resumen

Las listas enlazadas y los nodos son estructuras de datos utilizadas en programación para organizar y almacenar datos de manera dinámica. Estas estructuras permiten la implementación de datos que pueden crecer o reducirse en tamaño durante la ejecución de un programa, proporcionando flexibilidad en la gestión de colecciones de elementos.

La Programación Orientada a Objetos (POO) y el manejo de archivos de texto, como XML y TXT, pueden vincularse a través de la creación de clases que representen la estructura y los datos de los archivos correspondientes. Esto incluye la capacidad de crear clases para validar y procesar datos antes de escribirlos en un archivo de texto o después de leerlos.

Las gráficas desempeñan un papel esencial en la comprensión de patrones, relaciones y tendencias en los datos, convirtiéndose en herramientas valiosas para la visualización del usuario.

### Palabras clave

- Listas Enlazadas
- POO
- Manejo de archivos
- Graphviz
- Estructura XML

### Abstract

*Linked lists and nodes are data structures used in programming to organize and store data dynamically. These structures allow the implementation of data that can grow or shrink in size during program execution, providing flexibility in managing collections of elements.*

*Object-oriented programming (OOP) and handling of text files, such as XML and TXT, can be linked through the creation of classes that represent the structure and data of the corresponding files. This includes the ability to create classes to validate and process data before writing it to a text file or after reading it.*

*Graphs play an essential role in understanding patterns, relationships, and trends in data, becoming valuable tools for user visualization.*

### Keywords

- Linked Lists
- OOP
- File management
- Graphviz
- Xml structure

## Introducción

La programación se fundamenta en gran medida en la eficiente organización y manipulación de datos. Dos conceptos esenciales para lograr este propósito son los Tipos de Datos Abstractos (TDA) y las Listas Enlazadas.

Un TDA es una abstracción utilizada en programación para describir un conjunto de datos y las operaciones que se pueden llevar a cabo con ellos, sin profundizar en los detalles de cómo se implementan internamente dichas operaciones. Esto facilita la reutilización de código y la separación entre la implementación y los detalles de uso.

Por otro lado, una lista enlazada es una estructura de datos empleada para organizar elementos de manera dinámica. A diferencia de los arrays, que tienen un tamaño fijo, las listas enlazadas posibilitan la adición o eliminación sencilla de elementos en cualquier posición.

La comprensión y manipulación de estos conceptos resultan indispensables para desarrollar una aplicación capaz de gestionar valores o datos de manera efectiva.

## Desarrollo del tema

La habilidad de programación se ha convertido en esencial en la era digital y encuentra aplicación en diversos campos, abarcando desde el desarrollo de aplicaciones móviles y sitios web hasta la automatización industrial y la inteligencia artificial. En la industria tecnológica, se valora considerablemente la capacidad para escribir código efectivo y resolver problemas de manera lógica.

Siguiendo esta premisa, el proyecto se centra en la creación de una aplicación, lo cual implica considerar la presentación visual para el usuario, asegurándose de que sea amigable, fácil de comprender y utilizar.

Un archivo "xml" se emplea en diversas aplicaciones, como el intercambio de datos entre sistemas, la configuración de software, la representación de datos en la web (por ejemplo, en RSS y HTML), el almacenamiento de información estructurada en bases de datos, entre otros usos.

Relacionándolo con la aplicación en cuestión, es crucial que esta cuente con una estructura que incluya un prototipo para el menú principal, desde el cual se ejecutarán posteriormente las funciones.



*Figura 1. Página Principal.*

Fuente: Elaboración Propia, 2024.

Se puede iniciar de varias maneras, pero para saber cuál es la más eficaz es necesario comprender de qué va el enunciado.

### a. Estructura de Datos

Se definen diccionarios (clientes, bancos) y listas (transacciones, pagos) para almacenar información sobre clientes, bancos, transacciones y pagos respectivamente.

### b. Funciones para generar PDF

- crear\_pdf\_configuracion: Crea un PDF con la configuración a partir de datos XML.
- crear\_pdf\_transaccion: Crea un PDF de transacciones a partir de datos XML.

### c. Rutas de la API

- /cargar-configuracion: Procesa y carga la configuración desde un archivo XML.
- /cargar-transacciones: Procesa y carga transacciones desde un archivo XML.
- /reset: Restablece los datos a su estado inicial.
- /ingresos: Obtiene ingresos y genera un PDF con un gráfico de barras.
- /estado-cuenta/<nit\_cliente>: Obtiene el estado de cuenta de un cliente y genera un PDF con la información.

### d. Métodos de Apoyo

- generar\_y\_guardar\_respuesta\_config\_xml: Genera y guarda un archivo XML de respuesta para la configuración.
- generar\_y\_guardar\_respuesta\_transac\_xml: Genera y guarda un archivo XML de respuesta para las transacciones.
- validar\_nit, extraer\_fechas: Métodos de validación y extracción de datos.
- procesar\_config\_xml, procesar\_transac\_xml: Procesan datos XML para clientes, bancos, facturas y pagos.

### e. Métodos para consultar datos

- /clientes, /bancos, /transacciones, /pagos: Devuelven información sobre clientes, bancos, transacciones y pagos respectivamente.
- /info: Devuelve información del estudiante.

### f. Funciones para guardar y cargar datos

Guarda y carga los datos en/desde un archivo JSON.

Inicio de la aplicación: Carga los datos al iniciar la aplicación y ejecuta la aplicación en el puerto 8000 en modo de depuración. Este código forma parte de una aplicación web que permite cargar configuraciones, transacciones, consultar información, generar PDFs y visualizar gráficos relacionados con ingresos y estados de cuenta.

### Conclusiones

El programa constituye una aplicación web desarrollada en Python utilizando el framework Flask. Esta aplicación tiene como objetivo principal gestionar datos relacionados con clientes, bancos, transacciones y pagos, ofreciendo diversas funcionalidades como la carga de configuraciones desde archivos XML, el procesamiento de transacciones, consultas de información y generación de PDFs.

En términos de estructura y organización, el código sigue una arquitectura típica de aplicaciones web. Utiliza diccionarios y listas para almacenar y gestionar datos, así como funciones específicas para generar documentos PDF a partir de datos XML. Además, implementa rutas de API RESTful que permiten interactuar con la aplicación a través de solicitudes HTTP, como cargar configuraciones, obtener información sobre clientes, bancos, transacciones y pagos, entre otras operaciones.

### Referencias bibliográficas

AWS Amazon. (s.f.). AWS Amazon. Obtenido de <https://aws.amazon.com/es/what-is/xml/>

Graphviz. (s.f.). Graphviz . Obtenido de <https://graphviz.org/>

Salcedo, L. (2020). Mi Diario Python. Obtenido de <https://pythondiario.com/2018/07/linked-list-listas-enlazadas.html>

Universidad de san Carlos de Guatemala. (2024).  
Enunciado Proyecto 03.

## Anexos

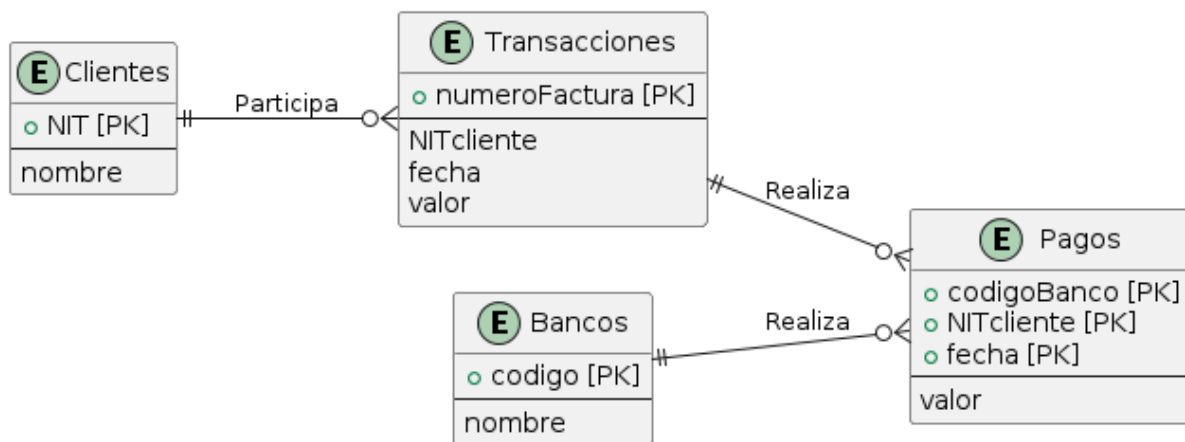


Figura 2. Diagrama Entidad – Relación.

Fuente: Elaboración Propia, 2024.