

---

## PROYECTO 1. IPC2 MARKET

---

201701010– Bryant Herrera Rubio

### Resumen

La aplicación está diseñada para gestionar un sistema de inicio de sesión y administración. El archivo `admin.py` define la clase `AdminWindow`, que configura una interfaz para tareas administrativas, incluyendo operaciones CRUD para productos. El archivo `login.py` contiene la clase `LoginWindow`, que establece una ventana de inicio de sesión donde los usuarios ingresan sus credenciales. Si se autentica como administrador o usuario normal, se muestra un mensaje de éxito y se cierra la ventana de inicio de sesión. En caso de credenciales incorrectas, se muestra un mensaje de error. Finalmente, `main.py` inicia la aplicación al crear una instancia de `LoginWindow` cuando se ejecuta como programa principal, facilitando la interacción del usuario con la interfaz de inicio de sesión. Este diseño modular permite una gestión eficiente de usuarios y administradores dentro de la aplicación.

### Palabras clave

Listas

Manejo de datos

Graphviz

Cola

GIT

### Abstract

*The application is designed to manage a login and administration system. The `admin.py` file defines the `AdminWindow` class, which sets up an interface for administrative tasks, including CRUD operations for products. The `login.py` file contains the `LoginWindow` class, which establishes a login window where users enter their credentials. If authenticated as an administrator or normal user, a success message is displayed, and the login window closes. In case of incorrect credentials, an error message is shown. Finally, `main.py` initiates the application by creating an instance of `LoginWindow` when executed as the main program, facilitating user interaction with the login interface. This modular design enables efficient management of users and administrators within the application.*

### Keywords

*Listas*

*Data Handling*

*Graphviz*

*Queue*

*GIT.*

## Introducción

En el ámbito de la programación y la informática, el manejo eficiente de listas y colas es fundamental para la organización y procesamiento de datos. Las listas, estructuras de datos flexibles que permiten almacenar elementos secuenciales, facilitan operaciones como la inserción, eliminación y búsqueda. Por otro lado, las colas, estructuras FIFO (primero en entrar, primero en salir), son esenciales para la gestión ordenada de elementos, especialmente en situaciones donde se requiere un orden estricto de procesamiento. Además, herramientas como Graphviz ofrecen recursos poderosos para visualizar relaciones y estructuras de datos complejas, facilitando así el análisis y la comprensión de información interrelacionada. Este ensayo explora la importancia de estas herramientas y técnicas en la programación, destacando su papel en la optimización y eficiencia de los algoritmos y aplicaciones informáticas.

## Desarrollo del tema

### Desarrollo Detallado

#### 1. Uso de Listas Circulares y Ortogonales

En el proyecto IPC 2 MARKET, se emplean estructuras de datos avanzadas como las listas circulares y ortogonales para gestionar eficientemente la información. Las **listas circulares** se utilizan, por ejemplo, para mantener un registro continuo de productos y usuarios, permitiendo navegación circular entre elementos. Esta característica es fundamental para operaciones como la rotación de productos en promoción o la gestión de usuarios activos.

Por otro lado, las **listas ortogonales** se implementan para manejar relaciones complejas entre entidades, como las conexiones entre usuarios y sus preferencias de compra, optimizando consultas y

actualizaciones cruzadas. Cada nodo en estas listas puede estar vinculado a múltiples nodos en otras listas, lo que facilita la gestión de relaciones complejas de datos.

### Clases y Estructuras de Datos Utilizadas

#### Listas Doblemente Enlazadas

Se implementaron dos tipos de listas doblemente enlazadas:

- **ListaDoble**: Utilizada para almacenar usuarios con métodos para agregar usuarios y generar gráficos de relaciones entre nodos.
- **ListaCircularDoble**: Empleada para almacenar productos, permitiendo la circularidad en la navegación de los nodos.

#### Lista Circular Simplemente Enlazada

- **ListaCircularSimple**: Utilizada para almacenar empleados, con métodos para agregar empleados y generar representaciones gráficas de nodos.

#### Lista Ortogonal

- **ListaOrtogonal**: Utilizada para almacenar actividades organizadas de manera ortogonal, permitiendo relaciones bidireccionales entre nodos.

Cada clase está diseñada para encapsular datos relacionados y funcionalidades específicas, promoviendo así un diseño orientado a objetos que facilita la mantenibilidad y extensibilidad del sistema.

#### 3. Implementación del Inicio de Sesión

El proceso de inicio de sesión se gestiona en la clase **LoginWindow**. Cuando un usuario ingresa sus credenciales, el sistema verifica la combinación de

nombre de usuario y contraseña utilizando estructuras de datos como diccionarios o bases de datos. Si las credenciales son correctas, se muestra un mensaje de éxito y se abre la ventana correspondiente (ya sea para administrador o usuario normal).

Este proceso garantiza la seguridad y autenticación adecuada, utilizando técnicas como hashing para almacenar contraseñas de forma segura y proteger la información sensible de los usuarios.

#### **4. Funcionalidad de Graphviz y Visualización de Datos**

Graphviz se utiliza para crear visualizaciones gráficas de las relaciones entre datos almacenados en listas y colas. Esta herramienta es crucial para entender la estructura interna del sistema y detectar posibles problemas de rendimiento o áreas de mejora.

Por ejemplo, se pueden generar diagramas que muestren cómo están conectados los usuarios con sus preferencias de compra o cómo se relacionan los productos en términos de categorías o precios. Esta visualización ayuda a los administradores a tomar decisiones informadas sobre la gestión de inventario, estrategias de marketing y personalización de la experiencia del usuario.

#### **5. Integración y Escalabilidad del Sistema**

La integración efectiva de listas circulares, ortogonales y visualización gráfica con Graphviz en IPC 2 MARKET no solo mejora la eficiencia operativa, sino que también prepara el sistema para escalar según las demandas del mercado y los usuarios. La combinación de estas técnicas permite una gestión robusta de datos, asegurando que el sistema pueda crecer y adaptarse dinámicamente a medida que se agregan más productos, usuarios y funcionalidades.

### **Conclusiones**

El proyecto muestra una aplicación práctica de estructuras de datos avanzadas y principios de programación orientada a objetos en Python. La combinación de diferentes tipos de listas (doblemente enlazadas, circulares y ortogonales) demuestra la flexibilidad y eficiencia de estas estructuras para manejar y visualizar datos complejos de manera organizada y efectiva en una interfaz gráfica.

Este enfoque no solo facilita la gestión de datos, sino que también mejora la experiencia del usuario final al proporcionar herramientas visuales intuitivas para explorar y manipular la información almacenada.

### **Referencias bibliográficas**

#### **Python Crash Course**

Matthes, E. (2019). Python Crash Course. No Starch Press.

#### **Pro Git**

Chacon, S., & Straub, B. (2014). Pro Git. Apress.

### **Extensión**

