
PROYECTO 1: IPC2 MARKET

202010367 – Marcelo André Juárez Alfaro
202200236 – Kevin David Tobar Barrientos

Resumen

El ensayo aborda la implementación de un programa informático para manipular y operar con patrones en el contexto de la automatización de procesos. Aplicando en el programa diferentes tipos de datos abstractos como listas simples, lista doblemente enlazada, lista circular doblemente enlazada, lista ortogonal, Cola, Pila. La relevancia de este tema radica en su aplicación en diversos sectores industriales y tecnológicos.

El desarrollo del proyecto IPC2 Market, permite cargar archivos de entrada en formato XML desde un perfil administrador, donde puede cargar usuarios, productos, empleados y actividades. Los diferentes archivos y perfiles, como usuario y empleados, se manejan por diferentes TDAs con sus respectivas operaciones básicas, leer, buscar, eliminar, añadir, etc. Para obtener funcionalidades como obtener productos y compras desde el perfil usuario. También se tienen funcionalidades como generación de reportes, donde se visualiza mejor la implementación de los TDAs. Se concluye que la herramienta informática ofrece una solución versátil y precisa para la gestión de patrones, contribuyendo a la optimización de procesos y con una interfaz óptima para el usuario.

Palabras clave

Optimización, versátil, XML

Abstract

The essay addresses the implementation of a computer program to manipulate and operate with patterns in the context of process automation. Applying different types of abstract data in the program such as simple lists, doubly linked list, circular doubly linked list, orthogonal list, Queue, Stack. The relevance of this topic lies in its application in various industrial and technological sectors.

The development of the IPC2 Market project allows you to upload input files in XML format from an administrator profile, where you can upload users, products, employees and activities. The different files and profiles, such as user and employees, are managed by different TADs with their respective basic operations, reading, searching, deleting, adding, etc. To obtain functionalities such as obtaining products and purchases from the user profile. There are also functionalities such as report generation, where the implementation of the TADs is better visualized. It is concluded that the computer tool offers a versatile and precise solution for pattern management, contributing to the optimization of processes and with an optimal interface for the user.

Keywords

Optimization, versatile, XML

Introducción

En el contexto actual de la digitalización, el comercio electrónico se ha convertido en una necesidad esencial. Este proyecto se centra en el desarrollo de IPC2 Market, una aplicación de escritorio diseñada para facilitar las compras diarias mediante una interfaz intuitiva y amigable. La relevancia del proyecto radica en su capacidad para integrar tipos de datos abstractos (TDA) y programación orientada a objetos (POO), utilizando Python y estructuras de datos avanzadas. Además, se aprovecha la visualización gráfica mediante Graphviz y la gestión dinámica de información con archivos XML. El proyecto busca responder a la pregunta: ¿Cómo implementar una solución integral y eficiente que maneje múltiples usuarios y grandes volúmenes de datos de manera ordenada y accesible? Este desarrollo no solo mejora la experiencia del usuario final, sino que también proporciona una base sólida para futuros proyectos de aplicaciones comerciales.

Desarrollo del tema

A. Implementación de Programación Orientada a Objetos (POO) en Python

La programación orientada a objetos (POO) es un paradigma fundamental en el desarrollo de software moderno, permitiendo la creación de estructuras de datos complejas y reutilizables. En el proyecto IPC2 Market, se ha utilizado Python para implementar POO, aprovechando sus características de simplicidad y flexibilidad. Clases, objetos, herencia y polimorfismo son algunos de los conceptos aplicados para crear una solución robusta y modular. La implementación de POO no solo facilita el mantenimiento del código, sino que también mejora su legibilidad y escalabilidad.

B. Uso de Estructuras de Datos Dinámicas

Las estructuras de datos dinámicas son fundamentales para manejar de manera eficiente la información en aplicaciones que requieren alta flexibilidad y rendimiento, como IPC2 Market. Se han empleado diversas estructuras de datos avanzadas para organizar y gestionar la información de usuarios, productos, empleados y actividades.

- **Listas Doblemente Enlazadas:** Se utilizan para almacenar usuarios, permitiendo una navegación bidireccional a través de la lista. Esta estructura es eficiente para operaciones de inserción y eliminación, ya que no requiere el desplazamiento de elementos como en las listas estáticas. Cada nodo contiene referencias a los nodos anterior y siguiente, facilitando el recorrido en ambos sentidos.
- **Listas Circulares Doblemente Enlazadas:** Utilizadas para los productos, estas listas permiten un acceso continuo, donde el último nodo está enlazado al primero, formando un bucle. Esta estructura es particularmente útil para aplicaciones que requieren un acceso repetitivo a los elementos, como un inventario de productos.
- **Listas Circulares Simplemente Enlazadas:** Implementadas para almacenar empleados, estas listas simplifican la gestión al mantener una referencia del último nodo apuntando al primero, permitiendo un ciclo continuo de acceso. Son eficientes en términos de memoria y operaciones básicas.

- **Listas Ortogonales:** Utilizadas para las actividades, estas listas bidimensionales permiten organizar eventos en una cuadrícula, donde las filas representan las horas del día y las columnas los días de la semana. Esto facilita la visualización y gestión de un calendario de actividades, permitiendo al administrador ver de manera clara las tareas programadas.

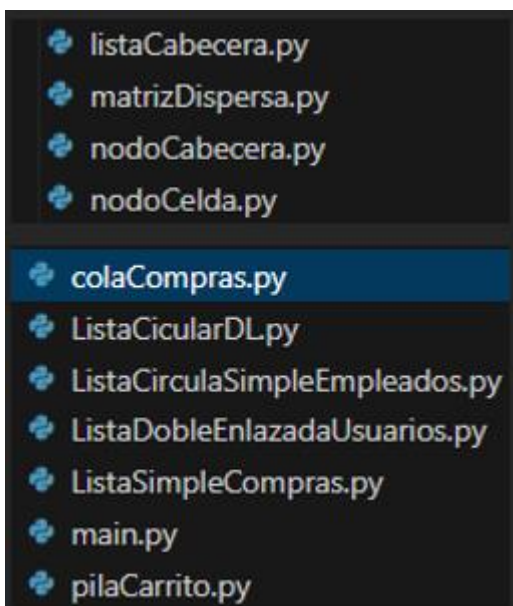


Ilustración 1

C. Integración de Archivos XML

La utilización de archivos XML como insumo para la lógica y el comportamiento de la solución es un aspecto clave del proyecto. XML proporciona una forma estandarizada de almacenar y transportar datos, lo que facilita la interoperabilidad entre sistemas. En IPCmarket, los archivos XML se utilizan para cargar datos de usuarios, productos, empleados y actividades, garantizando que la información sea consistente y fácilmente actualizable es

especialmente útil para la carga masiva de datos, donde múltiples archivos XML pueden ser procesados y fusionados sin perder la integridad de la información existente. La validación de estos archivos asegura que los datos importados cumplan con los formatos y restricciones necesarias, evitando errores y inconsistencias.

```
class XMLHandler:
    def __init__(self, file_path):
        self.file_path = file_path
        self.root = None

    def read_xml(self): #Lee el archivo XML y
                        #llama al método _parse_root() para analizar el elemento raíz.
        try:
            tree = ET.parse(self.file_path) # Parsea el archivo XML
            self.root = tree.getroot() # Obtiene el elemento raíz del árbol XML
            #self._parse_root(root)
        except FileNotFoundError:
            print("El archivo XML no se encontró.")
        except ET.ParseError:
            print("Error al analizar el archivo XML.")
```

Ilustración 2

D. Visualización de TDA's con Graphviz

La visualización de estructuras de datos abstractas (TDA) es una herramienta poderosa para el desarrollo y mantenimiento de aplicaciones complejas. En IPCmarket, se ha utilizado Graphviz, una herramienta de código abierto para la representación gráfica de estructuras de datos, para visualizar las diversas TDA implementadas.

```
import os

def graficar(self):
    try:
        codigo_dot = 'digraph G {\n'
        codigo_dot += 'rankdir=LR;\n'
        codigo_dot += 'node [shape=record];\n'

        actual = self.frente
        contador_nodos = 0

        # Crear nodos
        while actual is not None:
            codigo_dot += f'nodo{contador_nodos} [label="{actual.compraStr}";\n'
            contador_nodos += 1
            actual = actual.siguiente

        # Crear conexiones
        for i in range(contador_nodos - 1):
            codigo_dot += f'nodo{i} -> nodo{i + 1};\n'
```

Ilustración 3

Listas Doblemente Enlazadas: La visualización de la lista de usuarios mediante Graphviz permite representar gráficamente cada nodo con sus atributos y las conexiones bidireccionales entre ellos. Esto facilita la comprensión de la estructura y la detección de posibles errores en la implementación.

Listas Circulares Doblemente Enlazadas: La representación de productos en una lista circular muestra claramente la continuidad de los enlaces, lo que ayuda a visualizar el ciclo completo y asegurar que todos los nodos estén correctamente conectados.

Colas: Las solicitudes de compra se representan como una cola, con cada nodo mostrando el id del usuario, nombre del usuario, nombres de productos y el total. Esta visualización permite al administrador ver fácilmente el estado actual de la cola y gestionar las solicitudes de manera eficiente.

Listas Ortogonales: La visualización de actividades en una lista ortogonal proporciona un calendario claro, mostrando las actividades distribuidas por días y horas. Cada nodo en el gráfico incluye el ID de la actividad, nombre, descripción y el empleado asignado, permitiendo una fácil identificación y gestión de las tareas.

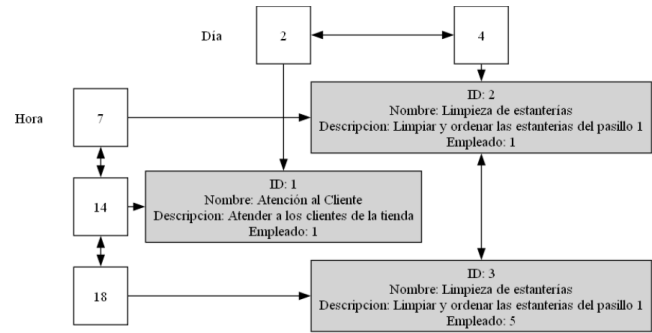


Ilustración 4

Graphviz no solo mejora la comprensión y depuración del código, sino que también sirve como una excelente herramienta de documentación. Las representaciones gráficas generadas pueden ser incluidas en los reportes y documentación del proyecto, proporcionando una visión clara y precisa de las estructuras de datos utilizadas. Además, estas visualizaciones son útiles para la comunicación entre desarrolladores y stakeholders, facilitando la explicación y justificación de las decisiones técnicas.

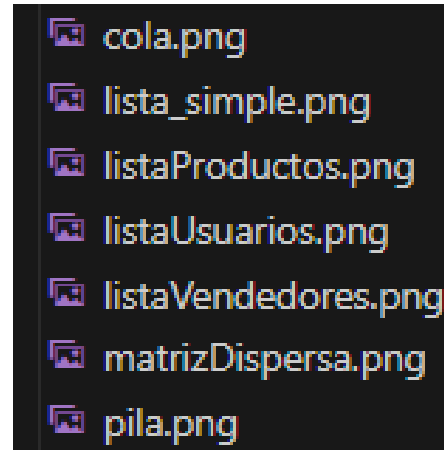


Ilustración 5

Conclusiones

El desarrollo de IPCmarket ha permitido explorar y aplicar de manera integral conceptos avanzados de programación orientada a objetos y tipos de datos abstractos. La implementación de estructuras de datos dinámicas y la integración con archivos XML han demostrado ser eficientes para manejar grandes volúmenes de información de manera ordenada y accesible. La visualización de TDA's mediante Graphviz ha facilitado la comprensión y verificación de la estructura de datos utilizada.

Este proyecto destaca la importancia de combinar técnicas de programación avanzadas con herramientas de visualización para crear aplicaciones robustas y amigables para el usuario. A través de esta implementación, se ha evidenciado la viabilidad y efectividad de estos enfoques en el desarrollo de aplicaciones comerciales.

Para futuras investigaciones, sería interesante explorar la escalabilidad de la aplicación en entornos de mayor demanda y la integración de tecnologías emergentes como la inteligencia artificial para mejorar la personalización y eficiencia del sistema.

Referencias bibliográficas

- Python para informáticos Versión 2.7.2 Charles Severance
- Fundamentos de programación, Algoritmos, Estructuras de Datos y Objetos, Luis Joyanes Aguilar, Cuarta Edición, McGraw-Hill
- Craig Larman; Introducción al análisis y diseño orientado a objetos. Prentice Hall

