
SOLUCIÓN CON TIPOS DE DATOS ABSTRACTOS Y VISUALIZACIÓN DE DATOS

202300476 – Alex Ricardo Castañeda Rodríguez

Resumen

El presente ensayo aborda el desarrollo del proyecto **IPCArt-Studio**, una plataforma innovadora para la creación y gestión de arte en píxeles, centrada en la implementación de Tipos de Datos Abstractos (TDA) bajo el paradigma de Programación Orientada a Objetos (POO). Se describen las funcionalidades principales, el diseño de la interfaz y la organización de la información mediante estructuras de datos complejas como listas enlazadas, pilas y colas. Además, se detalla la aplicación de Graphviz para la visualización gráfica de estos TDA.

Palabras clave: Tipos de Datos Abstractos, Programación Orientada a Objetos, POO, Pixel Art, Graphviz, Estructuras de Datos, Listas Enlazadas, Pilas, Colas, Visualización de Datos, Diseño de Interfaces, Herramientas de Programación.

Abstract

This essay addresses the development of the **IPCArt-Studio** project, an innovative platform for creating and managing pixel art. The project focuses on the implementation of Abstract Data Types (ADTs) under the Object-Oriented Programming (OOP) paradigm. The main functionalities, interface design, and information organization through complex data structures such as linked lists, stacks, and queues are described. Additionally, the use of Graphviz for graphical visualization of these ADTs is detailed.

Keywords: Abstract Data Types, Object-Oriented Programming, OOP, Pixel Art, Graphviz, Data Structures, Linked Lists, Stacks, Queues, Data Visualization, Interface Design, Programming Tools.

Introducción

La evolución de las herramientas digitales para la creación de arte ha llevado a la necesidad de desarrollar aplicaciones más especializadas y eficientes. En este contexto, el proyecto **IPCArt-Studio** se presenta como una solución integral que permite la gestión de arte en píxeles utilizando TDA y principios de POO. Este proyecto, desarrollado en el marco del curso *Introducción a la Programación y Computación 2*, tiene como objetivo aplicar los conocimientos adquiridos sobre estructuras de datos, algoritmos y visualización de la información.

La aplicación ofrece una experiencia interactiva para los usuarios, dividiéndolos en tres perfiles principales: administrador, artistas y solicitantes. Cada uno de estos roles posee un acceso diferenciado a las funcionalidades del sistema, permitiendo la gestión de usuarios, la administración de proyectos de arte y la visualización de los elementos gráficos generados por el sistema.

Objetivos

Objetivo General: Desarrollar una solución integral que implemente tipos de datos abstractos (TDA) y visualización de datos (Graphviz) bajo el concepto de programación orientada a objetos (POO).

Objetivos Específicos:

- Implementar POO para el desarrollo de la solución a través de lenguaje Python.
- Utilizar estructuras de programación secuenciales, cíclicas y condicionales.
- Visualizar TDA's por medio de la herramienta Graphviz.
- Utilizar archivos XML como insumos para la lógica y comportamiento de la

solución.

- Desarrollar una metodología de agrupamiento que permita optimizar la distribución de tuplas.

Desarrollo del tema

Estructuras de Datos Utilizadas

La organización de la información dentro de **IPCArt-Studio** se basa en estructuras de datos avanzadas que permiten una gestión eficiente de los usuarios, artistas, solicitantes y proyectos. A continuación, se describen las principales estructuras utilizadas:

- **Lista Doblemente Enlazada:** Se utiliza para la gestión de los solicitantes, permitiendo un recorrido bidireccional que facilita la modificación y eliminación de usuarios.
- **Lista Simplemente Enlazada:** Esta estructura se emplea para la administración de los artistas, permitiendo una navegación eficiente hacia adelante.
- **Cola:** Se utiliza para manejar las solicitudes de arte, asegurando un orden justo de atención bajo el principio *primero en entrar, primero en salir*.
- **Pila:** Esta estructura se emplea en la funcionalidad de carrito de figuras, permitiendo al solicitante acumular sus pedidos de arte antes de enviarlos a producción.
- **Matriz Dispersa:** La representación de las obras de arte en píxeles se almacena mediante matrices dispersas, optimizando la memoria y permitiendo una fácil manipulación.

Aplicación de Programación Orientada a Objetos (POO)

La implementación de POO permite encapsular los datos y las funciones relacionadas en clases, facilitando la modularidad y la reutilización de código. Se han desarrollado clases como `Usuario`, `Artista`, `Solicitante`, `Administrador`, `ColaSolicitudes`, entre otras, cada una con sus respectivos atributos y métodos.

Visualización de Estructuras con Graphviz

Para garantizar una correcta visualización de las estructuras de datos, se utiliza la herramienta Graphviz, la cual permite representar gráficamente las listas, pilas y colas. Las imágenes generadas se almacenan en la carpeta **Reportes** del sistema. Estas representaciones gráficas facilitan la verificación de la correcta implementación de los TDA.

Conclusiones

El desarrollo de **IPCArt-Studio** permitió la aplicación práctica de TDA bajo el paradigma de POO, utilizando estructuras de datos avanzadas y visualización gráfica con Graphviz. La correcta utilización de listas, pilas, colas y matrices dispersas garantizó la eficiencia en la manipulación de los datos. Este proyecto no solo facilitó la creación de arte en píxeles, sino que también sirvió como una experiencia educativa significativa en la implementación de sistemas orientados a objetos.

Anexos

Anexo 1: Código Fuente en Python

A continuación se muestra un fragmento del código fuente implementado en Python para la lectura y procesamiento de archivos XML, así como la agrupación de matrices de acceso:

Este fragmento de código muestra cómo se leen las matrices desde el archivo XML y se procesan para identificar patrones de acceso.

Anexo 2: Ejemplo de archivo XML de entrada

El siguiente es un ejemplo de un archivo XML que contiene la matriz de frecuencia de acceso. Este archivo sirve como insumo para el sistema desarrollado:

Este archivo define una matriz de 4x4 donde cada dato está etiquetado con su posición 'x' y 'y', y su valor correspondiente.

Anexo 3: Manual de Usuario

El siguiente es un manual básico de usuario que describe cómo ejecutar y utilizar el sistema para procesar archivos XML y generar gráficos de matrices:

- **Paso 1:** Ejecute el programa desde la consola utilizando el comando 'python main.py'.
- **Paso 2:** Seleccione la opción Cargar archivo.^{en} el menú de la consola e ingrese la ruta del archivo XML de entrada.
- **Paso 3:** Seleccione la opción "Procesar archivo" para agrupar las tuplas con patrones de acceso similares.

- **Paso 4:** Seleccione "Generar gráfica" para visualizar la matriz procesada mediante Graphviz.
- **Paso 5:** Opcionalmente, seleccione .^{Es}cribir archivo de salida" para generar un nuevo archivo XML con la matriz agrupada.
- **Paso 6:** Para finalizar, seleccione la opción "Salir" para cerrar el programa.

Este manual guía al usuario a través de las opciones disponibles en el sistema para cargar y procesar archivos, así como generar gráficos.

Anexo 4: Capturas de Pantalla

A continuación se muestra una captura de pantalla del sistema en ejecución, donde se observa el menú principal y los mensajes informativos mostrados al usuario:

Esta imagen muestra cómo se visualiza el sistema de menús desde la consola.



Figura 2: Captura de pantalla del sistema en ejecución.

Referencias

- Grady Booch, (1994). *Object-Oriented Analysis and Design with Applications*. Benjamin/Cummings Publishing Company.
- Robert C. Martin, (2002). *Agile Software Development, Principles, Patterns, and Practices*. Prentice Hall.
- E. Gamma, R. Helm, R. Johnson, J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.