
ART_IPC CREACIÓN DE IMÁGENES

202201131-Carlos Emanuel Sancir Reyes

Resumen

PCArt-Studio es una aplicación desarrollada con Python que utiliza listas enlazadas y nodos para implementar funcionalidades clave. La plataforma ofrece una interfaz intuitiva diseñada para la creación y gestión de pixel art, permitiendo a los usuarios alternar entre imágenes en color y escala de grises. Las imágenes se almacenan de manera optimizada utilizando matrices dispersas, asegurando eficiencia en la manipulación de proyectos grandes.

La galería de imágenes está implementada como una lista doble circular, lo que permite una navegación fluida entre las obras completadas. Además, el sistema gestiona solicitudes de pixel art a través de una pila y una cola: los usuarios acumulan imágenes en la pila para revisarlas, y luego estas se procesan en orden de llegada en la cola. Todo esto asegura un flujo de trabajo eficiente y organizado, optimizando la experiencia creativa del usuario.

- XML
- FLASK
- Graphivz

Abstract

PCArt-Studio is a Python application that uses linked lists and nodes to implement key functionalities. The platform offers an intuitive interface designed for pixel art creation and management, allowing users to switch between color and grayscale images. Images are stored in an optimized manner using sparse matrices, ensuring efficiency in handling large projects.

The image gallery is implemented as a circular double list, allowing for seamless navigation between completed works. In addition, the system manages pixel art requests through a stack and a queue: users accumulate images in the stack for review, and then these are processed in order of arrival in the queue. All of this ensures an efficient and organized workflow, optimizing the user's creative experience..

Palabras clave

- Nodos
- Matriz dispersa
- Listas Circulares

Keywords

Nodos,, Simple Lists, Circular lists, xml, doublé lists, graphivz

Introducción

Se desarrolló PArt-Studio, una plataforma diseñada para facilitar la creación y gestión de obras de pixel art, enfocada en ofrecer una herramienta eficiente y accesible para artistas digitales. La aplicación permite a los usuarios trabajar con imágenes en color y escala de grises, almacenando la información mediante matrices dispersas para optimizar el uso de recursos y mejorar el rendimiento en proyectos de gran tamaño.

Para garantizar una experiencia fluida, PArt-Studio implementa una galería interactiva utilizando una lista doble circular, lo que permite navegar de manera continua entre proyectos finalizados. Además, la aplicación incorpora una pila y una cola para gestionar solicitudes de pixel art, brindando a los usuarios un control ordenado y eficiente del flujo de trabajo. Estas funcionalidades fueron desarrolladas en Python, utilizando listas enlazadas y nodos para asegurar una implementación dinámica y eficaz.

Desarrollo del tema

La aplicación PArt-Studio está diseñada para facilitar la creación, edición y gestión de obras de pixel art, brindando una herramienta eficiente y

accesible para artistas y entusiastas del arte digital. El programa se centra en optimizar la manipulación de imágenes mediante estructuras de datos dinámicas, como listas enlazadas y nodos, logrando una implementación eficiente y organizada.

Se optó por el uso del lenguaje de programación Python debido a su simplicidad, flexibilidad y compatibilidad multiplataforma. Este lenguaje permite el manejo dinámico de memoria y la implementación de estructuras de datos necesarias para cumplir con las funcionalidades del programa. La interfaz gráfica fue desarrollada con librerías como Tkinter, que ofrece una ventana interactiva y fácil de usar para el manejo de las imágenes y herramientas de pixel art.

La solución implementada utiliza matrices dispersas para el almacenamiento de las imágenes, lo que permite optimizar el uso de la memoria y procesar grandes proyectos de manera eficiente. Esta técnica garantiza que solo se almacenen los píxeles necesarios, reduciendo el consumo de recursos sin comprometer el rendimiento.

```
<?xml version="1.0"?>
<Artistas>
  <Artista id="ART-221" pnd="8318">
    <NombreCompleto>Miguel Hernández</NombreCompleto>
    <CorreoElectronico>miguel.hernandez@sample.com</CorreoElectronico>
    <NumeroTelefono>818299431</NumeroTelefono>
    <Especialidades>Animación, Diseño Gráfico</Especialidades>
    <NotasAdicionales>Creativo y eficiente en pixel art.</NotasAdicionales>
  </Artista>
  <Artista id="ART-422" pnd="6086">
    <NombreCompleto>Juan Sánchez</NombreCompleto>
    <CorreoElectronico>juan.sanchez@mail.com</CorreoElectronico>
    <NumeroTelefono>286288522</NumeroTelefono>
    <Especialidades>Pixel Art, Concept Art</Especialidades>
    <NotasAdicionales>Gran atención al detalle en ilustraciones.</NotasAdicionales>
  </Artista>
  <Artista id="ART-811" pnd="3231">
    <NombreCompleto>Pablo Pérez</NombreCompleto>
    <CorreoElectronico>pablo.perez@mail.com</CorreoElectronico>
    <NumeroTelefono>628903857</NumeroTelefono>
    <Especialidades>Ilustración Digital, Animación</Especialidades>
    <NotasAdicionales>Apto para dibujar cosas bonitas.</NotasAdicionales>
  </Artista>
  <Artista id="ART-798" pnd="7633">
    <NombreCompleto>Ana Sánchez</NombreCompleto>
    <CorreoElectronico>ana.sanchez@mail.com</CorreoElectronico>
    <NumeroTelefono>928688553</NumeroTelefono>
    <Especialidades>Animación, Diseño Gráfico, Ilustración Digital</Especialidades>
    <NotasAdicionales>Especialista en retratos digitales.</NotasAdicionales>
  </Artista>
</Artistas>
```

Figura 1. Estructura de artitas

Fuente: elaboración propia,

Además, PArt-Studio cuenta con una galería implementada mediante una lista doble circular, que facilita la navegación fluida y continua entre las obras completadas. Esta estructura permite a los usuarios

revisar y seleccionar proyectos previos sin interrupciones, brindando una experiencia intuitiva y ágil.

Para la gestión de solicitudes de imágenes, se implementó una pila y una cola. La pila permite acumular y revisar las solicitudes antes de ser enviadas, mientras que la cola procesa dichas solicitudes en orden de llegada, garantizando un flujo de trabajo organizado y equitativo.

En conjunto, la combinación de Python, estructuras de datos dinámicas y herramientas gráficas proporciona una solución robusta y eficiente para la creación y gestión de pixel art. PArt-Studio no solo optimiza la manipulación de imágenes, sino que también ofrece un entorno intuitivo y funcional que facilita el trabajo creativo de los usuarios.

```
<?xml version="1.0" encoding="UTF-8" ?>
<figura>
  <nombre id="3">Rana</nombre>
  <diseño>
    <pixel fila="1" col="3">#000000</pixel>
    <pixel fila="1" col="4">#000000</pixel>
    <pixel fila="1" col="5">#000000</pixel>
    <pixel fila="1" col="10">#000000</pixel>
    <pixel fila="1" col="11">#000000</pixel>
    <pixel fila="1" col="12">#000000</pixel>
    <pixel fila="2" col="2">#000000</pixel>
    <pixel fila="2" col="3">#A9D11B</pixel>
    <pixel fila="2" col="4">#A9D11B</pixel>
    <pixel fila="2" col="5">#A9D11B</pixel>
    <pixel fila="2" col="6">#000000</pixel>
    <pixel fila="2" col="7">#000000</pixel>
    <pixel fila="2" col="8">#000000</pixel>
    <pixel fila="2" col="9">#000000</pixel>
    <pixel fila="2" col="10">#A9D11B</pixel>
    <pixel fila="2" col="11">#A9D11B</pixel>
    <pixel fila="2" col="12">#A9D11B</pixel>
    <pixel fila="2" col="13">#000000</pixel>
    <pixel fila="3" col="2">#000000</pixel>
    <pixel fila="3" col="3">#A9D11B</pixel>
    <pixel fila="3" col="4">#000000</pixel>
    <pixel fila="3" col="5">#A9D11B</pixel>
    <pixel fila="3" col="6">#A9D11B</pixel>
    <pixel fila="3" col="7">#A9D11B</pixel>
    <pixel fila="3" col="8">#A9D11B</pixel>
    <pixel fila="3" col="9">#A9D11B</pixel>
```

Figura 2. Estructura de Imagen

Fuente: elaboración propia,

ID = ID de la imagen

Pixel = Fila y Columna

Color = #000000 en hexadecimal

Estructura de los archivos XML

La aplicación obtiene la información necesaria mediante archivos con formato **XML**, que estructuran los datos de manera ordenada y fácil de procesar. A continuación, se describen los dos tipos de información más relevantes:

2.1 Configuración de imágenes

El archivo XML de las imágenes incluye las siguientes características:

- **Fila y columna:** Representan la posición exacta de cada píxel en la imagen.
- **Color:** Define el valor del color que tendrá el píxel, permitiendo la representación tanto en escala de grises como en colores RGB.

Esta información es procesada para generar las imágenes en la plataforma, utilizando **matrices dispersas**, lo que permite un almacenamiento eficiente al manejar únicamente las posiciones con datos relevantes y evitando el uso de memoria innecesaria.

2.2 Información de artistas y solicitantes

El archivo XML también almacena la información de los artistas y solicitantes, lo que permite gestionar perfiles de usuarios y sus especialidades. Cada artista tiene un identificador único (ID), contraseña, nombre, correo electrónico, número telefónico, especialidades y notas adicionales.

Tabla I.

Tabla de Endpoint

Método HTTP	Endpoint	Descripción	Parámetros / Cuerpo
GET	/usuarios	Obtener la lista de todos los usuarios.	Ninguno
GET	/usuarios/{id}	Obtener detalles de un usuario específico.	id (en la URL)
POST	/usuarios	Crear un nuevo usuario.	Cuerpo: { "nombre": "string", "email": "string", "contraseña": "string" }
PUT	/usuarios/{id}	Actualizar la información de un usuario existente.	id (en la URL), Cuerpo: { "nombre": "string", "email": "string" }

3. Optimización mediante estructuras dinámicas

3.1 Matriz dispersa para almacenamiento de imágenes

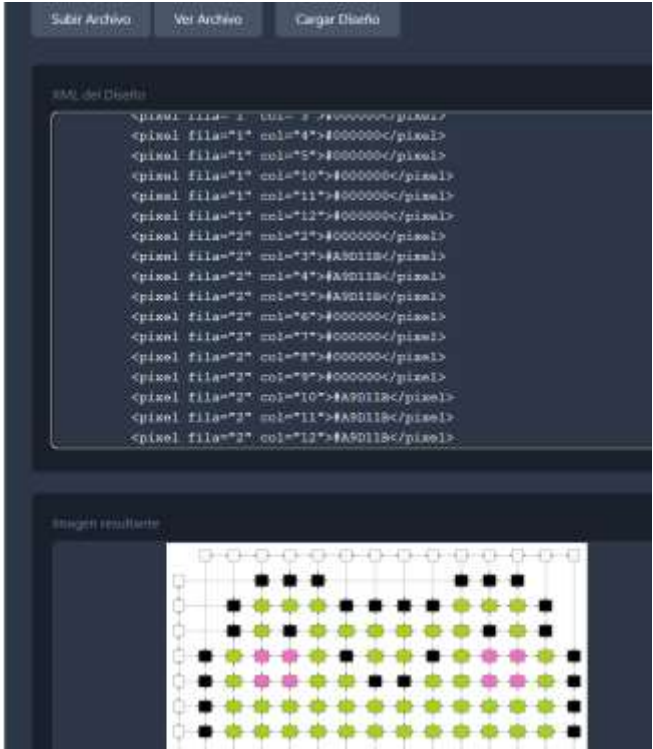
La **matriz dispersa** se utiliza para almacenar las imágenes de manera eficiente, guardando únicamente las posiciones (fila, columna) que contienen información de color. Esto permite una representación optimizada de obras de gran tamaño, reduciendo el uso de memoria y mejorando el rendimiento en la manipulación de imágenes.

Funcionamiento básico:

- Cada píxel relevante se guarda con sus coordenadas y color.
- Las posiciones vacías no ocupan espacio en memoria.

3.2 Usuario

El usuario podrá crear sus propias imágenes además de poder cambiar el diseño en dos tonos a su gusto.



Conclusiones

3.2 Usuario Editor

El usuario podrá elegir el id de la imagen que creo para poder editarla



3.3 Ayuda

En este apartado el usuario podrá conseguir ayuda del manual y datos del creado de esta plataforma.

3.4 Admin Carga masiva

El administrador podrá cargar a todos los usuarios mediante un archivo xml el cual se aguardara con persistencia y usuario que no cumplan con ciertos requisitos no se agregaran.



3.5 Ver usuario y xml

En este apartado podrá apreciar a todos los usuario y el xml con los datos cargados y poder apreciar los datos con fotografía.



3.6 estadística

Puede apreciar que usuario ha hecho más imágenes.



Conclusión

La aplicación **PCart-Studio** combina eficiencia y usabilidad al utilizar estructuras dinámicas como matrices dispersas, listas doblemente enlazadas y

sistemas de pila y cola. El uso de archivos **XML** permite una gestión organizada de datos de imágenes y usuarios, garantizando una solución robusta y escalable para artistas digitales y solicitantes de pixel art.

Referencias bibliográficas

Referencias bibliográficas USAC, T.d.(7 de Marzo de 2022). Proyecto 2, UEDi. Obtenido de https://uedi.ingenieria.usac.edu.gt/campus/pluginfile.php/283008/mod_resource/content/0/%5BIPC2%5DProyecto%20_2_1S2024-v2.pdf

C. J. Date, (1991). *An introduction to Database Systems*. Addison-Wesley Publishing Company, Inc.

DELETE	/usuarios/{id}	Eliminar un usuario.	id (en la URL)
GET	/productos	Obtener la lista de todos los productos.	Ninguno
GET	/productos/{id}	Obtener detalles de un producto específico.	id (en la URL)
POST	/productos	Crear un nuevo producto.	Cuerpo: { "nombre": "string", "precio": "number", "stock": "number" }
PUT	/productos/{id}	Actualizar la información de un producto existente.	id (en la URL), Cuerpo: { "nombre": "string", "precio": "number", "stock": "number" }

Tabla 2. Modelo del tipo de dato NodoCabecera.
Fuente: Elaboración Propia, 2024

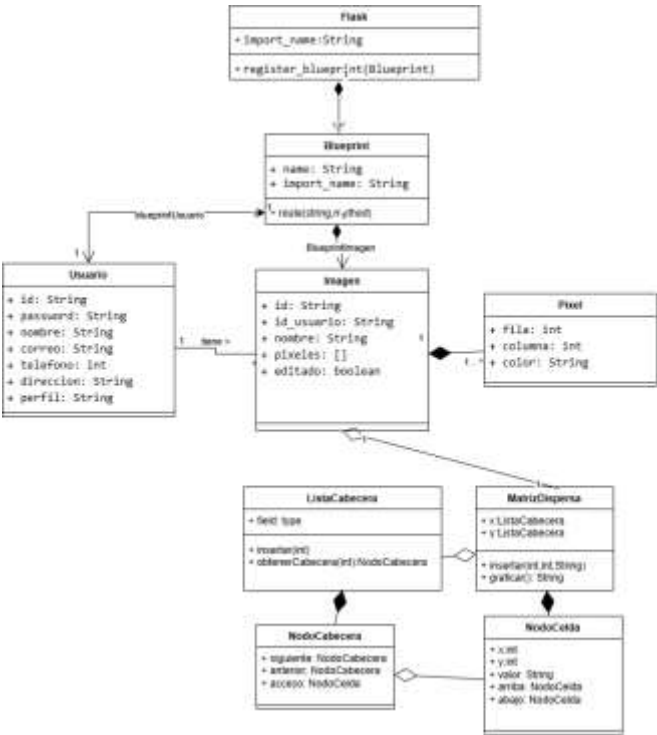


Figura 4. Modelo de
Fuente: Elaboración Propia, 2024