
IPCART-STUDIO WEB

202300694 – Eduardo Sebastian Gutierrez Felipe

Resumen

El siguiente ensayo presenta la descripción del desarrollo de la aplicación IPCart-Studio, una aplicación de escritorio con interfaz gráfica amigable al usuario, la cual simula una plataforma dedicada a la creación y gestión de arte en píxeles. En esta plataforma existe un usuario Administrador, el cual es el encargado de ingresar todos los usuarios al sistema por medio de un archivo XML. Luego, los usuarios con rol de solicitante pueden cargar el diseño que deseen para crear una matriz de píxeles a través de la carga de un archivo XML que contiene la fila, columna y color de los píxeles de la imagen. Las solicitudes se procesarán inmediatamente y se mostrará el resultado en el perfil del solicitante. Posteriormente, esta imagen se cargará a la galería global donde cualquier usuario podrá verla, así mismo, la imagen podrá ser editada por cualquier otro usuario.

Palabras clave

Cola, pila, listas, XML, clase

Abstract

The following essay presents the description of the development of the IPCart-Studio application, a desktop application with a user-friendly graphical interface, which simulates a platform dedicated to the creation and management of pixel art. In this platform there is an Administrator user, who is in charge of entering all users into the system through an XML file. Then, users with the requester role can upload the design they want to create a pixel matrix by uploading an XML file containing the row, column and color of the pixels in the image. The requests will be processed immediately and the result will be displayed in the requester's profile. Later, this image will be uploaded to the global gallery where any user can see it, and the image can also be edited by any other user.

Keywords

Queue, stack, lists, XML, class.

Introducción

Se desarrolló un programa que simula una aplicación de arte en píxeles. Para que el programa funcione debe existir un usuario Administrador, quien será el encargado de cargar los usuarios válidos para poder acceder al sistema. Los solicitantes serán los que carguen la imagen que desean a través de un archivo XML que contendrá la descripción de los píxeles que contiene la imagen.

Esta imagen estará disponible para ver por cualquier usuario del sistema, con su ID, nombre de la imagen y el ID del usuario que cargó el diseño. Además, cualquier usuario será capaz de editar cualquier imagen y adueñarse del diseño, que también será cargado en la galería global.

Todo el sistema se construyó utilizando enteramente Python, con un backend desarrollado en el framework Flask, y un frontend desarrollado en el framework Django.

Desarrollo del tema

La app consiste en una simulación de una plataforma de pixel art (arte en píxeles) en la que un usuario solicita la creación de una imagen, y el sistema es el encargado de crear el diseño en píxeles con una matriz dispersa. Esta solución fue elaborada en el lenguaje de programación Python ya que es un lenguaje multiparadigma que facilita la creación de backend y de frontend con sus diferentes frameworks Flask y Django, respectivamente, así como también permite la lectura de archivos XML con sus diversas bibliotecas. Para que el sistema funcione, existe un usuario Administrador que es el encargado de cargar los usuarios que tendrán acceso al sistema. Existe un único tipo de usuario, los solicitantes, que solicitan las imágenes a partir de un archivo XML que contiene

la descripción detallada de la imagen. El XML con los solicitantes que el Administrador carga debe contener la siguiente estructura:

```
<?xml version="1.0" encoding="utf-8"?>
<solicitantes>
  <solicitante id="IPC-000" pwd="1417">
    <NombreCompleto>Carlos Ruiz</NombreCompleto>
    <CorreoElectronico>carlosruiz@example.com</CorreoElectronico>
    <NumeroTelefono>30306453</NumeroTelefono>
    <Direccion>Ciudad de Guatemala, Guatemala</Direccion>
    <perfil>https://fastly.picsum.photos/id/102/200/200.jpg?hmac=mlVQr0-wbZ-PRsf_R4ofkUDR7yFIKqT3F8pfJNFZKc8</perfil>
  </solicitante>
  ...
</solicitantes>
```

Figura 1. Estructura del XML de solicitantes

Fuente: elaboración propia

donde:

solicitantes: es la raíz del archivo.

solicitante: es la etiqueta que contiene toda la información de un solicitante.

id: es el atributo ID único de cada usuario solicitante.

pwd: es el atributo que contiene la contraseña de la cuenta del solicitante.

NombreCompleto: es la etiqueta que contiene el nombre del solicitante.

CorreoElectronico: es la etiqueta que contiene el email del solicitante.

NumeroTelefono: es la etiqueta que contiene el número telefónico del solicitante.

Direccion: es la etiqueta que contiene la dirección del domicilio del solicitante.

perfil: contiene el link a la imagen de usuario del solicitante

Adicionalmente, el Administrador en su rol de encargado de gestionar el sistema, puede ver todos los usuarios del sistema de dos maneras, una por medio de tarjetas de presentación con la imagen y la información de cada usuario, y la otra es por medio

de un archivo XML donde se detalla toda la información de cada usuario, así como la lista de imágenes que ha cargado.

Por último, el Administrador también puede ver el apartado de estadísticas, donde se muestran dos gráficas de barras: una con el top 3 de usuarios con más imágenes cargadas y otra con la cantidad de imágenes cargadas por usuario de forma descendente.

Una vez cargados los usuarios, los solicitantes pueden iniciar sesión con su nombre de usuario y contraseña. En su módulo, puede cargar figuras, ver la galería o editar una imagen existente. La galería únicamente estará disponible cuando algún usuario cargue un XML con un diseño. En el apartado de Cargar Imagen, el solicitante puede cargar las figuras en XML que posteriormente se volverán una obra pixel-art. Las imágenes resultantes se guardarán en una base de datos XML en base 64, lo que favorece la ligereza de la imagen y su rápida renderización.

La estructura que debe contener el XML de figuras que subirá el Solicitante debe ser como la siguiente:

```
▼<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>
    ...
  </diseño>
</figura>
```

Figura 2. Estructura del XML de figuras

Fuente: elaboración propia

donde:

figura: es la raíz del archivo

nombre: es la etiqueta que contiene el nombre de la figura

id: es el atributo que contiene el ID de la figura

diseño: es la etiqueta que contiene la descripción de todos los píxeles que contiene la figura.

pixel: es la etiqueta que contiene el color hexadecimal correspondiente a ese pixel.

fila: atributo que indica la fila de la matriz dispersa en donde se colocará el pixel.

col: atributo que indica la columna de la matriz dispersa en donde se colocará el pixel.

Todos los usuarios tienen la posibilidad de editar una imagen, en el apartado de Editar Imagen. Aquí, el usuario debe ingresar el ID de la imagen que desea modificar y seleccionar la tonalidad en que quiere la nueva imagen. Cuando el sistema termina de procesar la imagen, se muestra en pantalla tanto la imagen original como la editada. Esta nueva imagen se enviará a la galería del sistema con un nuevo ID de imagen y con el ID del usuario que la editó.

Conclusiones

La aplicación IPCArt-Studio demuestra ser una solución completa y eficiente para la creación y gestión de arte en píxeles, destacándose por su diseño estructurado y funcional. La implementación de un usuario Administrador centraliza el control del

sistema, asegurando que solo los usuarios válidos puedan acceder, lo que mejora la seguridad y organiza el flujo de trabajo. La posibilidad de que los usuarios editen imágenes fomenta la colaboración y creatividad, aunque plantea desafíos en cuanto a la autoría y gestión de versiones. El uso de Python como lenguaje de desarrollo, junto con los frameworks Flask y Django, muestra un manejo eficaz de tecnologías modernas, mientras que la utilización de archivos XML y la codificación en base64 optimizan la gestión de datos y la renderización de imágenes. La automatización del proceso de creación de diseños mediante XML permite personalizar las figuras de manera eficiente, y las visualizaciones gráficas y presentaciones visuales facilitan la experiencia del usuario administrador. No obstante, sería beneficioso implementar mecanismos que reconozcan los derechos de los creadores originales para garantizar un entorno más justo. Además, la arquitectura basada en archivos XML y su diseño modular hacen que el sistema sea escalable y adaptable a futuras mejoras, como la incorporación de herramientas de edición avanzadas o nuevos roles de usuario. En conclusión, IPCArt-Studio combina funcionalidad técnica, facilidad de uso y colaboración creativa, con un gran potencial de crecimiento y refinamiento.

Referencias bibliográficas

USAC, T.d.(17 de Diciembre de 2024). Proyecto 2, UEDi. Obtenido de <https://uedi.ingenieria.usac.edu.gt/campus/mod/assign/view.php?id=877677>

Anexos

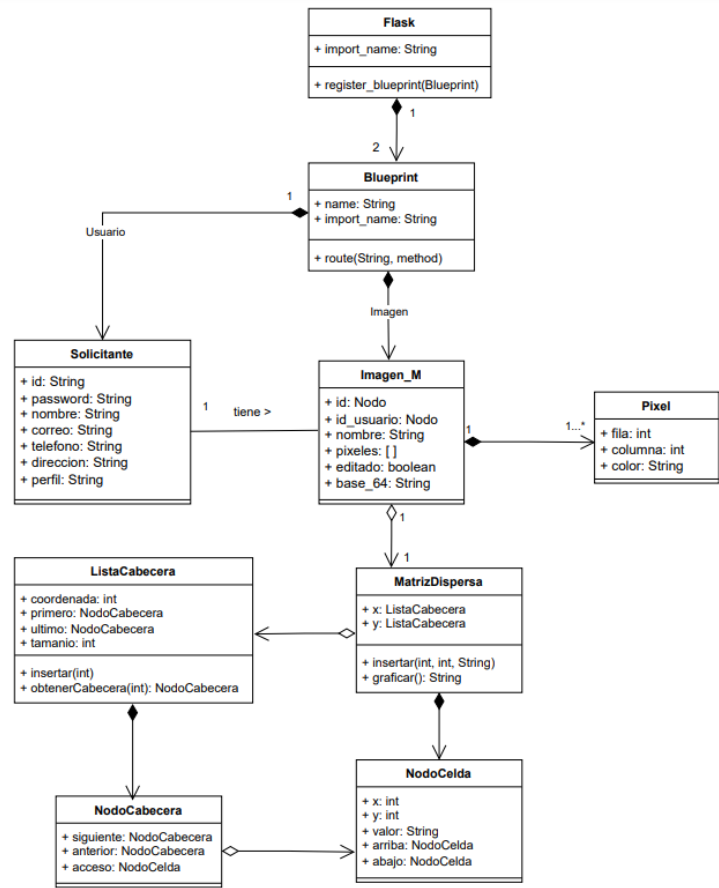


Figura 3. Diagrama de clases de la aplicación

Fuente: elaboración propia

Tabla I.

Tabla de endpoints.

Método	Endpoint
POST	/admin/cargarUsuarios
GET	/usuarios
GET	/usuarios/xml
POST	/login
GET	/usuarios/estadistica/editadas
GET	/usuarios/estadistica/cargadas
POST	/imagenes/carga/<string:id_usuario>
GET	/imagenes
POST	/imagenes/editar/<string:id_usuario>

Fuente: elaboración propia