

Introducción a la programación (CE1101)

Grupo 03

Profesor:

Leonardo Andrés Araya Martínez

Instituto Tecnológico Costa Rica

Escuela de Ingeniería en Computadores

Proyecto II Paintxel

Estudiante/Autor:

Anthony José Artavia Leitón

Documentación Externa

2024

Costa Rica

## Tabla de contenido

<b>Introducción .....</b>	<b>3</b>
<b>Objetivo general.....</b>	<b>3</b>
<b>Objetivos específicos .....</b>	<b>4</b>
<b>Descripción del Problema .....</b>	<b>4</b>
<b>Planes de solución .....</b>	<b>5</b>
<b>Herramientas y métodos utilizados.....</b>	<b>10</b>
<b>Conclusión .....</b>	<b>10</b>

## **Introducción**

Desde tiempos antiguos el ser humano a encontrado la manera de expresar artísticamente situaciones del ambiente que lo rodea, en el principio eran semi dibujos en rocas o trazos en tierra o con pinturas rupestres, pero con el tiempo esto se convirtió en uno de los caminos más eficientes para poder compartir, guardar y exhibir conocimiento de generación en generación, dando así un salto histórico en la manera como nos expresamos.

Hoy en día poseemos diversos dispositivos electrónicos que nos permiten hacer lo mismo que antaño, pero de una manera sumamente eficiente, actualmente se ha podido integrar la enseñanza digital que no es más que la utilización de medios electrónicos para demostrar e ilustrar nuevos conocimientos por medio de programas.

Bajo el contexto actual se nos encomienda poder desarrollar un programa tal que pueda utilizarse para propósitos educativos o entretenimiento cuya principal noción consiste en poder dibujar, trazar y hacer diferentes figuras para que, después, la persona dibujante pueda aplicar diferentes modificaciones a su creación, ya sea pintar, girar, reflejar entre otras acciones...

El propósito principal radica en poder crear dicho programa haciendo uso de todo el conocimiento acumulado hasta el momento gracias al curso de introducción a la programación, dicha creación deberá ser desarrollada bajo la Programación Orientada a Objetos (POO) y cualquier método para poder desarrollar y completar el problema encomendado.

## **Objetivo general**

Desarrollar un programa de Píxel Art utilizando Python, que permita la creación y manipulación de imágenes utilizando técnicas de programación orientada a objetos (POO) y matrices para la representación y modificación de los dibujos.

## **Objetivos específicos**

- Diseñar e implementar una interfaz gráfica interactiva en Python que permita al usuario crear, editar y guardar imágenes de Píxel Art.
- Utilizar matrices para representar los dibujos de Píxel Art, permitiendo así la manipulación eficiente de los píxeles y la aplicación de efectos especiales.
- Implementar clases y objetos en Python siguiendo el paradigma de programación orientada a objetos (POO) para organizar el código de manera modular y facilitar su mantenimiento y extensión.
- Integrar funcionalidades de manipulación de imágenes, como el cambio de color, el escalado y la rotación, mediante el uso de matrices y algoritmos adecuados.
- Documentar detalladamente el proceso de desarrollo del programa, incluyendo la estructura de clases, los métodos implementados, ejemplos de uso y explicaciones sobre el uso de matrices en la manipulación de dibujos de Píxel Art.

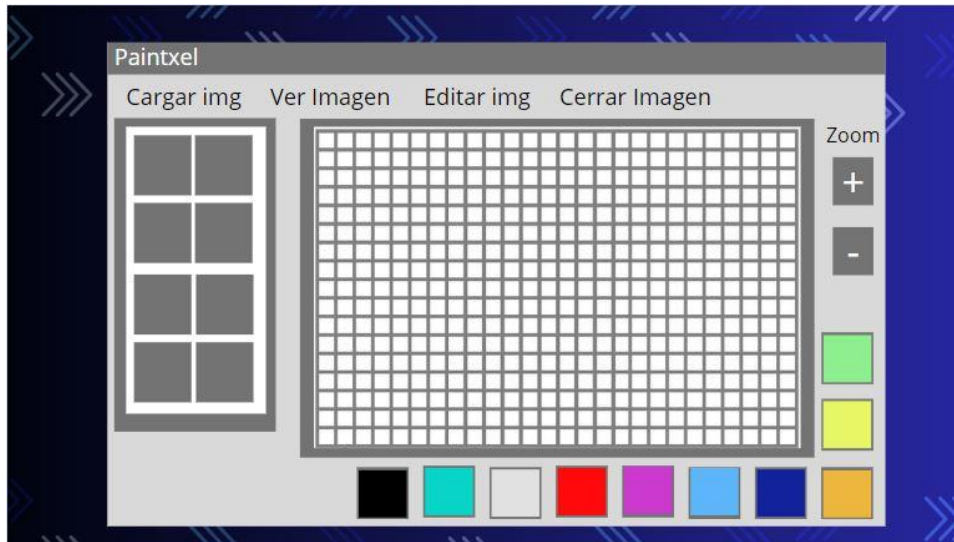
## **Descripción del Problema**

En el contexto de este proyecto, nos encontramos ante el reto de desarrollar un programa de Píxel Art con Python. Este programa tiene como objetivo principal proporcionar a los usuarios una plataforma creativa para la creación y manipulación de imágenes de Píxel Art, combinando conceptos de programación orientada a objetos (POO) y el uso de matrices para representar y transformar los dibujos. La complejidad del proyecto radica en la necesidad de integrar de manera fluida la manipulación de imágenes con la interacción del usuario, ofreciendo herramientas de edición potentes pero fáciles de usar. Esto implica implementar algoritmos eficientes para realizar operaciones como el cambio de color, la rotación y el escalado de imágenes, todo ello mientras se mantiene una interfaz gráfica atractiva y receptiva.

## Planes de solución

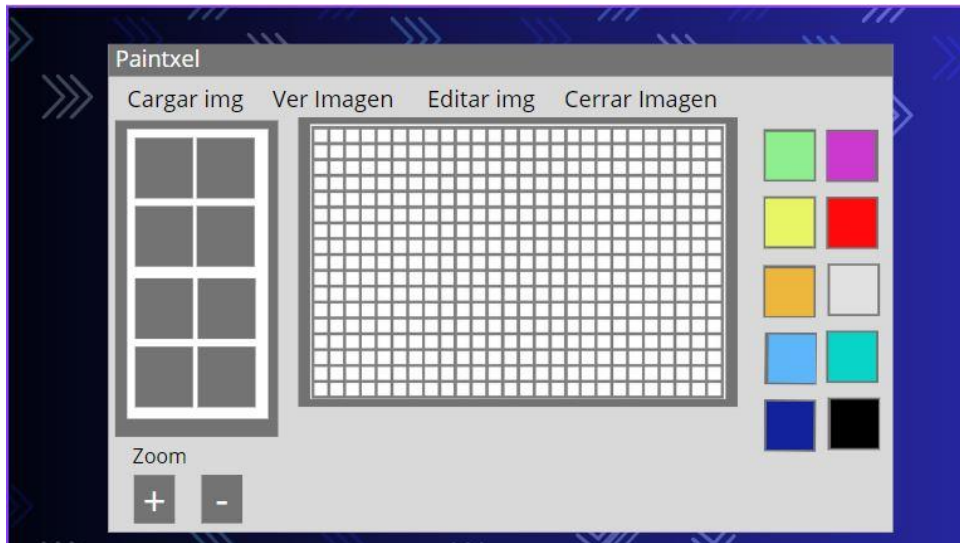
Plan 1:

Imagen de referencia para la creación de la interfaz gráfica:



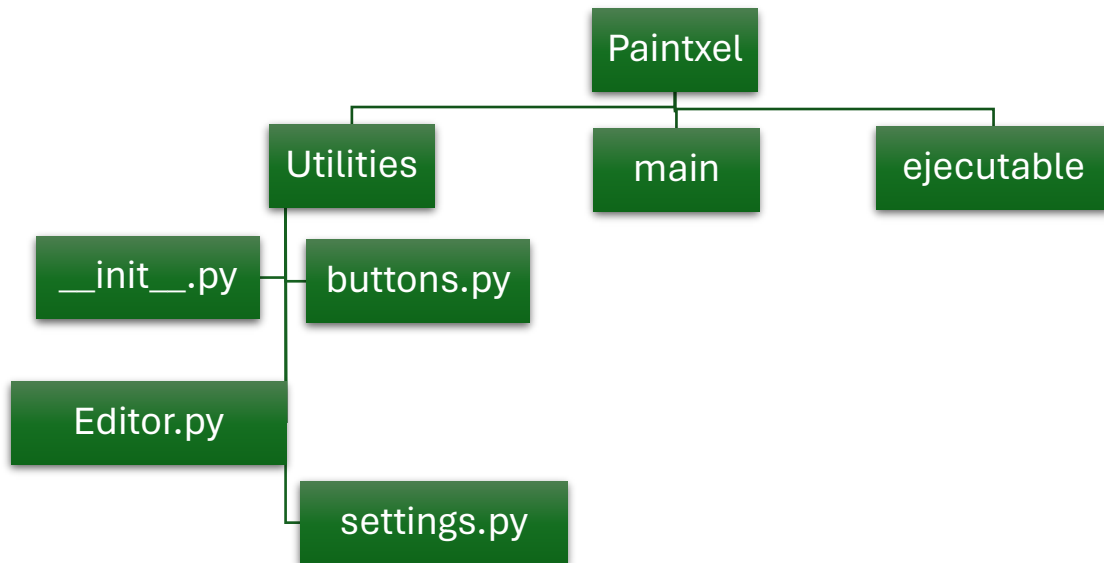
Plan 2:

Imagen de referencia para la creación de la interfaz gráfica:



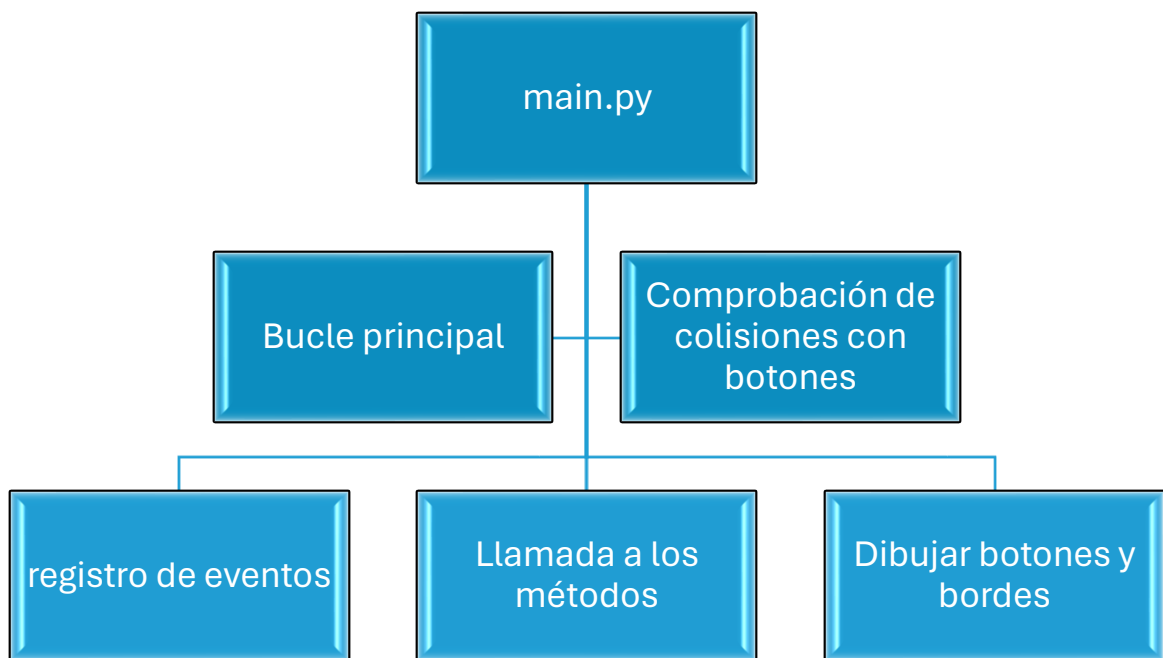
## Diagrama de organización del código:

Sistema de Carpetas:

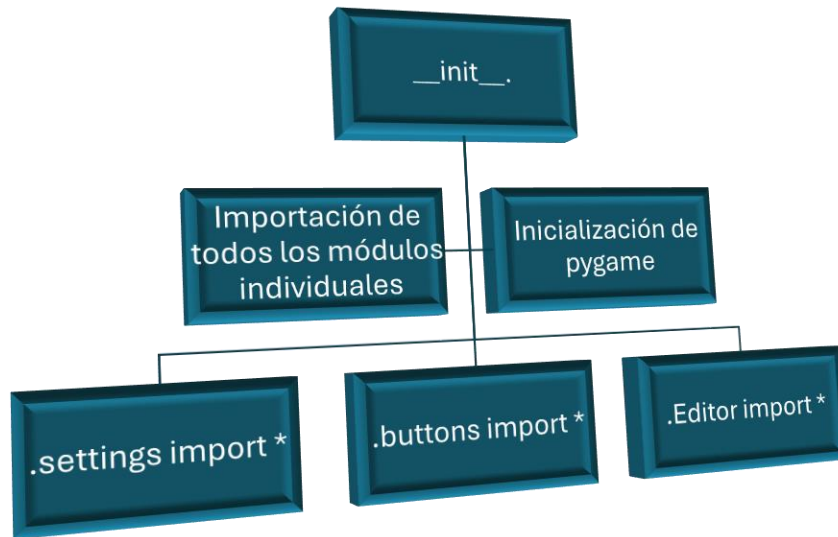


Organización del Código:

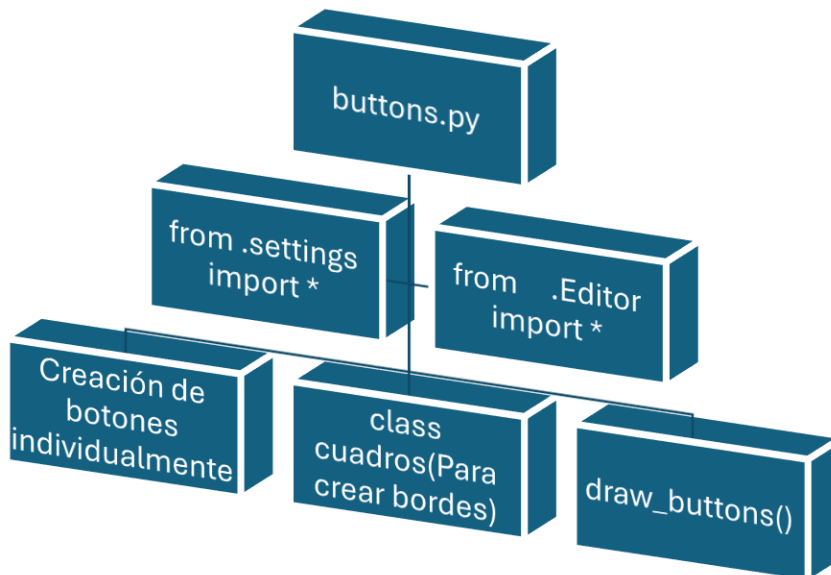
### Main.py:



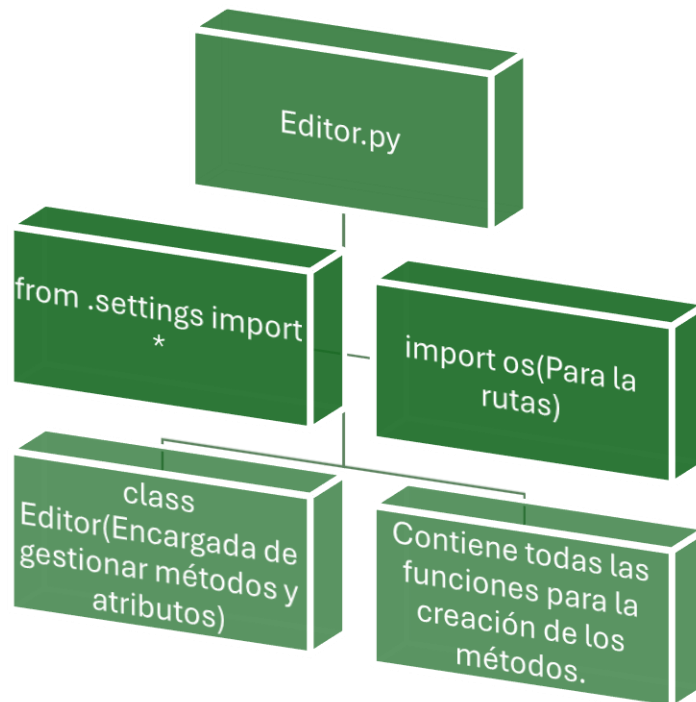
## init .py:



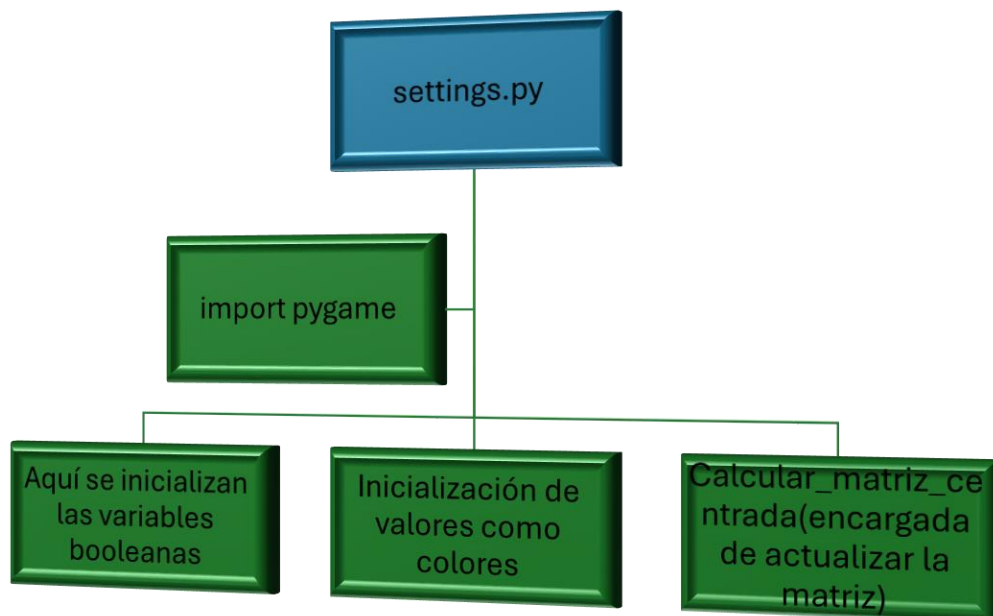
## Buttons.py:



## Editor.py:



## settings.py:





## **Pros y contras de las soluciones planteadas:**

### **1. Código en documento único:**

La implementación de un documento o archivo con todo el código dentro es una opción poco viable debido a que se desorganiza fácilmente el código, además de que no es la mejor manera de implementar una solución limpia

#### **Pros:**

- Rápido de armar.
- Simplificación de cantidad de archivos.
- Todas las funciones se encuentran en el mismo archivo, por lo tanto, no hay necesidad de importar archivos externos y se hace más fácil el llamar a dichas funciones.

#### **Contras:**

- Desorganización del código.
- Es menos legible por la cantidad de líneas de código y mezclas de funciones.
- Es más difícil encontrar errores.
- Más difícil aplicar buenas prácticas.

### **2. Código modular (varios archivos):**

Esta clase de implementación implica tener que organizar diferentes funcionalidades en distintos archivos, es decir, se tiene un archivo separado para cada propósito en específico.

#### **Pros:**

- Mucho más organizado
- Fácil de organizar el código y Más legible
- Encontrar errores es más fácil

#### **Contras:**

- Más complejo
- Merece tener un pensamiento estructural
- Más archivos por separado, no centralizado.

## Herramientas y métodos utilizados

- **Python:** Se utiliza el lenguaje de programación “Python” para poder programar la lógica del programa y sus funcionalidades.
- **Pygame:** Librería especializada en la creación de interfaces, utilizada para poder dar una apariencia mucho mejor al programa y en general poder hacer las diversas funcionalidades de una mejor manera.
- **YouTube:** Utilizado para investigar diferentes implementaciones a integrar en el programa, principalmente para poder hacer una “Lluvia de ideas” en base a videos explicativos.
- **Editor de texto (Visual Studio):** Se utiliza el Editor de texto “Visual Studio”, utilizado para poder tener un mejor manejo sobre el código y los archivos a utilizar, principalmente ayuda a detectar errores en la sintaxis y además permite organizar de mejor manera el código.
- **Copilot:** Se utiliza esta inteligencia artificial para poder crear las imágenes de los botones y diferentes interfaces.

## Conclusión

Durante la creación de este proyecto surgieron muchos inconvenientes donde fue necesario adaptar técnicas y algunos recursos para el desarrollo de este, se integraron las técnicas aprendidas en clase para la creación de un programa sostenible y parcial o totalmente funcional, de manera que cumpla con la mayoría de los requisitos, algunas de las técnicas utilizadas: Buenas prácticas a la hora de trabajar, manejo de listas y matrices, bucles, librerías para facilitar el desarrollo.

Con respecto a los recursos, estos últimos fueron limitados, pero se logró cumplir con la mayoría de las especificaciones haciendo uso de un buen manejo de tiempo y la adaptación a tecnología limitada para poder desarrollar el proyecto en una computadora no tan avanzada. En la culminación final del proyecto, se deja en claro la necesidad de los editores de imágenes y como la representación abstracta de dibujos desde la antigüedad resulta hoy en día en útiles demostrativos para la enseñanza y desarrollo del intelecto.