

Proyecto - Visor de Imágenes

Resultado de aprendizaje:	<ul style="list-style-type: none">• RAA1. Aplica conceptos de álgebra vectorial y transformaciones geométricas para representar objetos y procesar imágenes digitales en entornos bidimensionales. (RAP1, RAP3)• RAA4. Desarrolla interfaces gráficas interactivas, integrando eventos, controles de usuario y elementos visuales aplicados a soluciones computacionales. (RAP1, RAP2, RAP5)• RAA5. Comunica, documenta y presenta desarrollos gráficos utilizando lenguaje técnico apropiado, argumentación estructurada y herramientas de soporte. (RAP8)
Valor de calificación:	Entre 0.0 y 5.0
Tiempo estimado de desarrollo:	12 h
Dinámica de la actividad:	<ul style="list-style-type: none">• Trabajo individual o en pareja• Desarrollo de una aplicación de escritorio para visualización, transformación y análisis de imágenes• Entrega del código fuente funcional, con documentación y presentación del proyecto
Producto a entregar:	Aplicación funcional desarrollada en Python con interfaz gráfica (Tkinter o Qt), código documentado y manual de usuario
Documentos de Apoyo:	Guías de desarrollo de interfaces gráficas, material del curso sobre procesamiento de imágenes y librerías en Python

Herramientas:	Python, Tkinter o PyQt5, NumPy, Matplotlib
Instrumento de evaluación	Rubrica de evaluación para talleres
Fecha de apertura:	
Fecha de cierre:	

Descripción del Proyecto:

Este proyecto tiene como objetivo desarrollar un visor de imágenes interactivo en Python que permita al usuario aplicar múltiples transformaciones sobre imágenes digitales. Se integran funcionalidades de carga, visualización, edición (brillo, contraste, rotación, filtros), fusión de imágenes, generación de histogramas y exportación de resultados. El proyecto promueve la aplicación integrada de todos los conceptos vistos durante el curso.

Objetivos:

- Desarrollar una interfaz gráfica amigable para la visualización y edición de imágenes.
- Aplicar técnicas de transformación y filtrado de imágenes con NumPy y OpenCV.
- Implementar herramientas de análisis gráfico como histogramas y canales RGB/CMY.
- Gestionar múltiples imágenes, operaciones de fusión y herramientas interactivas (zoom, rotación).
- Documentar adecuadamente la aplicación desarrollada.

Metodología:

El estudiante desarrollará de manera progresiva los componentes del visor, integrando gradualmente las funcionalidades básicas y avanzadas. El proyecto se desarrolla durante las últimas semanas del curso, con revisiones parciales y asesorías periódicas del docente.

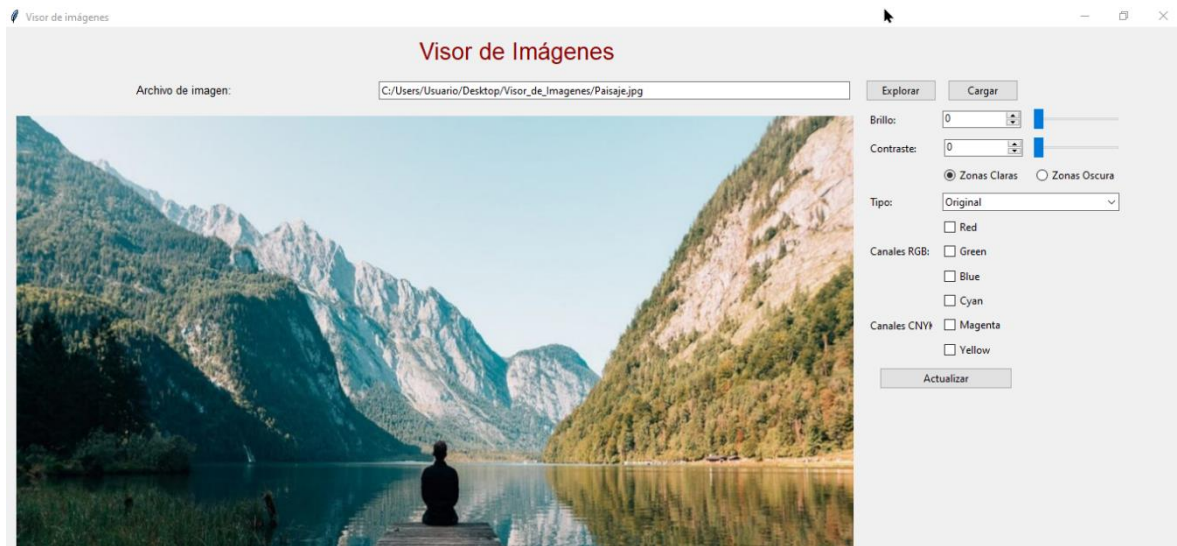
Recursos Necesarios:

- Computador con Python, Tkinter o PyQt5, NumPy, Matplotlib instalados
- Imágenes de prueba en formatos JPG, PNG y BMP
- Editor de código (VS Code, PyCharm, Jupyter)

Evaluación:

La evaluación se basa en la completitud, funcionalidad y calidad técnica del visor, así como en su interfaz, documentación y presentación oral.

Requisitos mínimos del visor:



Requerimientos:

1. Interfaz de Usuario:

- Crear una ventana principal con un título adecuado, p. ej., "Visor de Imágenes".
- Incluir un campo de texto para mostrar la ruta del archivo de imagen cargado.
- Botones:
 - **Explorar:** Abre un cuadro de diálogo para seleccionar una imagen del sistema de archivos.

- **Cargar:** Carga la imagen seleccionada y la muestra en la ventana principal.
- **Actualizar:** Aplica los ajustes seleccionados a la imagen mostrada.
- **Guardar:** Permite guardar la imagen modificada con un nombre nuevo y en una ubicación diferente.
- **Fusionar Imágenes:** Permite seleccionar una segunda imagen y fusionarla con la imagen principal cargada, aplicando una transparencia ajustable.
- Controles deslizantes para ajustar:
 - **Brillo:** Aumentar o disminuir el brillo de la imagen.
 - **Contraste:** Ajustar el contraste de la imagen.
 - **Rotación:** Permite rotar la imagen desde 0° hasta 360°.
 - **Transparencia de Fusión:** Ajustar la transparencia de la segunda imagen (de 0% a 100%) para fusionar con la imagen principal.
- Opciones de filtro de color:
 - **Zonas Claras y Zonas Oscuras:** Permitir la aplicación de un filtro para resaltar las zonas claras u oscuras de la imagen.
 - **Canales RGB:** Permitir al usuario seleccionar o desactivar los canales de color (Rojo, Verde, Azul) de la imagen.
 - **Canales CMY:** Permitir al usuario activar/desactivar los canales Cian, Magenta y Amarillo.
- Funciones adicionales:
 - **Zoom:** Permitir seleccionar un área de la imagen utilizando coordenadas iniciales (X, Y) y ajustar el zoom sobre la región seleccionada.

- **Binarizar Imagen:** Convertir la imagen a blanco y negro utilizando un umbral ajustable.
- **Visualizar Histograma:** Mostrar el histograma de la imagen cargada con la distribución de cada canal de color.
- **Negativo Imagen:** Saca el negativo de una imagen
- **Fusión de Imágenes:** Permitir al usuario cargar una segunda imagen y combinarla con la imagen principal, aplicando un valor de transparencia ajustable.

2. Funcionalidad:

- **Cargar y Mostrar Imágenes:**
 - Permitir al usuario cargar imágenes en formato JPG, PNG y BMP.
 - Mostrar la imagen cargada en el área de visualización de la ventana principal.
- **Ajustes de Imagen:**
 - Aplicar las transformaciones de brillo y contraste utilizando la biblioteca que se está realizando en la clase.
 - Permitir la rotación de la imagen desde 0° hasta 360°.
 - Implementar un zoom sobre la imagen a partir de un punto inicial (X, Y) y un factor de escala determinado por el usuario.
 - Binarizar la imagen aplicando un umbral para convertirla en blanco y negro.
- **Fusión de Imágenes:**
 - Permitir la carga de una segunda imagen desde el sistema de archivos.
 - Ajustar la transparencia de la segunda imagen y fusionarla con la imagen principal.

- La transparencia se ajustará utilizando un control deslizante, variando desde 0 (completamente opaca) hasta 1 (completamente transparente).
- **Visualización del Histograma:**
 - Generar y mostrar un histograma que muestre la distribución de los canales de color (R, G, B) y la intensidad de luminosidad.
- **Actualización Dinámica:**
 - Actualizar la vista de la imagen cada vez que se realice un cambio en los controles.

3. **Detalles de Implementación:**

- Usar tkinter o Qt para la creación de la interfaz gráfica.
- Usar matplotlib para la generación y visualización del histograma.
- Seguir buenas prácticas de programación, separando la lógica de la interfaz de usuario de la manipulación de imágenes.

4. **Criterios de Evaluación:**

- La interfaz gráfica es intuitiva y permite al usuario realizar las operaciones descritas.
- La aplicación es capaz de cargar y mostrar imágenes sin errores.
- Los ajustes de brillo, contraste, rotación y zoom se aplican correctamente a la imagen.
- Los filtros de color y binarización se aplican de manera adecuada y producen los resultados esperados.
- La fusión de imágenes se realiza de manera correcta y visualmente coherente, de acuerdo con el valor de transparencia definido por el usuario.
- El histograma se genera y muestra correctamente en una ventana adicional o en un área designada dentro de la aplicación.

- Se incluyen comentarios y documentación en el código para facilitar la comprensión del mismo.

5. Recomendaciones:

- Utilizar `tkinter.Frame` para organizar los elementos de la interfaz gráfica.
- Crear clases separadas para manejar la lógica del procesamiento de imágenes y la interfaz de usuario.
- Probar la aplicación con diferentes tipos y tamaños de imágenes para asegurar su funcionamiento.
- Implementar la funcionalidad de zoom de manera que permita ajustar tanto la región inicial como el factor de escala.

Funcionalidades Avanzadas (Opcionales):

- Implementar la funcionalidad de **Deshacer Cambios** para restaurar la imagen a su estado original.
- Añadir un botón de **Restaurar** para devolver la imagen a sus valores por defecto (sin ajustes aplicados).
- Incluir la opción de **Rotación Libre**, donde el usuario puede rotar la imagen con el ratón o utilizando el teclado.
- Permitir seleccionar áreas con el ratón para aplicar el zoom dinámicamente.
- Implementar la **fusión de imágenes con diferentes modos de superposición** (e.g., multiplicación, adición, diferencia).

Rúbrica de Evaluación

Criterio	RAA Evalua do	Excelente (4.5 – 5.0)	Bueno (3.5 – 4.4)	Aceptable (2.5 – 3.4)	Insuficiente (0.0 – 2.4)
Interfaz gráfica funcional y amigable	RAA4	Diseño profesional, intuitivo y con navegación clara	Diseño claro, interfaz ordenada	Interfaz simple, desorganizada o poco estética	Interfaz confusa, difícil de usar o sin interacción
Transformacio nes correctas	RAA1	Aplicación completa y precisa de todas las transformacio nes	Transformacio nes funcionales con errores menores	Transformacio nes incompletas o erróneas	Transformacio nes mal implementada s o ausentes
Fusión de imágenes	RAA4	Implementada con control fluido de transparencia	Funciona con leves fallas	Funciona parcialmente	No implementada o errónea
Visualización gráfica (histogramas)	RAA4	Histogramas y filtros visualizados correctamente	Visualización funcional con errores menores	Visualización incompleta o poco clara	Sin visualización o con errores graves
Documentació n del código	RAA5	Comentado adecuadamen te, incluye manual de usuario	Comentarios en secciones clave	Pocos comentarios o documentació n confusa	Sin comentarios o sin manual

Presentación y entrega del proyecto	RAA5	Entrega puntual, presentación clara y completa	Entrega completa con presentación comprensible	Entrega parcial o presentación deficiente	Sin entrega o sin presentación adecuada
-------------------------------------	------	--	--	---	---

Ponderación sugerida por criterio

Criterio	Peso (%)
Interfaz gráfica funcional	20%
Transformaciones correctas	20%
Fusión de imágenes	15%
Visualización gráfica (histogramas)	15%
Documentación del código	10%
Presentación y entrega del proyecto	20%
Total	100%

Elaborado por:

FRANCISCO ALEJANDRO MEDINA