

Universidad Rafael Landívar
Campus San Alberto Hurtado S.J de
Quetzaltenango
Facultad de Ingeniería
Manejo de Archivos



Proyecto del curso

Extractor de GIF'S

Iván Alexander Ordoñez López

1567523

Miguel Antonio Salguero Sandoval

1626923

INTRODUCCIÓN

El Extractor de Datos GIF es una herramienta gráfica desarrollada para simplificar el análisis y la gestión de archivos en formato GIF. Este formato de imagen animada, popular en aplicaciones web y redes sociales, contiene múltiples metadatos que describen su estructura, su colorimetría, la compresión utilizada, entre otros aspectos importantes.

El objetivo del proyecto es proporcionar una plataforma fácil de usar para cargar archivos GIF desde una carpeta o de manera individual, visualizar una previsualización de la animación y acceder a sus metadatos de manera detallada. Estos datos incluyen información técnica (como el tamaño, la paleta de colores, y la compresión) y datos contextuales (como las fechas de creación y modificación). Además, la aplicación permite guardar la información extraída en un archivo de texto, facilitando así su almacenamiento o análisis posterior.

En resumen, el proyecto busca ofrecer una interfaz intuitiva que permita a los usuarios explorar y extraer información detallada de archivos GIF sin la necesidad de herramientas complejas, ayudando tanto a usuarios finales como a desarrolladores y diseñadores gráficos a entender mejor el contenido y estructura de sus archivos GIF.

MANUAL DE USO

Descripción General

El "Extractor de Datos GIF" es una aplicación gráfica que permite cargar archivos GIF desde una carpeta o archivo específico, extraer detalles sobre el archivo y guardar esa información en un archivo de texto. La aplicación se basa en una interfaz gráfica para facilitar su uso y emplea herramientas para la visualización y manipulación de imágenes GIF.

Requisitos Previos

1. **Python:** Asegúrate de tener Python instalado en tu sistema.
2. **Bibliotecas necesarias:** La aplicación utiliza la biblioteca Pillow para gestionar las imágenes. Es recomendable instalar esta biblioteca antes de ejecutar la aplicación.

Iniciar la Aplicación

1. **Ejecutar el archivo principal:**
 - Abre el archivo main.py en tu sistema para iniciar la aplicación.
2. **Pantalla de Inicio:**
 - Al abrir la aplicación, aparecerá una pantalla de bienvenida con el título "Extractor de Datos GIF" y los nombres de los desarrolladores.
 - Haz clic en el botón Iniciar para acceder a la aplicación principal.

Funcionalidades Principales

1. Cargar Archivos GIF

En el panel izquierdo encontrarás varias opciones para cargar archivos o carpetas de GIF:

- **Cargar Carpeta:**
 - Haz clic en el botón **Cargar Carpeta** para seleccionar una carpeta completa.
 - La aplicación buscará todos los archivos GIF en esa carpeta (y subcarpetas) y los listará en el panel.

- **Cargar Archivo:**

- Selecciona **Cargar Archivo** para añadir un archivo GIF específico a la lista de archivos.

Todos los archivos seleccionados aparecerán en la lista de archivos en el panel izquierdo.

2. Mostrar Datos

- Selecciona un archivo GIF de la lista y haz clic en el botón **Mostrar Datos** para visualizar su información en el panel derecho.
- Los datos que se muestran incluyen detalles como:
 - Versión
 - Tamaño
 - Colores
 - Compresión
 - Formato Numérico
 - Color de Fondo
 - Número de Imágenes (fotogramas)
 - Fecha de Creación
 - Fecha de Modificación
 - Comentarios

3. Vista Previa del GIF

- Al seleccionar un archivo GIF y hacer clic en **Mostrar Datos**, también se mostrará una **vista previa del GIF** animada en el panel derecho.
- La animación cambiará de fotogramas automáticamente.

4. Guardar Datos en Archivo de Texto

- Si deseas guardar los datos de uno o varios archivos GIF, haz clic en **Guardar en TXT**.
- Selecciona la ubicación y el nombre del archivo donde quieres almacenar la información.

- La aplicación generará un archivo de texto que contiene los datos de cada archivo GIF cargado.

5. Salir de la Aplicación

- Para cerrar la aplicación, haz clic en **Salir** en el panel izquierdo, o simplemente cierra la ventana.

Solución de Problemas Comunes

1. Error al cargar un archivo GIF:

- Asegúrate de que el archivo seleccionado es un GIF. Otros formatos de imagen no son compatibles.

2. No se muestra la vista previa:

- Puede ser un problema con el tamaño del GIF o con la biblioteca Pillow. Es recomendable asegurarse de que esta biblioteca esté actualizada.

3. La aplicación no se abre:

- Verifica que Python esté correctamente instalado y que se hayan seguido los requisitos previos.

ESTRUCTURAS DE ALMACENAMIENTO

1. Metadatos de los Archivos GIF:

- **Tipo de datos:** Los metadatos se almacenan en un diccionario de Python, donde cada clave representa un atributo del GIF (por ejemplo, versión, tamaño, colores, compresión, etc.) y cada valor representa la información correspondiente.
- **Función de almacenamiento:** Los metadatos se extraen y almacenan en listas y diccionarios temporales en la memoria durante la ejecución.
- **Agregar:** Cuando se carga un archivo GIF, la función `obtener_datos_gif()` extrae los datos y los almacena en la lista `archivos_datos` de la aplicación principal.
- **Editar:** Los datos son de solo lectura; no se pueden editar.
- **Eliminar:** Los datos se eliminan de la memoria al cerrar la aplicación o al borrar la lista de archivos cargados.

2. Archivos de Salida (TXT):

- **Tipo de datos:** Los datos se guardan como texto en formato `.txt`.
- **Función de almacenamiento:** La función `guardar_datos_txt()` se encarga de crear el archivo y escribir los metadatos de cada GIF cargado.
- **Agregar:** Cada vez que se guarda un archivo, se añaden los datos de todos los GIFs procesados en un archivo de texto.
- **Editar:** Los archivos `.txt` generados pueden editarse externamente con cualquier editor de texto.
- **Eliminar:** Los archivos se pueden eliminar manualmente desde la carpeta `data/` o en el sistema de archivos.

3. Interfaz Gráfica:

- **Tipo de datos:** Los elementos de la interfaz (etiquetas, cuadros de texto, listas) son componentes de la biblioteca `tkinter`.
- **Función de almacenamiento:** La interfaz es transitoria y se almacena en la memoria mientras la aplicación está abierta.

- **Agregar:** Los elementos de la interfaz se añaden al iniciar la aplicación.
- **Editar:** Los elementos de la interfaz se actualizan dinámicamente según las interacciones del usuario (por ejemplo, al cargar una carpeta o archivo, la lista de archivos se actualiza).
- **Eliminar:** Los elementos de la interfaz se destruyen al cerrar la aplicación.

4. Vista Previa del GIF:

- **Tipo de datos:** Las imágenes de los fotogramas del GIF se almacenan en objetos de ImageTk.PhotoImage.
- **Función de almacenamiento:** Los fotogramas se almacenan en una lista en memoria, la cual se utiliza para la animación en la vista previa.
- **Agregar:** Los fotogramas del GIF se añaden a la lista gif_frames cada vez que el usuario selecciona un GIF para previsualizar.
- **Editar:** Los fotogramas no se pueden editar; solo se visualizan en un bucle de animación.
- **Eliminar:** La lista de fotogramas se vacía o elimina al cambiar de archivo o cerrar la aplicación.

Proceso de Gestión de Datos

- **Cargar datos:** Los datos de los GIF se cargan desde el sistema de archivos (carpetas o archivos individuales) mediante diálogos de selección de filedialog de tkinter.
- **Mostrar datos:** Los datos extraídos se presentan en un cuadro de texto, permitiendo al usuario visualizar la información detallada del GIF seleccionado.
- **Guardar datos:** La opción de guardar permite exportar los metadatos de todos los GIF cargados en un archivo de texto ubicado en data/.

Esta estructura asegura que los datos sean temporales (en memoria) o permanentes (guardados en archivos .txt), y facilita el manejo de los GIF en un entorno sencillo y ordenado.

ALGORITMOS UTILIZADOS

Algoritmo de Extracción de Metadatos de GIF

Este algoritmo se encarga de extraer información relevante de un archivo GIF, como la versión, el tamaño, los colores, la compresión y otros atributos.

Entrada: Ruta del archivo GIF.

Salida: Diccionario con los metadatos del archivo.

Pseudocódigo:

1. Abrir el archivo GIF en modo lectura.
2. Extraer los atributos principales:
 - Versión del GIF
 - Tamaño en píxeles
 - Número de colores
 - Tipo de compresión (si existe)
 - Formato numérico (modo de color)
 - Color de fondo
 - Número de imágenes (fotogramas) en el GIF
 - Fecha de creación y última modificación
 - Comentarios (si existen)
3. Almacenar los atributos en un diccionario.
4. Retornar el diccionario con los metadatos.

Implementación: Esta función es obtener_datos_gif().

2. Algoritmo de Búsqueda de Archivos GIF en una Carpeta

Este algoritmo recorre una carpeta y sus subcarpetas en busca de archivos con extensión .gif.

Entrada: Ruta de la carpeta seleccionada.

Salida: Lista de rutas de archivos GIF encontrados.

Pseudocódigo:

1. Inicializar una lista vacía `archivos_gif`.
2. Para cada subcarpeta y archivo en el directorio:
 - Si el archivo tiene extensión `.gif`, agregar su ruta a `archivos_gif`.
3. Retornar la lista `archivos_gif`.

Implementación: Esta función es `buscar_archivos_gif()`.

3. Algoritmo de Guardado de Datos en un Archivo de Texto

Este algoritmo guarda los metadatos de cada GIF cargado en un archivo de texto.

Entrada: Lista de metadatos de archivos y ruta del archivo de texto de salida.

Salida: Archivo `.txt` con la información de los GIF.

Pseudocódigo:

1. Abrir el archivo en modo escritura.
2. Para cada archivo GIF y sus metadatos:
 - Escribir el nombre del archivo.
 - Escribir cada atributo (clave) y su valor.
 - Añadir una separación (línea de guiones) entre los registros de diferentes GIF.
3. Cerrar el archivo.

Implementación: Esta función es `guardar_datos_txt()`.

4. Algoritmo de Animación para la Vista Previa de GIF

Este algoritmo permite la visualización en bucle de los fotogramas de un archivo GIF en la interfaz gráfica.

Entrada: Ruta del archivo GIF.

Salida: Visualización del GIF en un widget de la interfaz.

Pseudocódigo:

1. Cargar el archivo GIF y extraer cada fotograma.

2. Redimensionar y convertir cada fotograma para su uso en tkinter.
3. Almacenar los fotogramas en una lista gif_frames.
4. Establecer un índice inicial frame_index en 0.
5. Iniciar un bucle para mostrar los fotogramas:
 - Mostrar el fotograma actual.
 - Incrementar frame_index en 1 (o resetearlo si es el último fotograma).
 - Esperar un breve periodo (para el efecto de animación) y repetir.
6. Continuar hasta que el usuario cambie de GIF o cierre la aplicación.

Implementación: La función principal de esta animación es mostrar_gif() junto con actualizar_gif().

5. Algoritmo para la Interfaz Gráfica (Interacción del Usuario)

Este algoritmo describe la lógica de control y los eventos de la interfaz gráfica que permiten al usuario interactuar con la aplicación.

Eventos Principales:

1. **Cargar Carpeta:**
 - Activado cuando el usuario selecciona "Cargar Carpeta".
 - Ejecuta el algoritmo de búsqueda de GIF en la carpeta seleccionada y muestra los resultados en la lista de archivos.
2. **Cargar Archivo:**
 - Activado cuando el usuario selecciona "Cargar Archivo".
 - Ejecuta el algoritmo de extracción de metadatos para el archivo seleccionado y lo agrega a la lista de archivos.
3. **Mostrar Datos:**
 - Activado cuando el usuario selecciona un archivo en la lista y elige "Mostrar Datos".

- Muestra los metadatos del archivo GIF seleccionado en el cuadro de texto.
- Ejecuta el algoritmo de animación para mostrar una vista previa del GIF.

4. **Guardar en TXT:**

- Activado cuando el usuario selecciona "Guardar en TXT".
- Ejecuta el algoritmo de guardado, almacenando los metadatos de los archivos GIF en un archivo de texto.

5. **Salir:**

- Cierra la aplicación.

Implementación: Estos eventos son manejados por métodos específicos en la clase principal AplicacionGIFExtractor, como cargar_carpeta(), cargar_archivo(), mostrar_datos(), y guardar_txt().

CONCLUSIONES

1. Funcionalidad y Utilidad: El proyecto "Extractor de Datos GIF" es una herramienta eficaz para la extracción y visualización de metadatos de archivos GIF. La capacidad de obtener y visualizar detalles técnicos de los archivos GIF, como la cantidad de colores, el tamaño, el número de fotogramas, entre otros, resulta valiosa para quienes necesitan manipular o analizar archivos GIF, como diseñadores gráficos, desarrolladores, y usuarios que trabajan con contenido multimedia.

2. Facilidad de Uso: La interfaz gráfica desarrollada con tkinter permite que el usuario interactúe de forma intuitiva con la aplicación. El diseño de una pantalla de inicio, así como los botones de carga, visualización, y guardado, hacen que el proceso de uso sea sencillo y accesible, sin requerir conocimientos técnicos avanzados.

3. Organización Eficiente de la Información: La estructura del programa permite una organización clara de los datos. La función de almacenamiento en un archivo de texto (.txt) permite guardar los metadatos de los archivos GIF de manera estructurada y legible, facilitando la posterior consulta y revisión de la información.

4. Animación y Visualización: La implementación de un algoritmo que permite la visualización de la animación en los archivos GIF fue un logro importante. Esta característica mejora la experiencia del usuario al permitirle ver una vista previa del GIF seleccionado, algo especialmente útil para validar y revisar archivos rápidamente.

5. Modularidad y Escalabilidad: La estructura modular del código, organizada en funciones independientes, facilita la lectura y el mantenimiento del programa. Además, esta estructura permite que el proyecto sea escalable, pudiéndose adaptar para incluir nuevos tipos de análisis o formatos de archivos sin necesidad de modificar toda la aplicación.

6. Limitaciones Técnicas y Potencial de Mejora: A pesar de sus funcionalidades, el programa depende de la biblioteca PIL para la manipulación de imágenes, lo cual puede presentar ciertas limitaciones en el procesamiento de archivos GIF más complejos. Una futura implementación en una biblioteca de procesamiento de imágenes más avanzada, o la inclusión de otras características como filtros o editores de GIF, podría ampliar aún más la utilidad de esta aplicación.

7. Aplicación en Diferentes Contextos: Este proyecto puede ser útil en contextos educativos, como herramienta para enseñar sobre la manipulación de archivos de imagen y la extracción de metadatos. Además, en el ámbito profesional, esta aplicación puede servir para análisis y optimización de archivos GIF, especialmente en áreas relacionadas con el diseño y el desarrollo de contenidos web.

