

Manejo e implementación de archivos

Proyecto

Extractor de datos de archivos GIF

Descripción del proyecto

El proyecto será una aplicación gráfica que permita obtener la información de los archivos de imágenes GIF (Graphics Interchange Format), para lo cual hay que leer adecuadamente estos archivos, conocer su estructura y obtener la información que se guarda en ellos.

Todos los archivos GIF tienen información como:

- Número de versión
- Tamaño de imagen
- Cantidad de colores
- Tipo de compresión de imagen
- Formato numérico
- Color de fondo
- Cantidad de imágenes que contiene
- Fecha de creación
- Fecha de modificación
- Comentarios agregados

La información completa de la estructura de los archivos GIF puede encontrarse en:

- Página oficial sobre la estructura de archivos GIF <https://www.w3.org/Graphics/GIF/spec-gif89a.txt>
- Información en Wikipedia sobre datos generales de información contenidos en un archivo GIF: <https://en.wikipedia.org/wiki/GIF>
- Explicación byte por byte de cada parte de un archivo GIF: http://www.matthewflickinger.com/lab/whatsinagif/bits_and_bytes.asp
- Página con explicación de las estructuras en cada parte de un archivo GIF: <http://www.fileformat.info/format/gif/egff.htm>

La aplicación debe tener la capacidad de leer varios archivos GIF que se encuentren en una carpeta e incluso en carpetas dentro de la primera, lo que indica que hace una búsqueda de archivos GIF.



Toda la información recopilada de todos los archivos GIF debe de almacenarse en un archivo, que será utilizado solamente por su proyecto, así evitar tener que leer todos los datos de esos archivos cada vez que se abre la aplicación.

La información deberá tener al menos, por cada archivo GIF encontrado:

- Número de versión
- Tamaño de imagen
- Cantidad de colores
- Tipo de compresión de imagen
- Formato numérico
- Color de fondo
- Cantidad de imágenes que contiene
- Fecha de creación
- Fecha de modificación
- Comentarios agregados

Descripción del funcionamiento de la aplicación:

Supondremos un caso de uso de la aplicación:

La aplicación se inicia:

Un usuario normal inicia su aplicación, si es la primera vez que se corre esa aplicación, deberá aparecer una ventana para agregar una ruta o path. Para saber si es la primera vez que se ejecuta, lo único que debe de hacer es verificar si existe el archivo donde guardará su información; ya que, si no existe, quiere decir que nunca se ha ejecutado en esa máquina.

La aplicación agrega una ruta:

Un usuario desea agregar una carpeta con archivos GIF o se inicia por primera vez, debe aparecer una ventana donde el usuario pueda elegir una carpeta y una vez que la haya seleccionado, la aplicación debe de buscar todos los archivos GIF que existan en esa y todas las carpetas que contenga, realizando así una búsqueda recursiva.

La información que obtenga de los archivos GIF debe de ser almacenada en el archivo de la aplicación.

Ver la información de los archivos GIF:

Por cada archivo GIF que se tenga, el programa debe poder mostrar en algún lugar la lista de archivos encontrados organizados por la carpeta en la que fueron encontrados. Al seleccionar cada uno de estos archivos la aplicación debe mostrar toda la información recopilada.

Editando información recopilada:

La aplicación debe poder permitir alterar estos datos, no en el archivo GIF, pero si en la información que ha encontrado y que luego guardará en un archivo.

Opciones y restricciones para el proyecto

Los **equipos de trabajo estarán compuestos por dos personas**, como máximo.

Todos los integrantes del equipo de trabajo deben colaborar, todo se verá reflejado en un repositorio de GIT donde todos deben aportar código, si alguno no participa o participa muy poco en relación a los demás no tendrá la nota completa del proyecto.

La información puede guardarse en uno o en más archivos, así si el equipo de trabajo decidirá cómo quiere almacenar la información.

No se permite el uso de bibliotecas externas para manipulación de archivos GIF, tanto para almacenamiento como para leer el formato mencionado, por lo que todo el trabajo debe realizarse con lo único que provee el lenguaje Python, si se hace caso omiso a esta regla el **proyecto será anulado**.

Para cualquier otra función de la aplicación si se pueden utilizar bibliotecas, por ejemplo, para la realización de la interfaz gráfica (por ejemplo, TKinter) o para poder escribir texto con formato. Al utilizar estas bibliotecas se debe avisar por medio de un correo a JPMONROYRI@correo.url.edu.gt si es o no posible utilizarlo; en caso de no informar de su uso, también será invalidado el proyecto.

Reporte del proyecto:

El proyecto debe de tener un reporte, el cual será entregado de manera impresa, éste reporte debe tener el siguiente contenido:

1. Introducción
2. Manual de uso de la aplicación: Debe contener información de cómo utilizar la aplicación, en un orden lógico, llevando al lector a entender cómo debe funcionar la aplicación y no simples capturas de imagen con explicación de que es cada sección de una pantalla.
3. Estructura de almacenamiento: Debe de contener información detallada de cómo se almacena la información que se utiliza en la aplicación, describiendo los tipos de datos y como es que se agrega, borra y edita la información. Puede tomar como ejemplo el siguiente enlace: <https://www.datavoyage.com/mpgscript/mpeghdr.htm>
4. Algoritmos utilizados: Será un texto con información de algoritmos utilizados, por ejemplo, para ordenar la información o bien para buscar, detallando cómo se utilizaron y no solo indicando los nombres de los algoritmos
5. Conclusiones

Se recomienda utilizar Google Docs u Office 365 para realizar éstos documentos, ya que debe de ir enviando actualizaciones al catedrático para su debida corrección, antes de la entrega final y así asegurarse de obtener la nota completa. De igual forma estas plataformas permiten el trabajo colaborativo, por lo que también podrá revisarse el aporte de cada integrante del equipo en el documento.



Aspectos a calificar:

1. Funcionamiento
 1. Lectura de carpetas
 2. Lectura de información de archivos GIF
 3. Edición de datos
2. Organización de Información
 1. Estructura del archivo donde se almacena la información
 2. Facilidad de acceso a la información para editar o borrar datos
3. Búsqueda y despliegue de la información
 1. Algoritmos de búsqueda utilizados
 2. Interfaz amigable para uso
4. Del reporte
 1. Redacción
 2. Información adecuada para cada punto solicitado

Forma de entrega del proyecto:

Cada proyecto deberá contener su código fuente; el proyecto deberá entregarse en clase y deberá contener unos archivos de ejemplo.

Se deberá adjuntar un documento llamado README.txt donde se indiquen datos como:

- Nombre y carné de los estudiantes.
- Nombre y versión de SO utilizado para la ejecución
- Algunos comentarios extras sobre como ejecutar el programa o formas de ejecutarlo, instrucciones de prueba, etc., en caso de ser necesario.

Debe ser entregado en un repositorio de GIT.

Todos los proyectos serán revisados el día de la entrega para corroborar su correcto funcionamiento, para luego proceder a calificarlo detalladamente y asignar una nota final.

Fecha de entrega:

- **Primera fase:** miércoles 2 de octubre.
Para la primera entrega, el proyecto debe contar con lectura de todos los datos del formato GIF, no es necesario que se guarden aún, pero debe poder leer uno o más archivos GIF.
- **Segunda fase:** miércoles 30 de octubre.
Para la segunda fase, debe contener absolutamente todo lo requerido en este documento.