

Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

IC-1801 Taller de Programación

Prof. Mauricio Avilés

Proyecto Programado – Graficador de Espacio en Disco

Introducción

Con el avance de la tecnología, los dispositivos de almacenamiento secundario aumentan su capacidad incesantemente. Tal como su capacidad aumenta, así de intensivo es el uso que se les da. Es una situación común de los usuarios el darse cuenta que el espacio libre en disco se está agotando sin saber exactamente por qué. Ahí es cuando entran en juego los graficadores de espacio en disco; estos son un tipo de software que se encarga de ilustrar gráficamente el espacio del disco. Lo que estos programas hacen es analizar los archivos en disco, sean videos, instaladores, música y demás, y presentan uno o más reportes que clarifican la forma en que está siendo utilizado el espacio en disco.

Software a desarrollar

Su trabajo consiste en crear un graficador de espacio en disco utilizando el lenguaje de programación Python. Seguidamente, se detallan las características con las que debe contar el programa.

Requisitos funcionales:

1. Solicitar al usuario el directorio (carpeta o folder) donde se inicia el análisis de espacio.
2. Mostrar un gráfico que ilustre cómo se distribuye el tamaño del contenido de dicho directorio.
3. Debe mostrarse la distribución del tamaño del directorio y todos sus subdirectorios.
4. En la representación debe utilizarse un máximo de 6 niveles de profundidad. Esto no significa que el análisis no deba hacerse hasta la profundidad máxima.
5. Para cada directorio debe mostrarse nombre y tamaño. El tamaño debe presentarse utilizando la unidad más adecuada, ya sea KB, MB, GB o TB.
6. El gráfico debe hacer uso de diferentes colores para diferenciar los elementos.
7. Lista de los 10 archivos más grandes (mostrar ubicación completa, nombre y tamaño)
8. Lista de los 10 directorios con mayor cantidad de archivos directamente ubicados en el directorio (mostrar ubicación completa, nombre y cantidad)

Requisitos no funcionales:

1. El programa debe programarse utilizando Python con la biblioteca gráfica Pygame para representar los datos.
2. Debe usarse el manejo de errores apropiado para que el programa sea robusto.
3. El programa debe hacer uso de recursión para el análisis de los elementos en disco.

A continuación, se muestra un ejemplo con una posible apariencia para el programa a desarrollar. La forma de representar el análisis del espacio no necesariamente debe ser igual, el equipo de trabajo puede idear una representación propia siempre y cuando se cumpla con lo solicitado. Nótese que esta primera representación impide que se pueda mostrar el nombre de todas las carpetas analizadas, pero se muestra en las que es posible.

Para la implementación de este proyecto es necesario utilizar algunas funciones para obtener información del sistema de archivos del sistema operativo. Estas funciones se encuentran en los módulos **os** y **os.path**. La documentación completa de estos módulos puede encontrarse en las siguientes direcciones: <https://docs.python.org/3/library/os.html> y <https://docs.python.org/3/library/os.path.html#module-os.path>.

A continuación, se detallan algunas funciones importantes:

- `os.listdir(ruta)`: recibe un *string* con la ruta de una carpeta y retorna como resultado una lista de *strings* con los elementos que se encuentran en la carpeta, incluyendo directorios y archivos.
- `os.path.join(ruta, nombre)`: recibe un *string* con una ruta y el nombre de un elemento. Retorna como resultado la ruta absoluta del elemento, que es la unión de la ruta con el nombre del elemento.
- `os.path.isfile(ruta_absoluta)`: recibe un *string* con la ruta absoluta de un elemento. Retorna como resultado `True` si dicho elemento es un archivo, de otro modo retorna `False`.
- `os.path.isdir(ruta_absoluta)`: recibe un *string* con la ruta absoluta de un elemento. Retorna como resultado `True` si dicho elemento es un directorio, de otro modo retorna `False`.
- `os.path.getsize(ruta_absoluta)`: recibe un *string* con la ruta absoluta de un archivo. Retorna como resultado un entero con el tamaño en bytes del archivo.
- `os.path.basename(ruta_absoluta)`: recibe un *string* con la ruta absoluta de un elemento. Retorna como resultado el nombre del elemento sin la ruta.

Debe tomarse en cuenta que estas funciones lanzan una excepción si la ruta no es válida o hay alguna restricción para utilizarla (permisos). Debe utilizarse el manejo de errores adecuado cuando se invoquen.

Documentación

Toda subrutina debe llevar como documentación interna comentarios con lo siguiente:

1. Descripción de la función
2. Entradas
3. Salidas
4. Restricciones

En cuanto a la documentación externa, debe entregarse un manual de usuario en formato PDF que contenga los siguientes puntos:

1. Explicación del propósito del programa.
2. Instrucciones para la instalación del programa y cualquier requisito que tenga. Con imágenes de ejemplo.
3. Instrucciones de utilización de todas las características del programa. Con imágenes de ejemplo.

Forma de trabajo

El proyecto será desarrollado en equipos de dos o tres personas. No se aceptarán trabajos individuales, salvo previa autorización del profesor y con alguna justificación especial que lo amerite.

Entrega

El tiempo asignado para la tarea programada es de 2 semanas. Debe entregarse un archivo **archivo ZIP** que contenga lo siguiente:

1. PDF con el manual del usuario
2. Código fuente del programa en Python

Evaluación

El proyecto tiene un valor de 20% de la nota final, en el rubro de Proyectos Programados.

Desglose de la evaluación del proyecto:

Documentación: 20%

Programación: 80%

Recomendaciones adicionales

Pruebe cada subrutina de código individualmente. No implemente todo el programa sin verificar el funcionamiento por separado de cada una de sus partes. Esto dirige a errores que son más difíciles de encontrar.

Recuerde que el trabajo es en equipos, es indispensable la comunicación y la coordinación entre los miembros del subgrupo.

Comparta el conocimiento con los demás compañeros de grupo y de la carrera, la ciencia de la computación es una disciplina que requiere el traspaso libre de conocimientos. Se logran mejores resultados con la colaboración de todos que con el esfuerzo separado de diferentes personas.

No dude en consultar diferentes fuentes para satisfacer las dudas. Aparte de las búsquedas en internet, asegúrese de exponer sus dudas a sus compañeros, profesor y conocidos que estudien la carrera; en la mayoría de las ocasiones es más provechosa conversación de 10 minutos entre personas que están trabajando en lo mismo que pasar horas buscando la respuesta a una duda de forma individual.

No deje la documentación para el final, es buena práctica ir desarrollándola durante todo el transcurso del proyecto. Recuerde que la documentación debe ser concisa y puntual, por lo que en realidad no toma mucho tiempo al realizarla de esta forma.

Plagios no serán tolerados bajo ninguna circunstancia. Cualquier intento de fraude será evaluado con una nota de cero y se enviará una carta al expediente del estudiante. Siempre escriba su propio código.