

Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías
Ingeniería en computación
Seminario de solución de problemas de sistemas operativos
NRC 164138
Sección D07
Profesor Javier Rosales Martínez
Diego Armando Sánchez Rubio 217570609
Reporte Práctica 2.

Antecedentes.

En esta práctica se seguirá con el trabajo de procesamiento por lotes, ya que nuevamente se realizará una programa que trabaje de forma repetitiva el mismo proceso, en esta ocasión la práctica consistirá en un pequeño programa que simulará ser un antivirus, en dónde dentro de una series de carpetas se buscará un archivo con una extensión en particular y el programa debe ser capaz de borrarlo en cada una de las carpetas.

Para esta ocasión la extensión del archivo corresponderá a un “.png”, a su vez habrá otros archivos con extensiones diferentes cómo .txt .xlsx, etc.

Metodología.

En este caso el proceso es muy fácil, ya que se utilizará una librería que simplificará mucho más este proceso la cuál es “os” ya qué funciona para interactuar con el sistema operativo, como listar archivos y eliminar archivos, también se usará la librería “time” que en este caso sólo funcionará para algo que explicaré más adelante.

El programa funciona de una manera muy sencilla, primero se tiene una función llamada eliminarArchivos que es la que se encarga de la lógica para eliminar los archivos de las carpetas, esta funciona de la siguiente manera, recibe una ruta es que de dónde se va a empezar a leer y eliminar los archivos, después usa un ciclo For que hace un recorrido en todas las carpetas dentro de la ruta que se le da, esto con la función “os.walk(ruta)”, los atributos carpetaRaiz es la carpeta actual en la que se itera, después el guion bajo corresponde a una lista de las subcarpetas de la carpeta raíz y archivos es una lista con los nombres de los archivos dentro de la carpeta raíz. Después con otro For se hace un recorrido en los archivos leídos en cada carpeta y con un if se valida que si termina con “.png” entra al algoritmo que va a eliminar el archivo, en este caso lo que se hace es guardar en una variable la ruta de dónde se encuentra el archivo, para esto se usa la función path y join de la librería os, que lo que va a hacer y agarrar la dirección de la carpeta y el archivo que se quiere borrar, después se mostrará un mensaje que dice qué está eliminando y mostrará la dirección del archivo, por ultimo con remove se borrará el archivo que se encontró.

```
tabnine | Edit | Test | Explain | Document
def eliminarArchivos(ruta):
    for carpetaRaiz, _, archivos in os.walk(ruta):
        for archivo in archivos:
            if archivo.endswith(".png"):
                archivoPath = os.path.join(carpetaRaiz, archivo)
                print(f"Eliminando: {archivoPath}")
                os.remove(archivoPath)
```

La otra función que se tiene corresponde a el monitoreo de archivos de las carpetas, lo que hace esta función es obtener la ruta actual dónde está el código que se está ejecutando, esto con la función getcwd() de la librería os, después se usa un ciclo While infinito que lo qué hará es mandar a llamar al función anterior, después esperará 5 segundos hasta volver a repetir el proceso, para esto usamos la librería time y la función sleep y le indicamos el tiempo.

```

labnine | Edit | Test | Explain | Document
✓ def monitorearEliminar():
    rutaScript = os.getcwd()
    while True:
        eliminarArchivos(rutaScript)
        time.sleep(5) # Espera 5 segundos antes de volver a escanear

```

Para finalizar en el main simplemente se muestra un mensaje de inicio y se llama a la función anterior para empezar con la tarea.

```

if __name__ == "__main__":
    print("Inicio")
    monitorearEliminar()

```

Conclusiones.

Esta práctica fue bastante sencilla, ya que todo lo hace la misma librería `os` y de esta manera es más fácil manipular los archivos y lidiar con ellos, además de ser un algoritmo muy sencillo para buscar y eliminar los archivos, como todo programa creo que se puede mejorar aunque no sabría cómo hacerlo más optimo, aun así esta versión es bastante funcional.

Referencias.

os — Interfaces misceláneas del sistema operativo — documentación de Python - 3.10.16.

(n.d.). <https://docs.python.org/es/3.10/library/os.html>

time — Time access and conversions. (n.d.). Python Documentation.

<https://docs.python.org/3/library/time.html>