



**Mario Enrique Montes Jacuinde**

**Seminario de solución de problemas de sistemas operativos**

**Reporte practica 2**

**Javier Rosales Martínez**

**D06**

### **Antecedentes:**

Se debía de implementar un código que recibiera una ubicación de una carpeta y también a todos los archivos de la carpeta.

Se debe de identificar una extensión como un archivo maligno y se tiene que eliminar.

Se debe de poder recibir mas archivos con las mismas o diferentes extensiones y que la carpeta pueda seguir trabajando normal.

Se debe de hacer que si se vuelve a poner una carpeta con la extensión de un archivo maligno la elimine.

### **Metodologías:**

**Librerías utilizadas:** se usó la librería os para que se pudiera verificar si el archivo de entrada existe en la ruta especificada antes de que fuera procesado.

Se uso en este caso también la librería watchdog la cual nos ayuda a que pueda monitorear los cambios que se puedan hacer en el sistema en tiempo real, junto con esto también se utilizaron los componentes Observer esto para vigilar el directorio y la clase FileSystemEventHandler para que pueda interceptar los eventos.

**Funciones utilizadas:** La función eliminar(folder\_path) recorre los archivos que hay en la carpeta y elimina los que tienen la extensión especificada.

La función monitorear nos ayuda a que si hay alguna archivo nuevo o modificado con las especificaciones de la extensión lo elimina.

La parte de monitorear\_carpeta hace que se inicialice el monitoreo eliminando en tiempo real y ejecutándose como un ciclo.

**Diseño:** El programa se hizo para que se pudiera ejecutar de manera continua y así cumplir con las especificaciones de la practica.

### **Conclusión:**

El código nos da una solución dadas las especificaciones de la practica tanto para monitorear como para eliminar los archivos malignos que se puedan tener.

## Código:

```
import os
import time
from watchdog.observers import Observer
from watchdog.events import FileSystemEventHandler

extension = ['.bat']

def eliminar(folder_path):
    for root, dirs, files in os.walk(folder_path):
        for file in files:
            file_path = os.path.join(root, file)
            if any(file.endswith(ext) for ext in extension):
                print(f"Eliminando archivo maligno: {file_path}")
                os.remove(file_path)

class monitorear(FileSystemEventHandler):
    def creado(self, event):
        if not event.is_directory:
            file_path = event.src_path
            # Si el archivo recién creado tiene una extensión maligna, se elimina
            if any(file_path.endswith(ext) for ext in extension):
                print(f"Archivo maligno detectado y eliminado: {file_path}")
                os.remove(file_path)

    def modificado(self, event):
        if not event.is_directory:
            file_path = event.src_path
            # Si el archivo modificado tiene una extensión maligna, se elimina
            if any(file_path.endswith(ext) for ext in extension):
                print(f"Archivo maligno modificado detectado y eliminado: {file_path}")
                os.remove(file_path)

def monitorear_carpeta(carpeta):
    event_handler = monitorear()
    observer = Observer()
    observer.schedule(event_handler, folder_path, recursive=True)

def monitorear_carpeta(carpeta):
    event_handler = monitorear()
    observer = Observer()
    observer.schedule(event_handler, folder_path, recursive=True)
    observer.start()

    try:
        while True:
            eliminar(carpeta)
            time.sleep(10)
    except KeyboardInterrupt:
        observer.stop()

    observer.join()

if __name__ == "__main__":
    folder_path = input("Ingresar la ruta de la carpeta a monitorear: ")
    if os.path.isdir(folder_path):
        print(f"Monitoreando la carpeta: {folder_path}")
        monitorear_carpeta(folder_path)
    else:
        print("La ruta ingresada no es válida.")
```

## Referencias:

Python, R. (2022, March 12). *Detectar cambios en tiempo real en archivos (Watchdog) - Recursos*

*Python*. Recursos Python. <https://recursospython.com/guias-y-manuales/detectar-cambios-en-tiempo-real-en-el-sistema-de-archivos-watchdog/>