

# Algoritmo para actualizaciones de archivos en Python

## Descripción del proyecto

Este algoritmo de Python tiene como objetivo automatizar el proceso de actualización de una lista de direcciones IP autorizadas para acceder a contenido restringido en un entorno médico. El script lee un archivo de texto que contiene una lista de direcciones IP permitidas y otra lista con las direcciones IP que deben ser revocadas. A continuación, compara ambas listas y elimina las direcciones IP que coincidan, actualizando el archivo original con la lista filtrada.

Esta automatización garantiza que solo el personal autorizado pueda acceder a la información sensible de los pacientes, mejorando así la seguridad de los datos médicos,

## Abrir el archivo con la lista de permisos

```
import_file = "allow_list.txt"

with open(import_file, 'r') as file:

    # El resto del código irá aquí
```

### Explicación:

- **import\_file = "allow\_list.txt":** Asignamos el nombre del archivo a una variable para facilitar su uso.
- **with open(import\_file, 'r') as file::** Abrimos el archivo en modo lectura ('r'). El bloque with se encarga automáticamente de cerrar el archivo al finalizar, incluso si ocurre una excepción.

## Leer el contenido del archivo

```
with open(import_file, 'r') as file:

    ip_addresses = file.read()
```

### Explicación:

- **ip\_addresses = file.read():** Leemos todo el contenido del archivo y lo almacenamos en la variable ip\_addresses.

## Convertir la cadena en una lista

```
with open(import_file, 'r') as file:
    ip_addresses = file.read()
    ip_addresses = ip_addresses.split('\n')
```

### Explicación:

- **ip\_addresses = ip\_addresses.split('\n'):** Dividimos la cadena en una lista, utilizando el salto de línea como separador. Esto asume que cada dirección IP está en una línea diferente.

## Iterar a través de la lista de eliminación

```
remove_list = ["192.168.1.100", "10.0.0.1"] # Ejemplo de lista de eliminación
```

```
with open(import_file, 'r') as file:
    ip_addresses = file.read()
    ip_addresses = ip_addresses.split('\n')
```

```
for ip in remove_list:
    # Código para eliminar la IP si está en la lista de autorizaciones
```

### Explicación:

- **remove\_list = ["192.168.1.100", "10.0.0.1"]:** Creamos una lista de ejemplo con las direcciones IP a eliminar.

## Eliminar direcciones IP que están en la lista de eliminación

```
remove_list = ["192.168.1.100", "10.0.0.1"]
```

```
with open(import_file, 'r') as file:
    ip_addresses = file.read()
    ip_addresses = ip_addresses.split('\n')
```

```
for ip in remove_list:
    if ip in ip_addresses:
        ip_addresses.remove(ip)
        print(f"Removed IP: {ip}") # Mensaje opcional para verificar
```

#### Explicación:

- **if ip in ip\_addresses::** Verificamos si la IP a eliminar está en la lista de autorizaciones.
- **ip\_addresses.remove(ip):** Eliminamos la IP si se encuentra.
- **print(f"Removed IP: {ip}"): Imprimimos un mensaje opcional para verificar las eliminaciones.**

## Actualizar el archivo con la lista revisada de direcciones IP

```
remove_list = ["192.168.1.100", "10.0.0.1"]
```

```
with open(import_file, 'r') as file:
    ip_addresses = file.read()
    ip_addresses = ip_addresses.split('\n')
```

```
for ip in remove_list:
    if ip in ip_addresses:
        ip_addresses.remove(ip)
```

```
with open(import_file, 'w') as file:
    file.write('\n'.join(ip_addresses))
```

#### Explicación:

- **with open(import\_file, 'w') as file::** Abrimos el archivo en modo escritura ('w') para sobrescribir el contenido.
- **file.write('\n'.join(ip\_addresses)):** Unimos los elementos de la lista ip\_addresses en una cadena, separándolos por saltos de línea, y escribimos la cadena resultante en el archivo.

## Resumen

El código completo realiza las siguientes tareas:

1. Lee el archivo con la lista de direcciones IP autorizadas.

2. Crea una lista con las direcciones IP a eliminar.
3. Itera sobre la lista de direcciones a eliminar y las remueve de la lista de autorizaciones si se encuentran.
4. Sobrescribe el archivo con la lista de autorizaciones actualizada.

**Recuerda:**

- **Adapta la lista `remove_list`** a tus necesidades específicas.
- **Considera agregar validaciones** para evitar errores, como verificar que el archivo exista o que la lista de eliminación no esté vacía.
- **Puedes mejorar la eficiencia** para listas muy grandes utilizando estructuras de datos más eficientes, como conjuntos.