1. Importar los módulos necesarios: virus_total_apis, key.key_api, os y time.

2.Definir la función analizar_urls que toma el argumento archivo_urls:

2.1. Crear una instancia de PublicApi utilizando la clave de API proporcionada en api_key.

2.2. Inicializar una lista vacía urls_segura.

2.3. Abrir el archivo archivo_urls en modo lectura.

2.4. Iniciar un bucle para iterar sobre cada línea (URL) en el archivo:

2.4.1. Eliminar los caracteres de espacio en blanco al principio y al final de la URL.

2.4.2. Llamar al método get_url_report de la instancia de PublicApi con la URL como argumento y almacenar la respuesta.

2.4.3. Hacer una pausa de 1 segundo (usar time.sleep(1)).

2.4.4. Intentar procesar la respuesta:

2.4.4.1. Verificar si el código de respuesta es 200 (éxito).

2.4.4.2. Verificar si la clave "positives" está presente en los resultados de la respuesta.

2.4.4.3. Si el valor de "positives" es mayor que 0:

2.4.4.3.1. Imprimir un mensaje indicando que la URL es maliciosa.

2.4.4.3.2. Hacer una pausa de 3 segundos (usar time.sleep(3)).

2.4.4.3.3. Imprimir un mensaje indicando que se eliminará el archivo asociado.

2.4.4.3.4. Eliminar el archivo archivo_urls utilizando os.remove.

2.4.4.4. Si el valor de "positives" es igual a 0:

2.4.4.4.1. Imprimir un mensaje indicando que la URL es segura.

2.4.4.4.2. Agregar la URL a la lista urls_seguras.

2.4.4.4.3. Abrir el archivo urls_seguras.txt en modo append.

2.4.4.4.4. Escribir la URL en el archivo urls_seguras.txt con un salto de línea.

2.4.4.4.5. Cerrar el archivo urls_seguras.txt.

2.4.5. Capturar cualquier excepción que ocurra durante el procesamiento de la URL e imprimir un mensaje de error.

2.5. Cerrar el archivo `archivo_urls`.

3. Fin