

PUESTA EN PRODUCCION SEGURA

Unidad 1. Trabajo



23 de NOVIEMbre de 2023

Carlos DÍAZ MONTES

ESPECIALIZACIÓN DE CIBERSEGURIDAD

Índice

[URLLIB 2](#_Toc151638263)

[¿Qué es URLLIB? 2](#_Toc151638264)

[Como instalar urllib 2](#_Toc151638265)

[¿Para qué sirve cada submódulo de URLLIB? 2](#_Toc151638266)

[Ejemplos de usos 3](#_Toc151638267)

# URLLIB

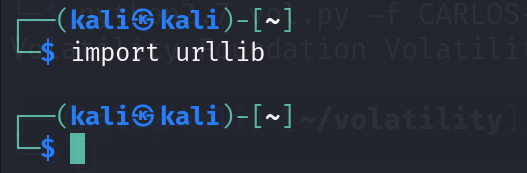
# ¿Qué es URLLIB?

El módulo urllib en Python es un conjunto de módulos que proporciona un conjunto de herramientas para trabajar con URL.

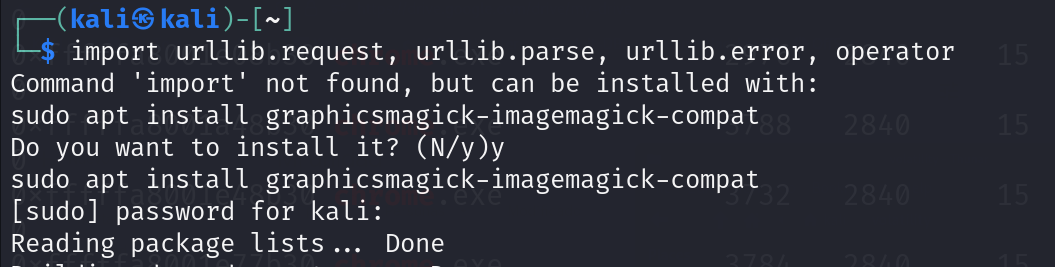
# Como instalar urllib

El módulo urllib es parte de la biblioteca estándar de Python, por lo que generalmente no es necesario instalarlo por separado. Sin embargo, si estás utilizando una versión de Python más antigua (como Python 2.x), es posible que ya esté instalado. En caso de que estés utilizando Python 3, el módulo debería estar disponible de forma predeterminada.

Para comprobar si urllib está disponible, simplemente intenta importarlo en tu script o en la consola de Python:



Ahora lo que debes de importar son los submodulos :



# ¿Para qué sirve cada submódulo de URLLIB?

**- urllib.request:** Permite abrir y leer URLs. Con este módulo, puedes realizar solicitudes HTTP básicas, como GET y POST, y gestionar las respuestas del servidor.

**- urllib.parse:** Proporciona funciones para analizar (parsear) URLs, construir URLs y manipular sus componentes. Es útil para dividir una URL en partes y viceversa.

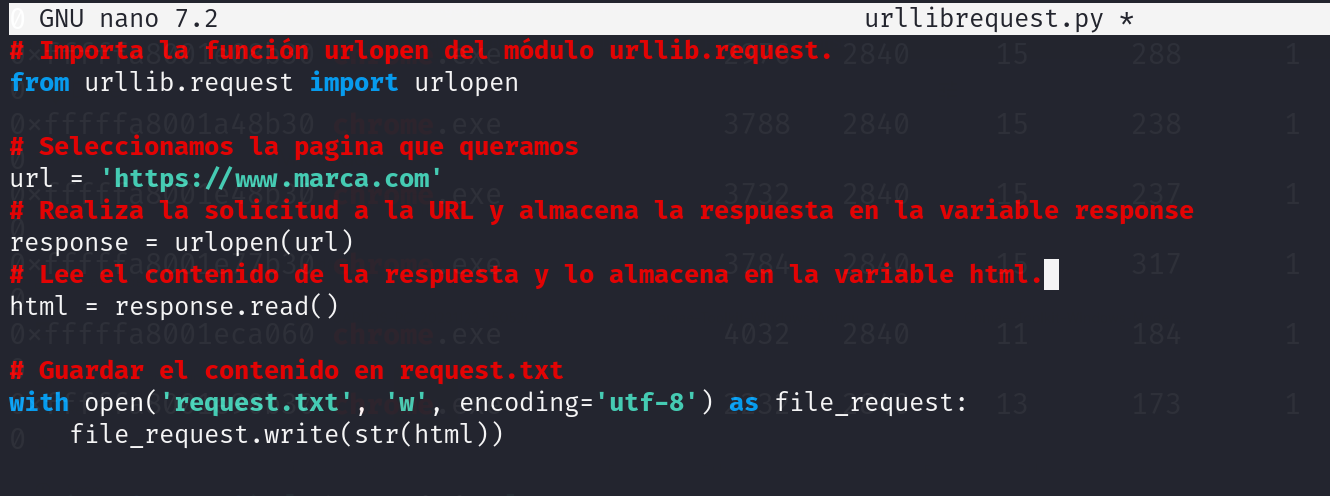
**- urllib.error:** Contiene excepciones específicas para manejar errores relacionados con operaciones de URL, como errores de red, errores HTTP, etc.

**- urllib.robotparser**: Ayuda a analizar el archivo robots.txt de un sitio web para determinar si ciertos agentes web pueden o no acceder a ciertas partes del sitio.

# Ejemplos de usos

**Urllib.request**

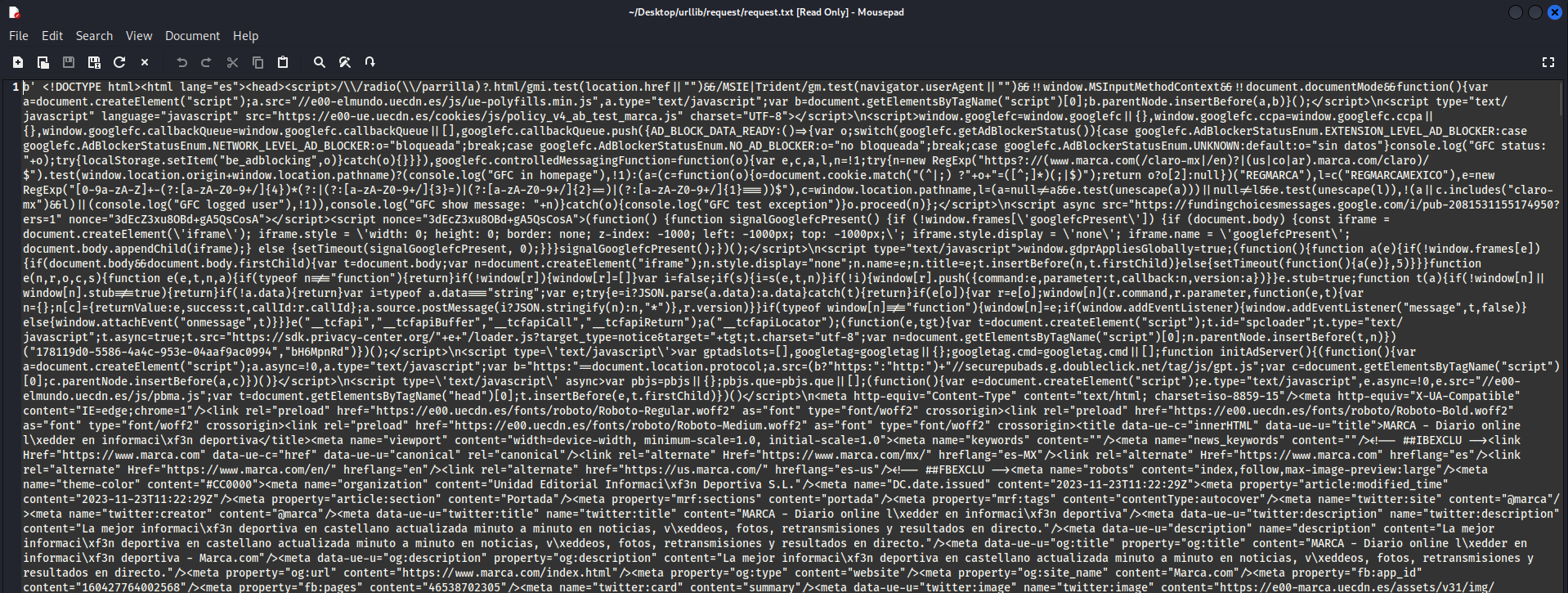
Vamos a realizar una solicitud HTTP básica:



Ahora comprobamos que nos ha devuelto un archivo llamado request txt con el contenido html de la pagina:

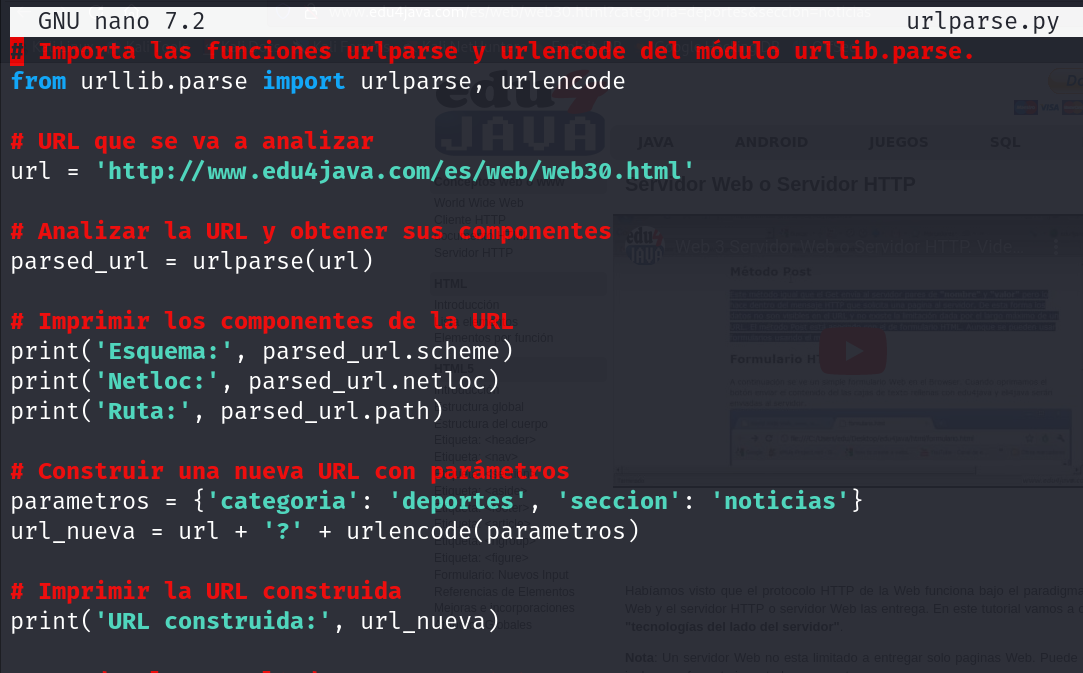


El contenido de la pagina:

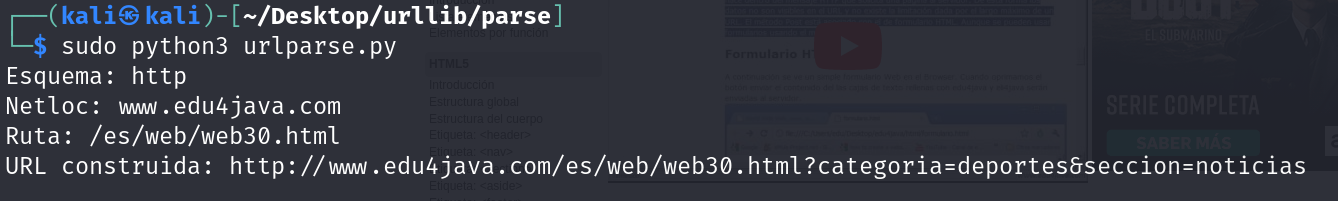


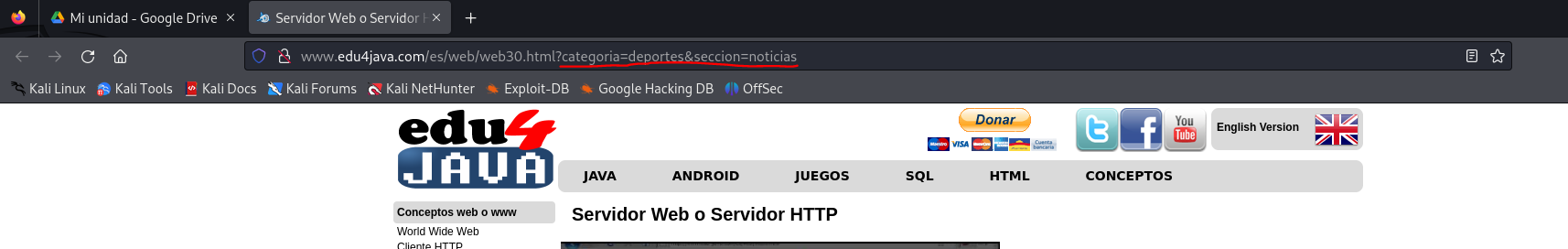
**urllib.parse**

Analizamos y construimos URLs**:**



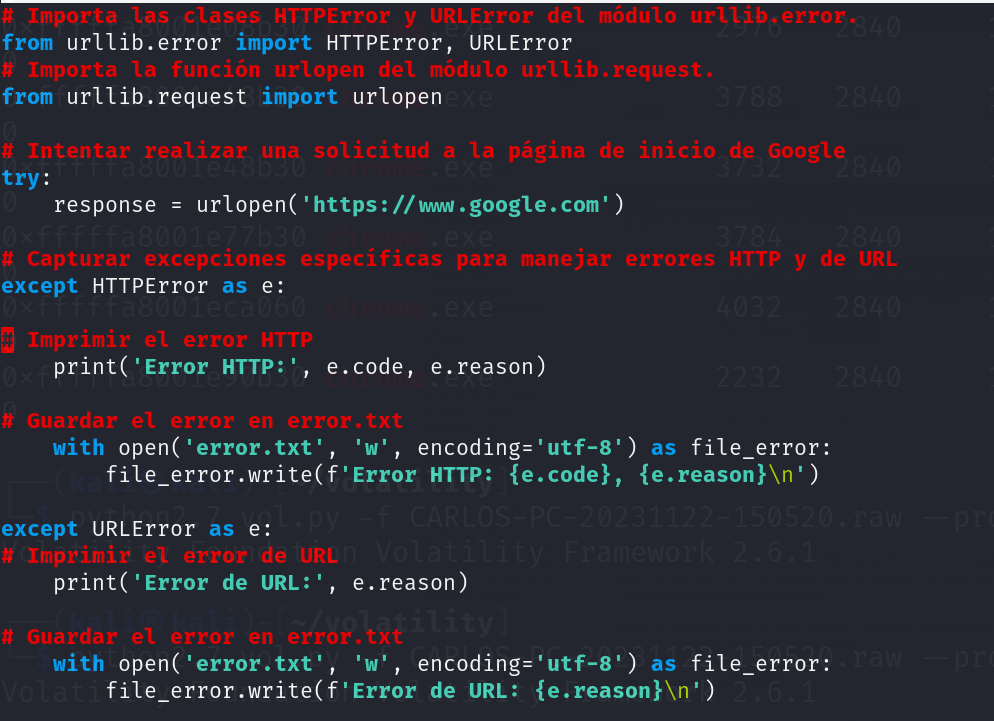
Como resultado:



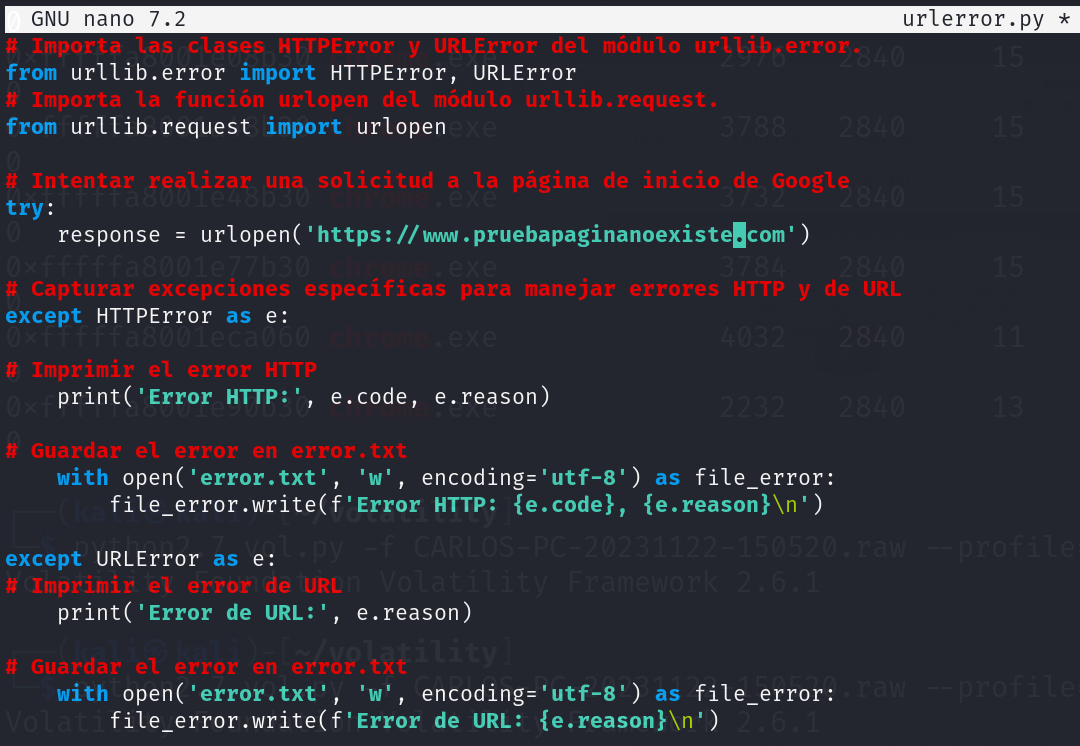


**urllib.error**

Manejamos errores de URL, por ejemplo vamos a buscar un error en Google.es (no vamos a encontrar ninguno claramente):



Ahora vamos a buscar errores en una página inventada por ejemplo:

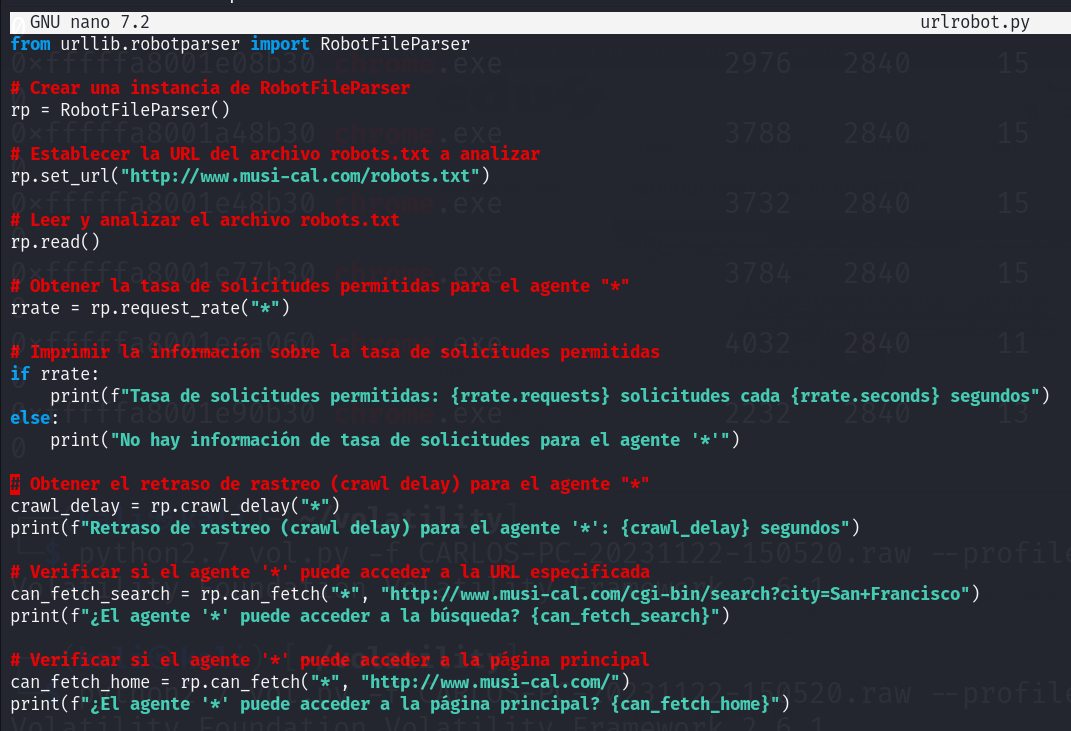


Y ahora vemos el resultado del error:

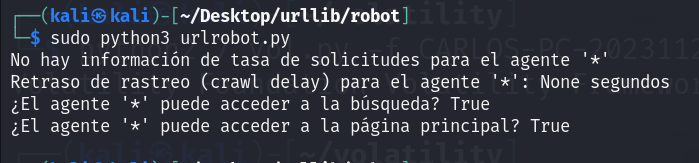


**urllib.robotparser**

El archivo robots.txt es utilizado por los propietarios de sitios web para comunicar a los motores de búsqueda y otros agentes web qué partes de su sitio web pueden o no ser rastreadas. Analizamos el archivo robots.txt:



Aquí podemos ver el resultado:



Si la función can\_fetch(useragent, url) devuelve True, significa que, según las reglas establecidas en el archivo robots.txt, el agente de usuario especificado (por ejemplo, un motor de búsqueda) tiene permiso para rastrear la URL proporcionada. En otras palabras, la página puede ser rastreada por ese agente de usuario.

Si la función devuelve False, significa que el acceso está prohibido y el agente de usuario especificado no tiene permiso para rastrear la URL. En este caso, la página no debería ser rastreada por ese agente según las reglas del archivo robots.txt.