Logo

Description automatically generatedLogo

Description automatically generated with medium confidence

*Practica: 05*

*Nombre de la Practica: Request y Beautiful soap*

Instituto Politécnico Nacional.

Escuela Superior de Cómputo.

Licenciatura en ciencia de datos.

Nombre de la materia: Desarrollo de Aplicaciones para el Análisis de Datos

Grupo: 4AV1

Profesora: Sandra Luz Morales Guitron

Lopez Mendez Emiliano

INDICE

[INTRODUCCION 3](#_Toc182472905)

[DESAROLLO 3](#_Toc182472906)

[Protocolo HTTP y librería REQUESTS 3](#_Toc182472907)

[Web scraping con REQUESTS Y BEAUTIFULSOUP 6](#_Toc182472908)

[CONCLUCIONES 6](#_Toc182472909)

# INTRODUCCION

En esta practica se llevo la instalación de un Editor de programación para poder programar con la versión mas actualizada de Python. Escogiendo la computadora que vamos a usar, el sistema operativa y el Editor que se usaran.

# DESAROLLO

Se llevo acabo el análisis y configuración de varios códigos ya adentro de pypass. Se modificaron para poder imprimir los resultados deseados y también para poder imprimir dichos programas se hizo un archivo aparte llamado args.py con las variables que se piden del main.

## Protocolo HTTP y librería REQUESTS

A continuación se lleva acabo un ejemplo de los entresijos del protocolo HTTP, lo cual es fundamental para sortear los obstáculos que nos pondrán las webs que queramos scrapear en el futuro. Además, veremos cómo acceder a los datos, tanto de la petición del cliente como de la respuesta del servidor, con la librería requests.

A black screen with white text

Description automatically generated

Se importa la librería requests, que se utiliza para hacer solicitudes HTTP.

A screenshot of a computer program

Description automatically generatedDefine la URL de un producto de Amazon y un encabezado User-Agent para imitar una solicitud desde un navegador común, lo cual puede ayudar a evitar bloqueos de Amazon por considerar la solicitud como automatizada.

A screen shot of a computer screen

Description automatically generated

Envía una solicitud GET a la URL de Amazon con los encabezados especificados. La variable res.status\_code guarda el código de estado de la respuesta, y un valor de 200 indica que la solicitud fue exitosa.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

res.reason devuelve una descripción del estado de respuesta (por ejemplo, 'OK'). res.ok es un valor booleano que indica si la solicitud fue exitosa (True si el código de estado es 200).

A screen shot of a computer

Description automatically generated

Abre un archivo llamado codigo\_600.html y guarda el contenido HTML de la página obtenida. Esto permite ver el código HTML del producto en Amazon sin necesidad de una conexión en tiempo real.

A black and grey line

Description automatically generated Aquí podemos ver como ya se guardo el código en nuestra maquina

A screenshot of a computer program

Description automatically generated

El contenido de la pagina html

A computer screen shot of a backpack

Description automatically generated

A screenshot of a computer

Description automatically generated

Podemos ver como ya solo es el contenido html guardado en nuestra computadora y no es la pagina web

A screen shot of a computer

Description automatically generated

Muestra las cookies establecidas por Amazon en la respuesta. Estas cookies pueden ser necesarias para hacer solicitudes posteriores a Amazon.

A screen shot of a computer

Description automatically generated

Muestra los encabezados de la solicitud que se envió, incluidos el User-Agent, la codificación aceptada, y la conexión.

## Web scraping con REQUESTS Y BEAUTIFULSOUP

# CONCLUCIONES

A lo largo de esta práctica se trabajó con diversos conceptos fundamentales de programación, incluyendo estructuras de control, funciones recursivas, generadores, y programación orientada a objetos. Cada ejercicio permitió explorar un aspecto particular, desde manipulación de listas y validación de datos hasta el uso de decoradores y excepciones personalizadas. Además, se implementaron algoritmos clave como el ordenamiento por burbuja, verificación de palíndromos y cálculo de números de Fibonacci, lo que reforzó la comprensión de técnicas de resolución de problemas. Esta práctica fue fundamental para aplicar conocimientos teóricos de forma práctica, mejorando la capacidad de diseñar soluciones eficientes y estructuradas en Python.