



## Práctica 2 – Protocolo HTTP

### 1. Introducción

Un **socket** se puede definir como un extremo de un enlace de comunicación bidireccional entre dos programas que se comunican por la red (se asocia a un número de puerto).

- Se identifica por una dirección IP de la máquina y un número de puerto.
- Existe tanto en TCP como un UDP.

En esta práctica trabajaremos con sockets en Python y con el protocolo HTTP. Para el desarrollo, utilizaremos un notebook de Jupyter, y para revisar el tráfico generado, haremos uso de Wireshark.

La práctica es individual, tiene una duración de 2 semanas y podéis hacer uso de alguna de las herramientas de IA generativa a vuestro alcance.

### 2. Tareas a realizar

A continuación, se describen las tareas que deben realizarse para completar la práctica. Cada una de ellas ha de solucionarse dentro de una única celda de un notebook de Jupyter.

#### Tarea 1. Peticiones HTTP usando el paquete requests

Utilizando el paquete requests de Python, enviar las siguientes peticiones. Para todas ellas, mostrar por pantalla el código de estado de la respuesta HTTP, las cabeceras HTTP recibidas en la respuesta y el cuerpo de entidad de dicha respuesta:

1. Petición GET a <https://httpbin.org/html>
2. Petición GET a <https://httpbin.org/html2>
3. Petición GET a <https://httpbin.org/forms/post>
4. Petición POST a <https://httpbin.org/post>. En esta petición el cuerpo de entidad de la petición debería contener valores para todos los atributos del formulario obtenido en la petición GET del punto previo.

#### Tarea 2. Petición HTTP usando sockets TCP

Utilizando el paquete socket de Python, enviar la petición HTTP “GET /html HTTP/1.1” al puerto 80 de httpbin.org. Incluir la cabecera Host y su valor correspondiente.



### Tarea 3. Petición HTTP segura

De nuevo, con el paquete requests, enviar las siguientes peticiones:

1. Petición GET a <http://httpbin.org/html>
2. Petición GET a <https://httpbin.org/html>

Capturar el tráfico con Wireshark y analizar las diferencias entre ambas peticiones.

### Tarea 4. Analizar el contenido de una respuesta HTTP

Utilizar la librería BeautifulSoup, para analizar el contenido HTML de la web creada en la práctica 1 de la asignatura. Para ello, dentro de las opciones de BeautifulSoup escoger html.parser. En concreto, seguir los siguientes pasos:

1. Emplear el paquete requests para conectarse a la web creada en la práctica 1.
2. Contar el número de veces que aparece la etiqueta HTML paragraph (p) en la respuesta obtenida.
3. Mostrar el contenido de todas las etiquetas h1, h2 y p dentro del contenido HTML obtenido.

Capturar el tráfico con Wireshark al ejecutar esta tarea.

## 3. Evaluación

Esta práctica supondrá hasta **0.35 puntos** de la nota final de la asignatura.

Para evaluar la práctica se realizará una defensa en el laboratorio de prácticas (en el grupo correspondiente a cada alumno) para comprobar su correcto funcionamiento y se realizará una entrega a través del Campus Virtual incluyendo:

- El notebook de Jupyter implementado.
- Una captura de tráfico en formato pcap correspondiente a la ejecución del código de la tarea 4.
- Responder a las preguntas sobre el uso de IA Generativa en la práctica.

La fecha límite de entrega será el **27 de febrero de 2026** a las 23:59. No se admitirán entregas fuera de plazo. Para que la práctica sea evaluada, es necesario realizar tanto la entrega como la defensa.