

**Universidad EAFIT**  
**ST0263: Tópicos Especiales en Telemática, 2022-2**  
**Laboratorio No 1.**

***Fecha de Entrega: 07-agosto-2022***

**Título:** Implementación de un mini-servidor Web (HTTP) minimalista sobre Sockets TCP

**Conceptos Fundamentales:**

Dos o más procesos / aplicaciones en Internet se pueden comunicar de diferentes maneras para implementar un servicio distribuido. Uno de los mecanismos de comunicación más básico y de bajo nivel es a través de Sockets, lo cual plantea una tubería o enlace de comunicación de intercambio de mensajes entre los procesos.

Una de las aplicaciones estándar más populares es la Web, ampliamente conocido y el cual consiste en su versión más simple (web 1.0) en un cliente, browser o navegador (ej: Chrome, Edge, Firefox, etc) que se conecta a un Servidor Web como Apache Web Server para transferir Recursos (páginas web, pdfs, etc) al Cliente.

La web es su concepto más amplio ([https://es.wikipedia.org/wiki/World\\_Wide\\_Web](https://es.wikipedia.org/wiki/World_Wide_Web)) utiliza 3 estándares fundamentales:

1. Un Protocolo estándar de comunicación entre un browser y un servidor web: HTTP
  - a. [https://es.wikipedia.org/wiki/Protocolo\\_de\\_transferencia\\_de\\_hipertexto](https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto)
2. Un formato de datos principal de intercambio: HTML
  - a. <https://es.wikipedia.org/wiki/HTML>
3. Un localizador de recursos web en Internet: URL
  - a. [https://es.wikipedia.org/wiki/Localizador\\_de\\_recursos\\_uniforme](https://es.wikipedia.org/wiki/Localizador_de_recursos_uniforme)

Como un mecanismo para entender y dominar los estándares de la Web (HTTP, HTML y URLs), realizarán un programa con los siguientes requerimientos:

1. Implementar un Mini-servidor web minimalista, que implemente únicamente el método GET (versión 0.9 del protocolo HTTP). Debe recibir peticiones HTTP y procesarlas. El procesamiento de la petición debe permitir el localizar el recurso solicitado en el sistema de archivos local al mini-servidor. Dicha aplicación debe funcionar de la siguiente forma:
  - a. El mini-servidor debe ser capaz de escuchar peticiones a nivel de sockets TCP en cualquier puerto. Dado que implementa la funcionalidad de un servidor web, debe preferiblemente ejecutarse el puerto 80.
  - b. El mini-servidor debe ser capaz de procesar el mensaje que llega por el socket, el cual es un mensaje HTTP Request, procesar dicho mensaje, localizar en el recurso y en caso de que, el recurso exista enviar dicho objeto por el sockets a

través de un HTTP Response. En caso de que el servidor no posea el recurso, debe enviar el mensaje respectivo al cliente vía HTTP response.

2. El mini-servidor debe ser concurrente, es decir, debe permitir conectarse varios browsers al mismo tiempo.
3. El mini-servidor debe decodificar el protocolo HTTP-Request y HTTP-Response en consola, en un modo de resumen de la información. Si hay una opción o parámetro que no es decodificado por el programa, deberá sacar el mensaje 'Opción o Valor Desconocido'. (ver programas como Wireshark para decodificar protocolos - <https://www.wireshark.org/>)
4. El servidor deberá ser desplegado en una máquina virtual en AWS Academy, con IP Elástica, oyendo por el puerto estándar 80. (si bien, el programa del mini-servidor debe ser configurable el puerto).
5. El servidor deberá entregar por defecto archivos del tipo .html o .htm, sin embargo, deberá al menos entregar otros tipos de archivos como .pdf u otros formatos.

Condiciones del proyecto:

1. Ud lo puede implementar en cualquier lenguaje de programación de su preferencia.
2. Hay cientos de implementaciones públicas de estos servidores, si ud reusa algún código, deberá referenciar en el informe final en el github (readme.md) la URL donde lo extrajo y responder por todos los detalles de implementación.

Otras Referencias:

- [Socket de Internet - Wikipedia, la enciclopedia libre](#)
- [Network socket - Wikipedia](#)
- [Socket Programming in Python \(Guide\) – Real Python](#)
- [Unix Socket Tutorial - Tutorialspoint](#)

**Fecha de entrega:**

Hasta el 7 de agosto 2022, hora 23:59. Por Interactiva Virtual

Trabajo individual.