

***Materia:*** *Aplicaciones para Comunicaciones en Red*

***Grupo:*** *3CM17*

***Nombre del Trabajo:***

*Reporte de la Práctica 4. Servidor WEB*

***Alumnos:***

*Peña Atanasio Alberto*

*Martínez Coronel Bryan Yosafat*

***Maestro:***  *Moreno Cervantes Axel*

# Introducción

## Servidor Web

Un servidor web (server) es un ordenador de gran potencia que se encarga de “prestar el servicio” de transmitir la información pedida por sus clientes (otros ordenadores, dispositivos móviles, impresoras, personas, etc.)

Los servidores web (web server) son un componente de los servidores que tienen como principal función almacenar, en web hosting, todos los archivos propios de una página web (imágenes, textos, videos, etc.) y transmitirlos a los usuarios a través de los navegadores mediante el protocolo HTTP (Hipertext Transfer Protocol).

## Métodos HTTP más comunes

HTTP define un conjunto de métodos de petición para indicar la acción que se desea realizar para un recurso determinado. Aunque también pueden ser sustantivos, estos métodos de solicitud a veces son llamados verbos HTTP. Cada uno de ellos implementan una semántica diferente, pero algunas características son similares compartidas por un grupo de ellos: ej. Un método de solicitud puede ser seguro, idempotente (en-US) o almacenable en caché .

**GET**El método GET solicita una representación de un recurso especifico. Las peticiones que usen el método GET sólo deben recuperar datos.

**HEAD**El método HEAD pide una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta.

**POST**El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

**PUT**El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.

**DELETE**El método DELETE borra un recurso en específico.

# Desarrollo

El proyecto cuenta con 3 archivos principales, como se muestran en la Figura 1.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 1

## Archivo ServidorWeb.java

Este archivo lee un puerto, donde correrá la aplicación. Lee un pool de conexiones y acepta la conexión de clientes. Se aprecia de mejor manera en la Figura 2.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 2

## Archivo Mime.java

Este archivo indica el tipo de contenido de la respuesta del servidor, todas las extensiones de la respuesta se ven en la Figura 3.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 3

## Archivo Manejador.java

Este archivo es el que da toda la funcionalidad al servidor, por que es el que controla las peticiones que se hagan desde un navegador y más específicamente, de la aplicación Web llamada Postman, ya que esta última hace peticiones HTTP del tipo HEAD, DELETE y PUT, dichas peticiones no las puede hacer un navegador común

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 4

La clase tiene un atributo socket, donde se hará la conexión, tiene un flujo de salida, un MIME que es el tipo de respuesta y, por último, contiene un flujo de entrada, el constructor sólo crea los atributos de la clase. Se ve en la Figura 4.

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Figura 5

El método eliminarRecurso de la Figura 5, lo que hace es buscar un recurso en una cierta ruta especificada desde la URL y luego verifica si existe dentro del servidor, si no existe manda un código de error 404, pero si existe, lo elimina del servidor.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente  
Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente  
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 6

En la figura 6 se ve el uso del método enviaRecurso el cual es de ayuda para los métodos HTTP Get, Post y Head, lo que se hace es buscar un archivo y luego mandarlo a través de un flujo de salida, también retorna un encabezado con los códigos de error 404 o 202 si la acción se ejecutó correctamente.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 7

El método de la Figura 7 obtiene el nombre del recurso solicitado a través de la URI

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 8

El método obtenerParametros recupera el cabecero de respuesta para que se puedan ejecutar ciertos métodos, se ve de mejor forma en la Figura 8.

Texto

Descripción generada automáticamente   
Texto

Descripción generada automáticamente  
Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente  
Texto

Descripción generada automáticamente  
Interfaz de usuario gráfica, Texto

Descripción generada automáticamente  
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 9

El método run() de la Figura 9, es el Hilo de ejecución que está esperando solicitudes de un cliente, en pocas palabras, se puede describir como un menú que analiza qué método HTTP ejecutar, luego manda a llamar a los métodos correspondientes para que se pueda implementar el método elegido, todo esto a través de la ayuda de un ENCABEZADO, una URL y un MIME.

# Conclusiones

Esta práctica fue un reto grande, debido a que se tuvo que leer y comprender mucho para poder implementar los métodos HTTP correctamente, pero de igual forma nos llevamos un gran aprendizaje ya que comprendimos de una manera más precisa cómo es que funcionan los métodos POST y GET, ambos muy importantes para el uso de las aplicaciones en red.

También conocimos cómo es que trabajan los MIMES, los encabezados de respuesta y los distintos códigos de error.