Taller Clientes y Servicios

Arquitecturas Empresariales (AREP)

Laura Alejandra Izquierdo Castro[1]

[1] *Escuela Colombiana de Ingeniería Julio Garavito, Bogotá, Colombia*

Fecha: 07/09/20121

Resumen— En el presente artículo se describe la arquitectura del prototipo publicado en Github a través de los conceptos de cliente y servicio.

Palabras clave— Arquitectura de aplicaciones distribuidas; Cliente; Servicios.

Abstract— This article describes the architecture of the prototype published on Github through the concepts of client and service.

Keywords— Distributed Application Architecture; Client; Services.

INTRODUCCIÓN

En el presente artículo describimos la arquitectura de un servidor Web comprendida en desarrollo con diseño, pruebas y conclusiones.

Con **Servidor web** se puede hacer referencia a hardware o software, o a ambos trabajando juntos.

- En cuanto a **hardware**, un servidor web es una computadora que almacena los archivos que componen un sitio web (ej. documentos HTML , imágenes, hojas de estilos CSS y archivo JavaScript) y los entrega al dispositivo del usuario final. Está conectado a internet y es accesible a través de un nombre de dominio como mozilla.org.

- En cuanto a **software**, un servidor web tiene muchas partes encargadas del control sobre cómo tienen acceso los usuarios a los archivos, por lo menos un servidor HTTP. Un servidor HTTP es una pieza de software que comprende URLs (direcciones web) y HTTP (el protocolo que tu navegador usa para ver las páginas web).

Al nivel más básico, siempre que un navegador necesite un archivo almacenado en un servidor web, el navegador hará una solicitud al servidor mediante la vía HTTP. Cuando la petición llega al servidor web correcto (hardware), el servidor HTTP (software) envía el archivo antes solicitado, tambien a través de HTTP.Contributors

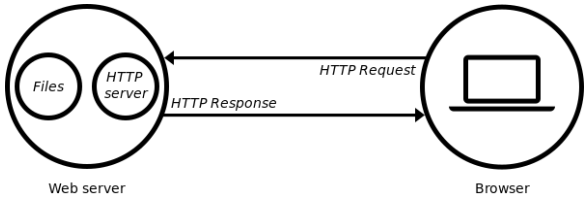

Fig. 1: Diagrama de concepto Cliente-Servicio

DESARROLLO

Para la construcción del servidor web se realizó el diseño descrito en la **Fig. 2**; el cual soporta múltiples solicitudes seguidas (no concurrentes). Así como retorna todos los archivos solicitados, incluyendo páginas html e imágenes. También se construye un sitio web con javascript para probar el servidor. Se usa solo Java y las librerías para manejo de la red.

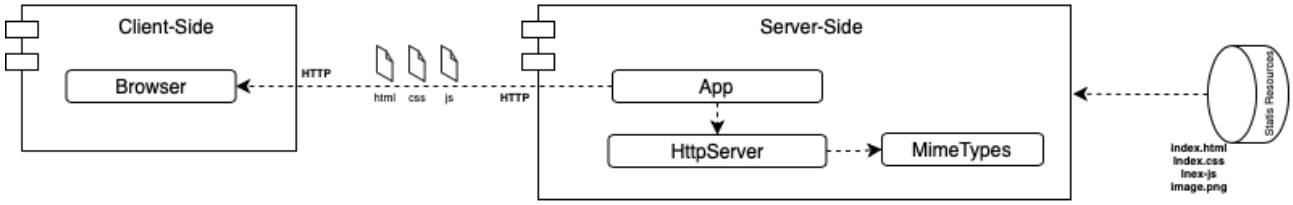

Fig. 2: Diseño de Arquitectura del servidor Web

RESULTADOS

Se tiene una arquitectura simple, con la que se puede procesar peticiones GET sobre archivos de tipo *html*, *css*, *js* y *png* almacenados en los recursos estáticos del servidor de la forma presentada en la **Fig-3**. Contributors

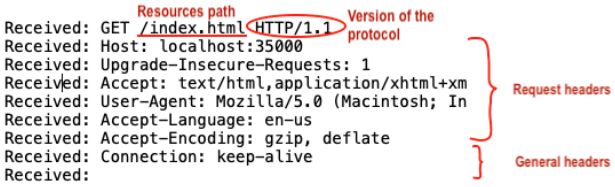

Fig. 3: Estructura mensajes HTTP

Recursos de Mimetype text

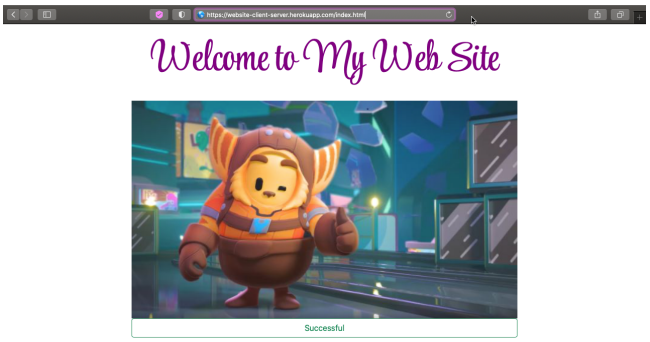

Fig. 4: Solicitud a recurso estático /index.html

En el la solicitud del recurso de la **Fig-4**, las solicitudes del cliente correspondientemente serán: GET /index.html, GET /index.css y finalmente GET /index.js, como se muestra en la **Fig-5**. Correspondiendo a Mimetype text/js

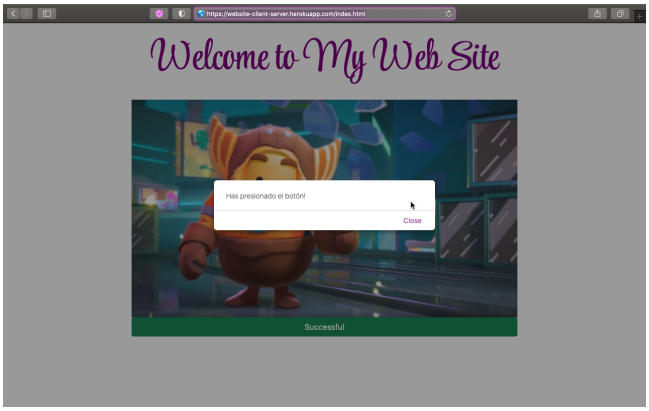

Fig. 5: Solicitud a recurso estático /index.js

Recursos de Mimetype image

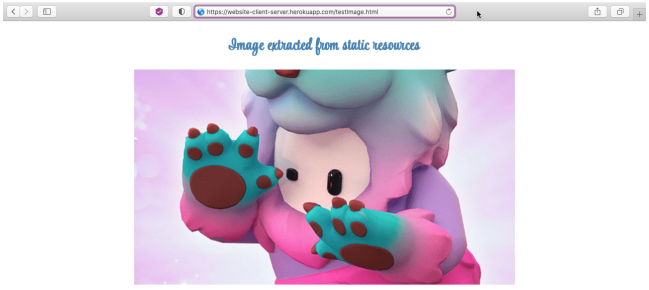

Fig. 6: Solicitud a recurso estático /testImage.html

En la solicitud del recurso de la **Fig-6**, la solicitud del cliente se hace sobre un archivo html que tiene incrustada una imagen que se obtiene con: GET /testImage.html y GET /fall.jpg

CONCLUSIONES

La construcción de un servidor web puede ser posible sin el uso de frameworks, simplemente utilizando las librerias *.net* de java.

Se llegó también a un entendimiento para el manejo de peticiones GET, puesto que se puede identificar el recurso obteniendo la estructura completa de un *request* (headers, body) enviado desde el cliente y acceder a él en disco, para retornarlo.

REFERENCIAS

[1] Contributors, M. (2021a). "Mensajes http". Tomado de `https://developer.mozilla.org/es/docs/Web/HTTP/Messages` (5/09/2021).

[2] Contributors, M. (2021b). "Que es un servidor web?" Tomado de `https://developer.mozilla.org/es/docs/Learn/Common_questions/What_is_a_web_server` (5/09/2021).