

Servidor Web en Java (Septiembre 2019)

Santiago Rocha Durán

I. INTRODUCCIÓN

Un servidor Web es un programa que utiliza HTTP (Hypertext Transfer Protocol) para servir los archivos que forman páginas Web a los usuarios, en respuesta a sus solicitudes, que son reenviados por los clientes HTTP. Un servidor web opera en un ordenador aguardando las solicitudes de parte del navegador web de un cliente, brindando los datos solicitados para componer una página web o, en su defecto, un mensaje de error. Los servidores web pueden ser de dos clases:

- **Estáticos:** los archivos se envían tal y como están almacenados.
- **Dinámicos:** Realizan operaciones con los datos antes de enviar respuesta al cliente

Típicamente se usan servidores web como Apache para el desarrollo de aplicaciones más grande pero en esta ocasión se va a describir el desarrollo de un servidor web en java.

II. ARQUITECTURA DE SOFTWARE

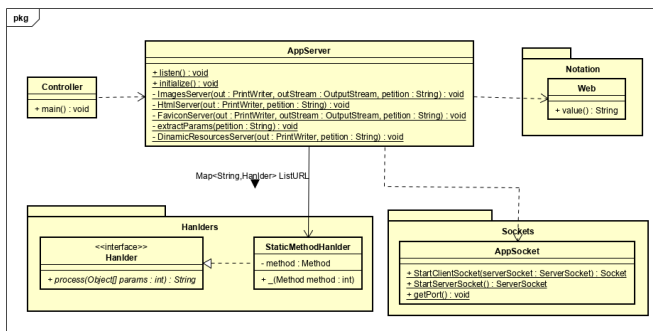


Fig. 1. Diagrama que describe la arquitectura del proyecto.

A continuación se detalla la arquitectura del servidor web, esta arquitectura busca el bajo acoplamiento de sus componentes con el objetivo de continuar agregando características y funcionalidades a este.

- **Handler:** El objetivo de este paquete es tener una interfaz que permita manejar e invocar los métodos de los POJOS que el usuario cargue sin necesidad de conocer concretamente el tipo o sus parámetros
- **Notation:** El objetivo de este paquete es crear anotaciones sobre el código que le permitan al usuario poner el nombre por el cual se va a buscar los recursos dinámicos que desea cargar
- **Sockets:** El objetivo de este paquete es iniciar los sockets tanto de cliente como servidor para comenzar a escuchar las peticiones del browser o de cualquier otro cliente por un puerto definido

El usuario del servidor web debe cargar sus archivos en la carpeta app con el fin de que sean encontrados y ejecutados por el servidor

Finalmente el flujo del programa sería el siguiente:

- 1) El cliente (browser) envía una petición que es recibida por los sockets
- 2) AppServer revisa que la petición este bien formada y selecciona el recurso solicitado
- 3) En caso de ser un recurso estático lo devuelve con el encabezado apropiado de los contrario delega el trabajo al Handler para que maneje los parámetros y la respuesta
- 4) El recurso es enviado como respuesta al cliente y mostrado en el browser

III. ARQUITECTURA DE DESPLIEGUE

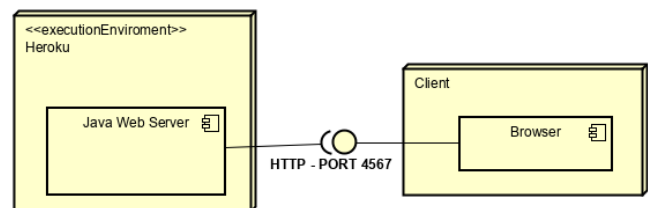
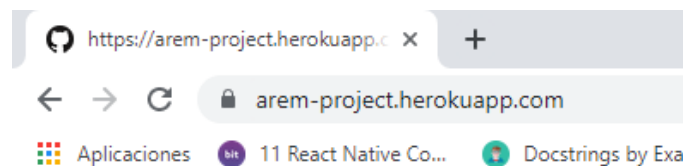


Fig. 2. Diagrama que describe como se despliega el proyecto.

IV. PRUEBAS FUNCIONALES

A. Recursos Estáticos



Pagina de inicio

/app/recursoDinamico

/recursoEstatico

Fig. 3. Índice de la pagina.



Fig. 4. Recurso .html que viene con una imagen .png

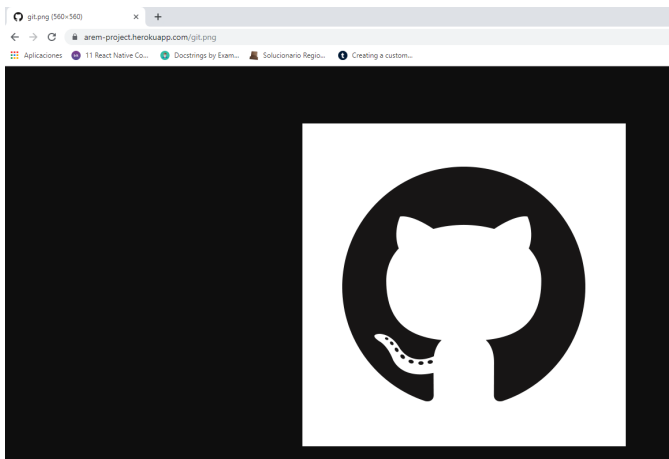


Fig. 5. recurso .png

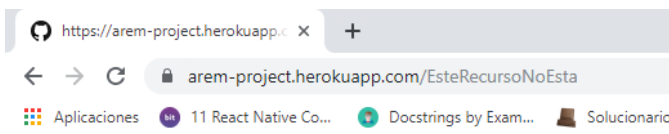


Fig. 6. Error mostrado cuando no se encuentra un recurso

B. Recursos Dinamico

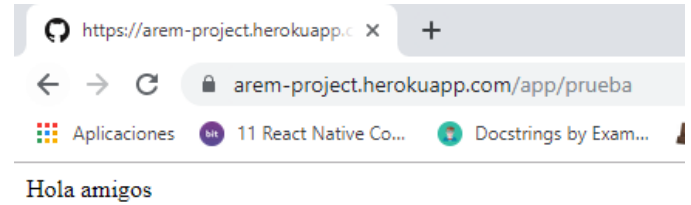


Fig. 7. Recurso dinámico que devuelve una cadena.

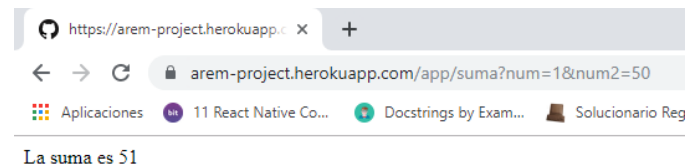
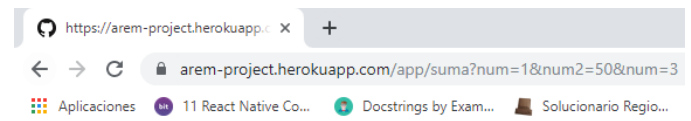


Fig. 8. Recurso dinámico que suma los dos números de los parámetros.



Error al cargar el recurso

Fig. 9. Error en el recurso dinámico.

V. CONCLUSIÓN

El uso de POJOS y anotaciones permiten estructurar mejor el uso y administración de un servidor web a nivel de código. Adicionalmente le brinda al usuario una guía para manejar los recursos que quiere cargar al servidor y como dicho servidor esta esperando que estos sean cargados para que el flujo del software sea correcto y la salida sea la esperada por el usuario

REFERENCES

- [1] María Estela Raffino. (2019). Servidor Web. 06 de septiembre del 2019, de Conceptos Sitio web: <https://concepto.de/servidor-web/>
- [2] osgroup. Que es un servidor web. 06 de septiembre del 2019, de osgroup Sitio web: <https://www.osgroup.co/que-es-un-servidor-web/>