

Escuela Colombiana de Ingenieria Julio Garavito

TALLER DE ARQUITECTURAS DE SERVIDORES DE APLICACIONES, META PROTOCOLOS DE OBJETOS, PATRÓN IOC, REFLEXIÓN

David Andres Herrera
Profesor: Luis Daniel Benavides Navarro,
Ph.D.

Ingenieria de Sistemas
Bogotá
Febrero del 2021

List of Contents

1	Introducción	II
2	Marco teórico	II
3	Arquitectura	II
4	Conclusiones	III
5	Referencias	III

1 Introducción

Este proyecto trata de una implementación de un mini spring usando "POJOS", lo que buscamos en el proyecto es poder hacer que cuando el usuario consulte a nuestro servidor, nuestra aplicación sea capaz de redireccionar a lo que quiere el servidor dependiendo de las anotaciones que tenga, así poder consultar imágenes y páginas html. También se hace un despliegue en heroku y una integración continua en -circleCI.

En este documento lo que se quiere lograr es que el lector se de cuenta de la importancia y la utilidad que tiene Spring para los servidores y la facilidad que le puede dar al desarrollador al crear servicios web.

2 Marco teórico

- **Servidor Web:** Un servidor web es una pieza de software para la comunicación y la pieza fundamental para establecer una comunicación entre el servidor donde están alojados los datos y el computador del cliente, esto permite una conexión de tipo direccional o bidireccional y síncrona o asíncrona. Este servicio ayuda a que lo que está en el servidor de datos se pueda mostrar y generar de manera correcta para que sea más fácil la lectura en el computador del cliente.
- **Spring:** es una tecnología dentro de Spring que ayuda al desarrollador a seleccionar los jars con maven y a desplegar el servidor, ya que al momento de crear la aplicación con maven o gradle y descargar las dependencias necesarias nos ayuda con estas dos cosas. Además una de las facilidades que brinda springboot es que podemos compilar nuestra aplicación web como un archivo jar, eso nos permite ejecutar la aplicación de una manera más fácil.
- **POJO:** Un POJO es el nombre que se utiliza para enfatizar en el uso de clases que no dependen de un framework.

3 Arquitectura

La arquitectura de esta solución está compuesta por tres directorios demo, httpserver y miniSpring.

- El directorio demo cuenta con la clase que se encarga de las anotaciones y devolver la consulta que tenga cada anotación
- El directorio httpserver cuenta con dos clases que se encargan de crear el servidor http que se va a usar para el miniSpringboot
- El directorio miniSpringBoot cuenta con las clases necesarias que reciben la solicitud del servidor http y llama a la clase de las anotaciones para consultar que es lo que tiene la anotación, si la anotación no se encuentra esta clase manda un error diciendo que la anotación que se busca no está.

4 Conclusiones

La practica del laboratorio ayuda a obtener un conocimiento basico de como utilizar Heroku y como desplegar la aplicacion web en heroku de manera local y remota, ademas de esto se aprende a hacer la integracion continua con CircleCi

Al consttruir nuestro miniSpring nos damos cuenta de como es el funcionamiento basico de este framWork y de la ayuda que este frameWork le da al desarroyador para facilitar la creacion de la aplicacion web.

5 Referencias

- Caules, C., 2021. ¿Qué es Spring Boot?. [online] Arquitectura Java. Available at: <<https://www.arquitecturajava.es-spring-boot/>> [Accessed 26 February 2021].
- Mirandas. 2021. Spring vs Spring Boot: Cual es la diferencia?. [online] Available at: <<https://www.mirandas.work/es/posts/spring-vs-spring-boot-cual-es-la-diferencia/>> [Accessed 26 February 2021].