

Taller de Arquitecturas de Servidores de Aplicaciones, Meta Protocolos de Objetos, Patrón IOC y Reflexión

Juan Sebastián Muñoz Dorado

13 de Septiembre del 2021

**Profesor:
Luis Daniel Benavides Navarro**

Arquitecturas Empresariales



Tabla de Contenido

1	Prerrequisitos	2
2	Introducción	2
3	Diseño	3
3.1	Diagrama de Clases	3
3.2	Diagrama de Componentes	4
3.3	Diagrama de Despliegue	5
4	Conclusiones	6
5	Referencias	7

1 Prerrequisitos

Se utilizará Maven como una herramienta para la construcción y gestión del proyecto, el código fue desarrollado con lenguaje de programación Java; por lo tanto es necesario para la ejecución tener las dos herramientas en las versiones correctas.

- Java versión 8 o superior
- Maven versión 3.5 o superior

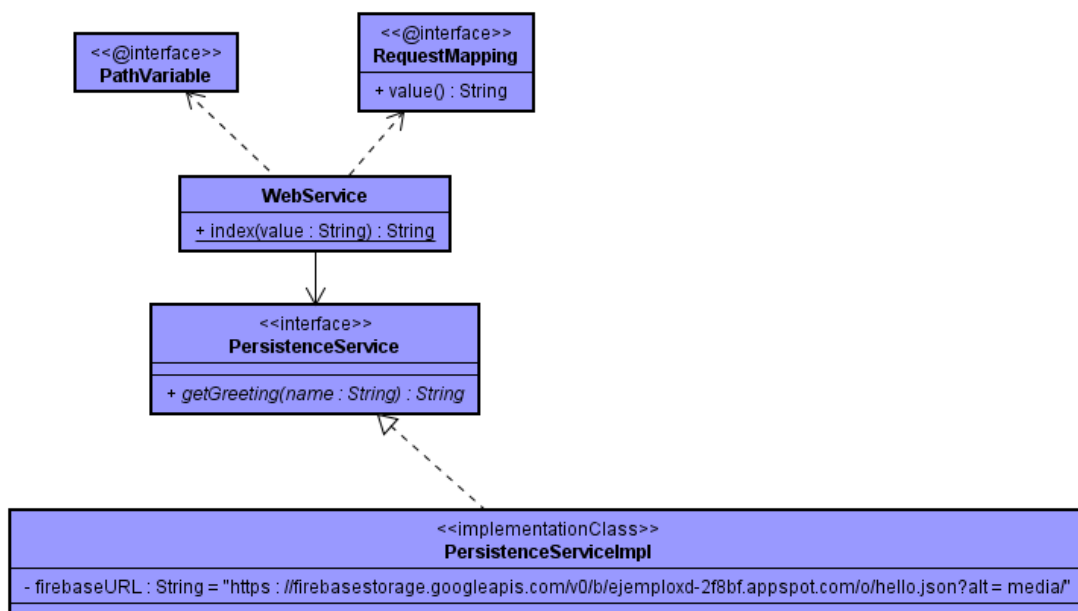
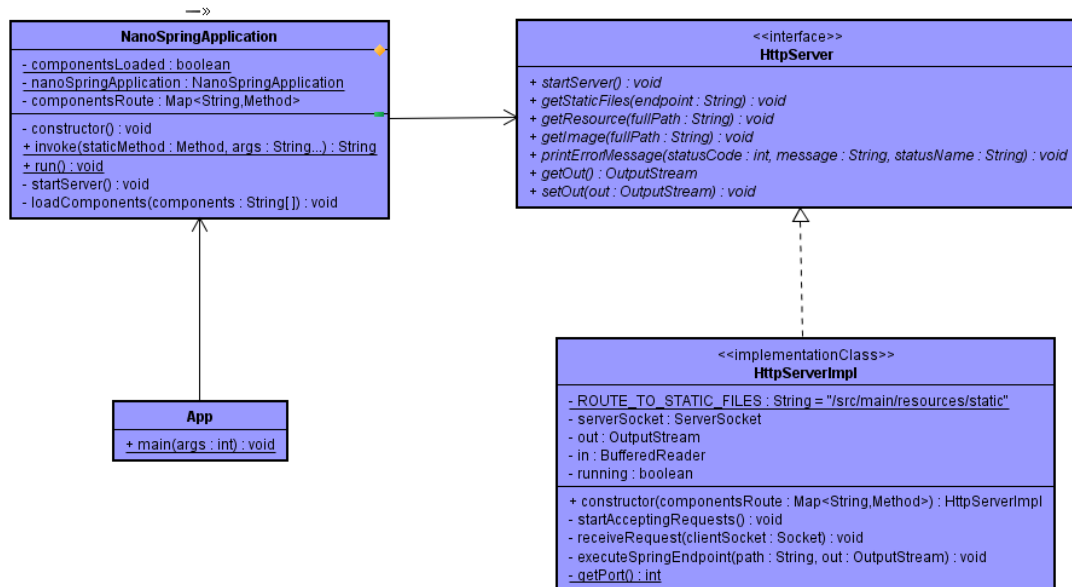
2 Introducción

En el Taller de Arquitecturas de Servidores de Aplicaciones, Meta protocolos de objetos, Patrón IoC, Reflexión se explorarán los conceptos de esquemas de nombres y de clientes y servicios. El laboratorio explorará la arquitectura de las aplicaciones distribuidas en internet, para así tener una idea más clara del funcionamiento detallado de una aplicación web capaz que tiene la capacidad de recibir múltiples solicitudes no concurrentes, creando un framework IoC para la construcción de aplicaciones web a partir de POJOS permitiendo publicar servicios web get y post para poder acceder a recursos estáticos como páginas web, javascripts, imágenes CSSs, entre otras cosas; el despliegue se hizo usando un servidor web llamado Heroku para poder acceder a ella de manera totalmente remota. Para verificar el funcionamiento de cada uno de los requisitos, se realizó unas simulaciones con el servicio ya arriba en Heroku, en la cual el usuario tiene una interfaz de usuario en la que tiene dos recursos, un saludo y la obtención de una imagen.

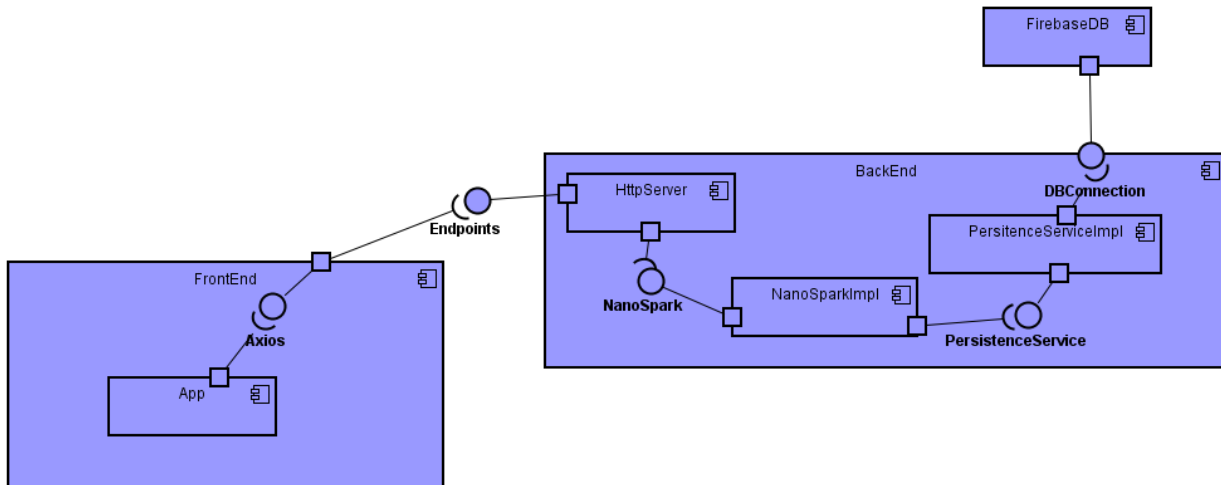
3 Diseño

3.1 Diagrama de Clases

A continuación se presentara los diagramas de clases junto a un diagrama donde se presenta el servicio que se implemento para la solución del laboratorio



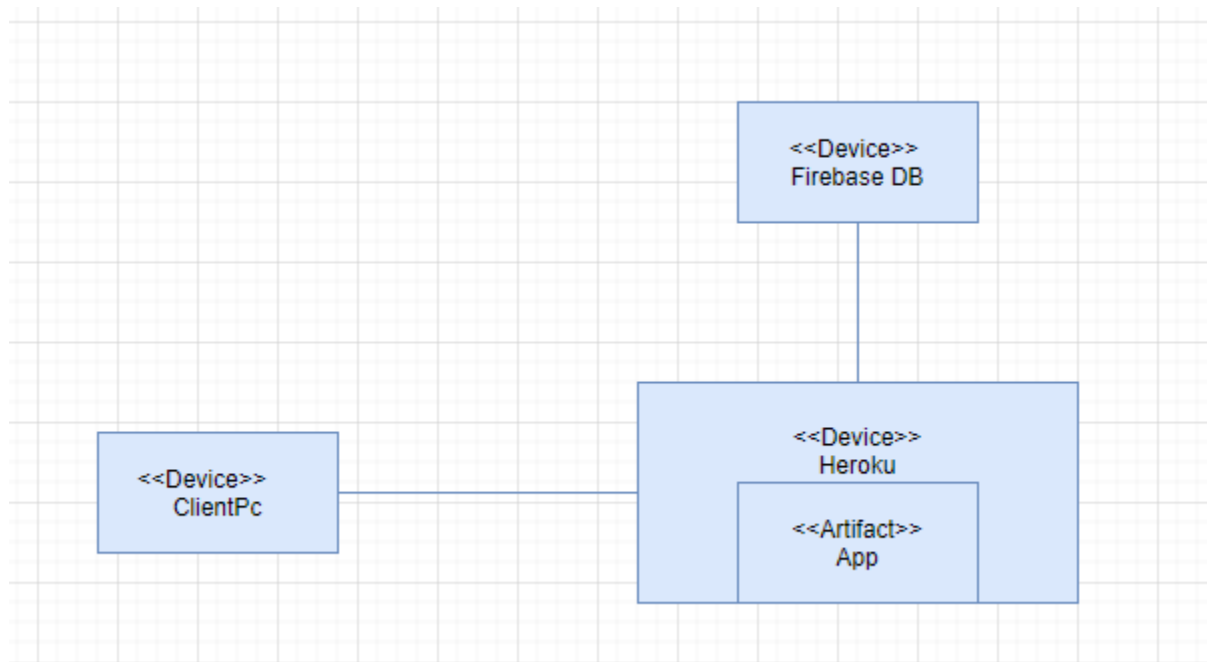
3.2 Diagrama de Componentes



La aplicación tiene tres componentes principales, FrontEnd, BackEnd y FirebaseDB.

El componente que tiene más funcionalidades del FrontEnd es App, el componente lee la información que registra el usuario y por medio de la función Axios accede a HttpServer para usar el endpoint.

3.3 Diagrama de Despliegue



4 Conclusiones

- Es importante el uso de las interfaces para lograr una extensibilidad del proyecto sin tener que hacer cambios de archivos
- El despliegue en Heroku permitió poder ejecutar la aplicación solicitada, probando el funcionamiento de la aplicación web en nube y compilando el código en cualquier ordenador, utilizando el protocolo HTTP para la comunicación cliente-servidor para asegurar una total ejecución del código de manera remota.
- Por otro lado, se implementó la integración continua con CircleCI en todo el código fuente para llevar un control de calidad del código, para asegurar que el código esté funcionando totalmente sin ningún problema tanto de compilación como de los resultados retornados después de realizar las respectivas operaciones, y así poder desplegar la aplicación sin presentar ningún tipo de errores.

5 Referencias

- [1] Ing. Emiliano Marini. *El Modelo Cliente/Servidor*. URL: <https://www.linuxito.com/docs/el-modelo-cliente-servidor.pdf>.
- [2] Eugen Paraschiv. Spring @RequestMapping. url: <https://www.baeldung.com/spring-requestmapping>
- [3] baeldung. Spring @PathVariable. url: <https://www.baeldung.com/spring-pathvariable>.