

# Taller de Modularización con Virtualización e Introducción a Docker y a AWS

Germán Andrés Ospina Quintero

12 de Marzo del 2021

## Contents

<b>1</b>	<b>Glosario</b>	<b>2</b>
<b>2</b>	<b>Resumen</b>	<b>5</b>
<b>3</b>	<b>Introducción</b>	<b>6</b>
<b>4</b>	<b>Estructura</b>	<b>7</b>
<b>5</b>	<b>Diseño</b>	<b>9</b>
<b>6</b>	<b>Arquitectura</b>	<b>10</b>
<b>7</b>	<b>Pruebas</b>	<b>11</b>

# 1 Glosario

Java: Es un tipo de lenguaje de programación y una plataforma informática, creada y comercializada por Sun Microsystems en el año 1995. Se constituye como un lenguaje orientado a objetos, su intención es permitir que los desarrolladores de aplicaciones escriban el programa una sola vez y lo ejecuten en cualquier dispositivo. (Content, 2019)

Git: Es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

(Wikipedia, 2021) Maven: Es una herramienta de software para la gestión y construcción de proyectos Java creada por Jason van Zyl, de Sonatype, en 2002. Es similar en funcionalidad a Apache Ant, pero tiene un modelo de configuración de construcción más simple, basado en un formato XML. (Wikipedia, 2020)

Spark: Es un conjunto de librerías para el desarrollo de aplicaciones web en Java inspirado en el framework Sinatra para Ruby. Spark incluye por defecto el servidor de aplicaciones Jetty de manera que las aplicaciones web creadas con Spark pueden lanzarse como cualquier programa Java. (Wikipedia, 2020) AWS: Amazon Web Services es una colección de servicios de computación en la nube pública que en conjunto forman una plataforma de computación en la nube, ofrecidas a través de Internet por Amazon.com. (Wikipedia, 2021)

Docker: Es un proyecto de código abierto que autom-

atiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. (Wikipedia, 2021)

HTML: siglas en inglés de HyperText Markup Language, hace referencia al lenguaje de marcado para la elaboración de páginas web. (Wikipedia, 2021)

JavaScript: Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. (Wikipedia, 2021)

JQuery: Es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. (Wikipedia, 2021)

JSON: Es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje. (Wikipedia, 2021)

Balanceador de cargas: Es un dispositivo de hardware o software que se pone al frente de un conjunto de servidores que atienden una aplicación y, tal como su nombre lo indica, asigna o balancea las solicitudes que llegan de los clientes a los servidores usando algún algoritmo. (Wikipedia, 2021)

Monolito: Hace referencia al software que combina en

una misma plataforma la capa IU (User Interface), la capa de Bussines (capa de negocio) y la capa de datos (BBDD, JSON...). (JAVA DESDE 0, 2019)

Microservicios: Es una aproximación para el desarrollo de software que consiste en construir una aplicación como un conjunto de pequeños servicios, los cuales se ejecutan en su propio proceso y se comunican con mecanismos ligeros. (Wikipedia, 2021)

## 2 Resumen

Para este taller se realizó la implementación de una aplicación web con un cliente y con un servicio REST, el método HTTP POST. Esta aplicación cuenta con un campo de texto para escribir un mensaje, un botón para realizar la acción de guardado y una tabla que muestra los diez últimos mensajes almacenados. El cliente web se implementó con el uso de HTML, JavaScript y JQuery para realizar la petición POST al servidor con los datos del mensaje (descripción y fecha) y para recibir la confirmación de que los datos fueron persistidos correctamente por parte del Backend. El servidor cuenta con un servicio REST POST, desarrollado con Maven y Spark, que recibe los datos del mensaje en un formato JSON y realiza, con un servidor interno, otra petición POST a uno de los tres servidores (LogService) encargados de almacenar los datos del mensaje en una base de datos MongoDB previamente implementada, según le indique el algoritmo de balanceo de cargas. Con el ánimo de no implementar un monolito que hiciera difícil el mantenimiento posterior de la aplicación, al tener que manejar la aplicación como un todo, para cada uno de los tres servicios presentados (la aplicación web, el LogService y la base de datos) se almacena su lógica, desarrollada en Java, en repositorios GitHub independientes y se modularizó con la ayuda de imágenes y contenedores Docker. Además, se hizo una virtualización de estos servicios con instancias EC2 Linux de AWS.

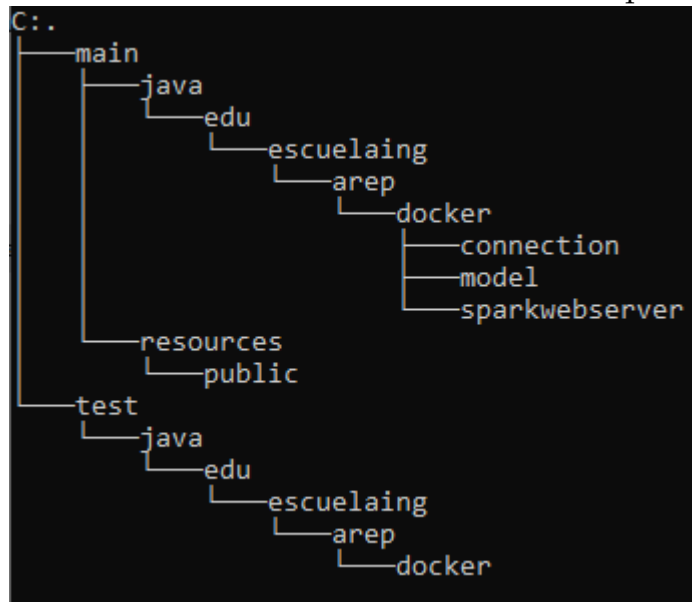
### 3 Introducción

Como se mencionó brevemente en el resumen, una aplicación monolítica puede tener varias complicaciones al momento de realizar sobre esta un análisis de funcionamiento a través de pruebas como comunmente se hace. La aplicación que se presenta en este artículo se hizo en base a microservicios. Estos contenedores de funcionalidades, presentan varios beneficios sobre una arquitectura monolítica, como por ejemplo: se puede reducir el acoplamiento al separar cada una de la funcionalidades técnicas en contenedores independientes, permite evolucionar la aplicación más fácilmente, el producto final es, en comparación, mucho más pequeño y, como ya se había dicho, es sencillo identificar los problemas de rendimiento, es decir, es más fácil realizar una monitorización técnica.

En las siguiente secciones del documento, se hablará acerca de la estructura del taller, es decir, cómo están distribuidos los directorios de este; así mismo, se hablara de el diseño de la aplicación con el uso del diagrama de clases, de la arquitectura necesaria para hacer el despliegue de este con un diagrama de despliegue y de las pruebas que se realizaron para validar su correcto funcionamiento.

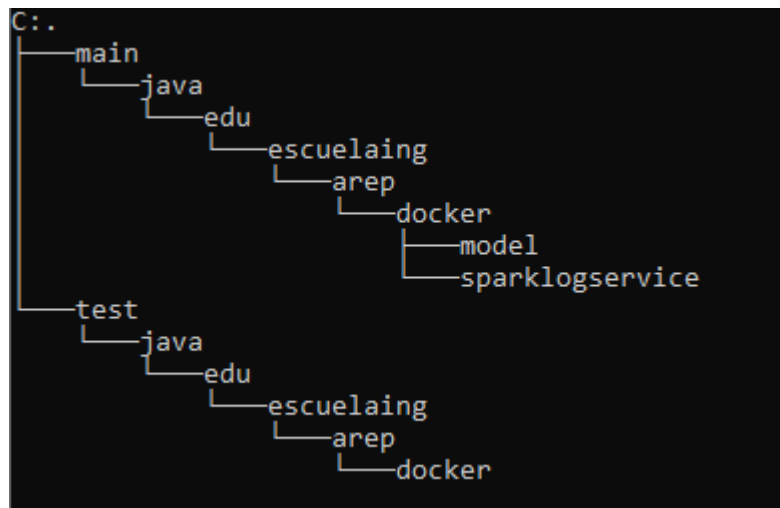
## 4 Estructura

En esta sección, se presentan dos estructuras de archivos, puesto que los servicios de la aplicación web y el log service se desarrollaron en directorios separados.



La aplicación web presenta dos carpetas principales: main y test. La primera, presenta todo el código fuente de la aplicación, estructurado en la carpeta docker donde se encuentran los paquetes: connection, model y sparkwebserver. En el primero se encuentra la lógica del servidor interno que utiliza la aplicación para realizar la petición POST, el segundo contiene una clase mensaje para ser mapeada con los datos del mensaje recibidos desde el cliente web y por último, sparkwebserver presenta el servidor, desarrollado en SparkJava, que atiende la petición POST. La segunda cuenta con unos métodos para hacer pruebas unitarias.

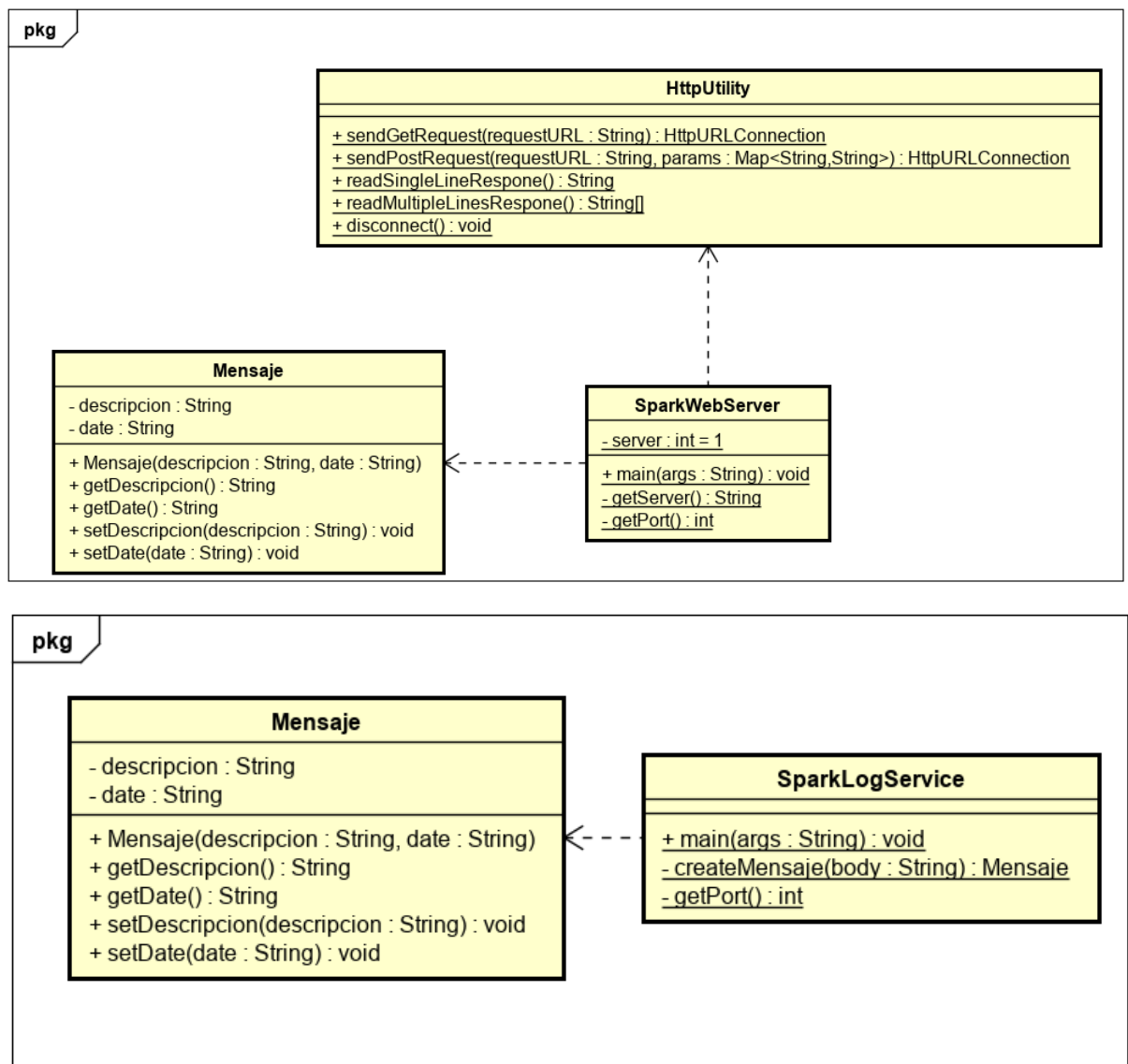




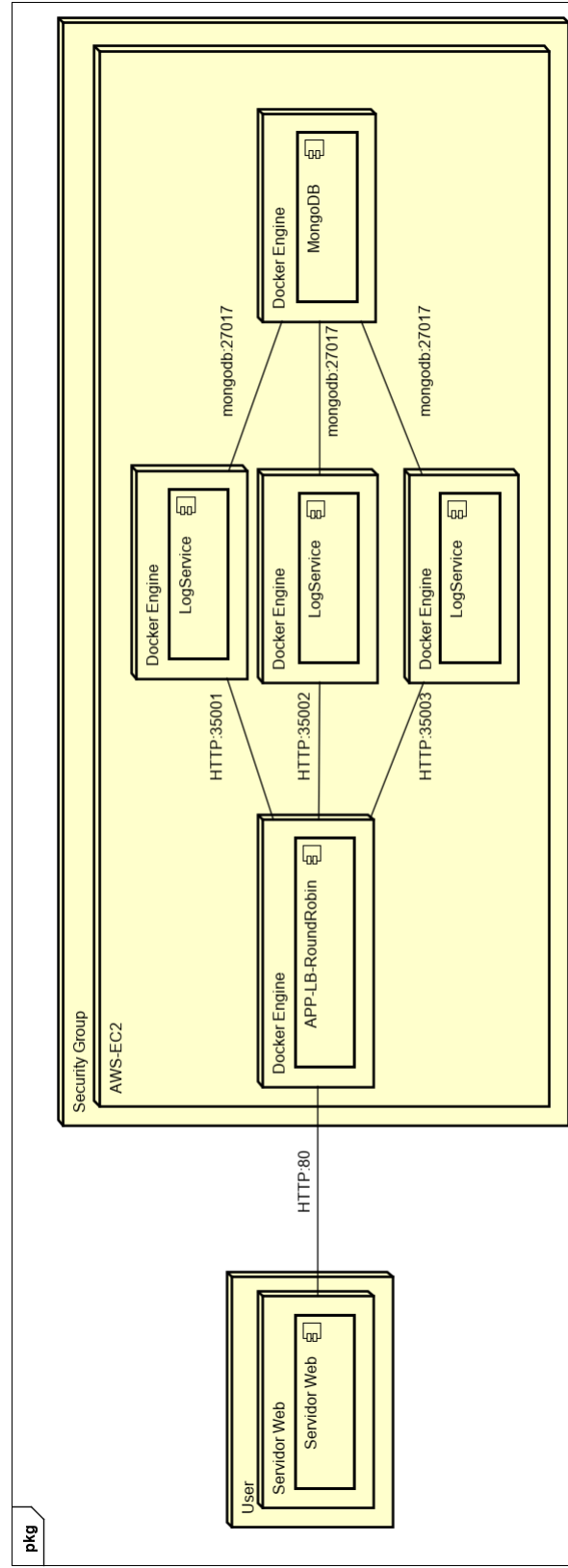
El servicio log también tiene a main y test como directorios principales. El primero, presente el código fuente de la aplicación, estructurado en la carpeta docker donde se encuentran los paquetes: model y sparklogservice. El primero, contiene la clase mensaje, diseñada para ser mapeada con los datos del mensaje obtenidos desde el servidor interno de la aplicación web. El segundo, contiene el servidor del log service que atiende las peticiones POST con los datos del mensaje y los almacena en la base de datos MongoDB. El segundo cuenta con unos métodos para hacer pruebas unitarias.

## 5 Diseño

Diseño de la aplicación web y del log service respectivamente.



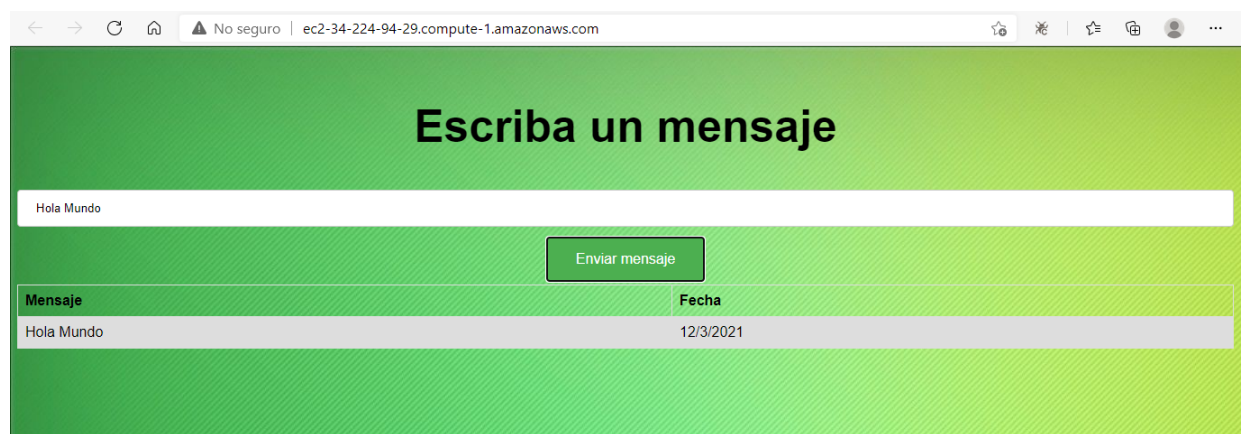
## 6 Arquitectura



## 7 Pruebas

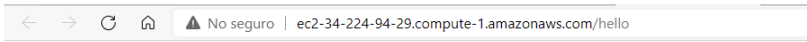
Se realizaron 4 pruebas para validar la correctitud del funcionamiento de la aplicación

La primera prueba valida que se agregue satisfactoriamente un mensaje



La segunda prueba valida que se agreguen satisfactoriamente cinco mensajes





Hola Mundo!