**DOCUMENTO PROYECTO FINAL DIPLOMADO JAVA**

Hanssel Ricardo Neira Mercado

Ingeniería de Sistemas, Corporación Unificada Nacional de Educación Superior(CUN)

Diplomado JAVA

Ing. Andres Rubiano

23 de abril de 2022

**TABLA DE CONTENIDO**

[1. ANALSIS DEL PROBLEMA 3](#_Toc101648713)

[2. REQUERIMIENTOS NO FUNCIONALES. 3](#_Toc101648714)

[3. REQUERIMIENTOS FUNCIONALES 3](#_Toc101648715)

[4. JUSTIFICACIÓN 3](#_Toc101648716)

[5. ALCANCES 4](#_Toc101648717)

[7. SOLUCIÓN 4](#_Toc101648718)

[8. CONCLUSIÓN 9](#_Toc101648719)

## ANALSIS DEL PROBLEMA

Una entidad del estado esta desarrollando un portal para gestionar sus servicios, para lograr ese fin, se planteó ya la arquitectura del sistema donde principalmente se desea aislar el backend del frontend siendo así que para todo el desarrollo en el front se planteó usar REACT y para la parte del back JAVA con el framework Spring Boot y en base de datos MYSQL. Actualmente se está desarrollando el Login del sistema, por ello un equipo se encuentra trabajando en el frontend, entonces se requiere un desarrollador que se encargue de realizar el backend para dicho fin.

## REQUERIMIENTOS NO FUNCIONALES.

* REQ 0188: Que la clave al ser recibida por el backend en formato JSON, sea encriptada al ser almacenadas en la base de datos
* REQ 0189: Que para acceder a los servicios del backend sea necesario que halla seguridad garantizada por tokens

## REQUERIMIENTOS FUNCIONALES

REQ 0288: El backend deberá poder recibir los mensajes del frontend en el momento de creación de un usuario donde se recibirá su nombre, apellido, email y password en formato JSON y deberá procesarlo y si aplica guardarlo en la base de datos

REQ 0289: Para que un objeto JSON enviado por el frontend aplique para ser añadido a la base de datos, este no deberá contener un email anteriormente guardado en la base de datos, si es así deberá imprimir un mensaje en consola indicando que está repetido

REQ 0290: Internamente al recibir el objeto JSON con el nuevo usuario el sistema deberá asignarle un ID que será un número consecutivo, también le asignará un user\_id de 40 dígitos aleatorios como la lleve principal.

REQ 0291: El backend tendrá un segundo endpoint para recibir un usuario y un password del frontend en formato JSON, comparará si estos datos existen en la base de datos y si es así devolver un mensaje 200 en caso contrario deberá enviar un mensaje 400.

## JUSTIFICACIÓN

En todos los sistemas informáticos es de vital importancia que exista un Login seguro que cumpla con requisitos de seguridad mínimos como lo es la encriptación del password y que haya tokens para la comunicación, dado que en la actualidad los problemas de seguridad han incrementado y aún más en las entidades del gobierno. Por ello, es vital que la encriptación del password se haga al añadir nuevos usuarios a la base de datos y que haya un protocolo de seguridad entre el backend y frontend por medio de tokens.

## ALCANCES

1. Este proyecto se enfocará solamente en el backend necesario para realizar las funcionalidades requeridas con JAVA JDK 11, Spring boot y MySQL. Por ende, se deberá seguir el framework Spring Boot.
2. Las pruebas del backend se realizarán por medio de postman para validar que los servicios están recibiendo y enviando la información correcta.
3. El backend deberá recibir mensajes tipo JSON para la creación de usuarios y solamente en caso de que haya campos nulos o el email ya esté en la base de datos rechazará la petición de creación
4. Para el Login el backend deberá comparar el mensaje JSON que viene del frontend con el email y password, y en caso de que corresponda los datos con uno existente en la base de datos devolverá un mensaje 200, en caso contrario enviará un mensaje 400.
5. Se creerá un paquete para las entidades, modelos, seguridad, servicios y uno compartido para implementar una DTO que se comunique con todas las capas.
6. Para el control de versiones se utilizará GITHUB.

## SOLUCIÓN

El proyecto se realizó de manera satisfactoria cumpliendo con los requerimientos funcionales y no funcionales del sistema.

Aquí se puede ver como se generaron los respectivos módulos determinado en el alcance:

Captura de pantalla de computadora

Descripción generada automáticamente

Esta es la respuesta al momento de generación de un nuevo usuario y no se envía un dato:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

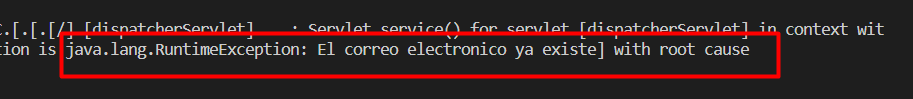
Esta es la respuesta cuando se envían todos los datos, pero ya hay un email en la base datos:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



Esta es la respuesta cuando se envían todos los datos y el correo no está repetido:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Esta por su parte, es la respuesta cuando en el endpoint de Login se envían datos incorrectos para el Login:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Y este es cuando el usuario y la contraseña si corresponde a un usuario en la base de datos:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Repositorio en GitHub:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Por último, así luce la base de datos:

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

## CONCLUSIÓN

La verdad fue una muy buena experiencia este proyecto, dado que me permitió tener un acercamiento más real a la industria. Actualmente, es raro ver empresas que realicen sus interfaces graficas en JAVA por ejemplo o que no utilicen un framework para su backend. Entonces poder trabajar con estas tecnologías, aunque es un poco frustrante al principio, resulta muy retador y permite en un futuro poder aplicar o ofertas laborales con un mejor background.

La arquitectura propuesta en este proyecto donde se aísla completamente el backend y el frontend es una buena practica a mi parecer en términos de la escalabilidad, dado que, si tengo un servicio de Login que se comunica con un protocolo definido con un front, para el back es transparente si el front se desarrolla en Angular, React u otros.

La curva de aprendizajes de Spring Boot es dura a mi parecer, dado que, si bien se tiene algunas bases de JAVA en la carrera, en las carreras universitarias no se hace énfasis a cómo enfrentarse a un framework, aprender en sí un framework, entonces resulta un poco más difícil su aprendizaje.

La arquitectura de inyección de dependencias es vital en la industrial, pero igual, en el pregrado no se nos explica muchos de patrones de Arquitectura, los cuales son bastante importantes para evitar malas practicas en el desarrollo cuando se inicia como junior.