# **Evaluación: JAVA**

Desarrolle una aplicación que exponga una API RESTful de creación de usuarios.

Todos los endpoints deben aceptar y retornar solamente JSON, inclusive al para los mensajes de error.

Todos los mensajes deben seguir el formato:

{"mensaje": "mensaje de error"}

## **Registro**

* Ese endpoint deberá recibir un usuario con los campos "nombre", "correo", "contraseña", más un listado de objetos "teléfono", respetando el siguiente formato:

{

"name": "Juan Rodriguez",

"email": "juan@rodriguez.org",

"password": "hunter2",

"phones": [

{

"number": "1234567",

"citycode": "1",

"contrycode": "57"

}

]

}

* Responder el código de status HTTP adecuado
* En caso de éxito, retorne el usuario y los siguientes campos:
  + id: id del usuario (puede ser lo que se genera por el banco de datos, pero sería más deseable un UUID)
  + created: fecha de creación del usuario
  + modified: fecha de la última actualización de usuario
  + last\_login: del último ingreso (en caso de nuevo usuario, va a coincidir con la fecha de creación)
  + token: token de acceso de la API (puede ser UUID o JWT)
  + isactive: Indica si el usuario sigue habilitado dentro del sistema.
* Si caso el correo conste en la base de datos, deberá retornar un error "El correo ya registrado".
* El correo debe seguir una expresión regular para validar que formato sea el correcto. (aaaaaaa@dominio.cl)
* La clave debe seguir una expresión regular para validar que formato sea el correcto. (Una Mayuscula, letras minúsculas, y dos números)
* El token deberá ser persistido junto con el usuario

## **Requisitos mandatorios**

* Plazo: 2 días.
* Banco de datos en memoria.
* Gradle como herramienta de construcción.
* Pruebas unitarias (Deseable: Spock Framework).
* Persistencia con Hibernate.
* Framework Spring Boot.
* Java 8 o superior. (Usar características propias de la versión)
* Entrega en un repositorio público (github, gitlab o bitbucket) con el código fuente.
* Entregar diagrama de componentes de la solución y al menos un diagrama de secuencia.
* README.md debe contener las instrucciones para levantar y usar el proyecto.

## **Requisitos deseables**

* JWT cómo token

Diagrama de Componentes:

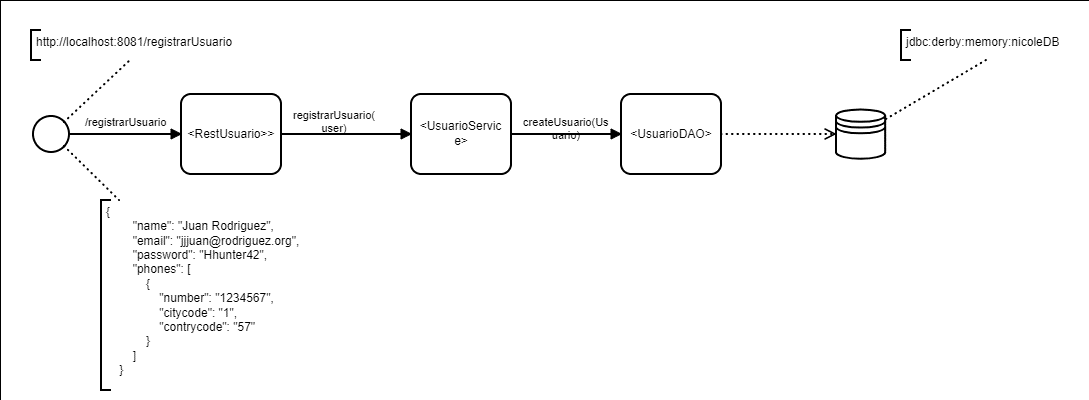
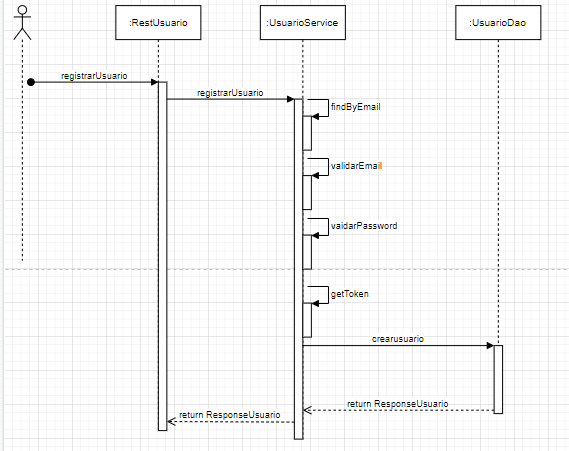
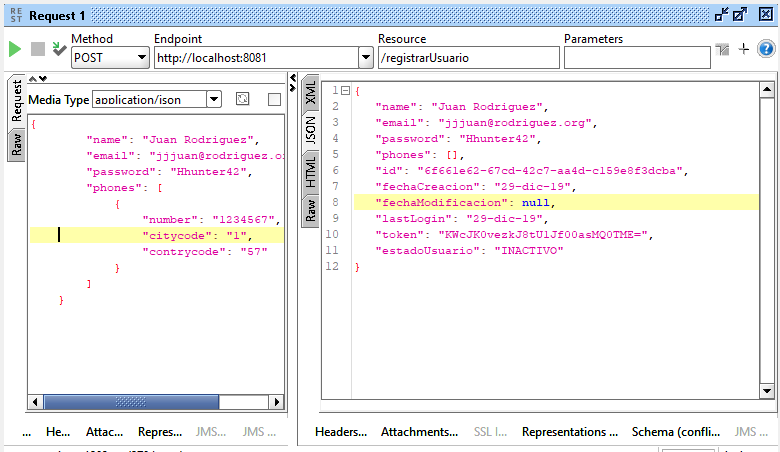


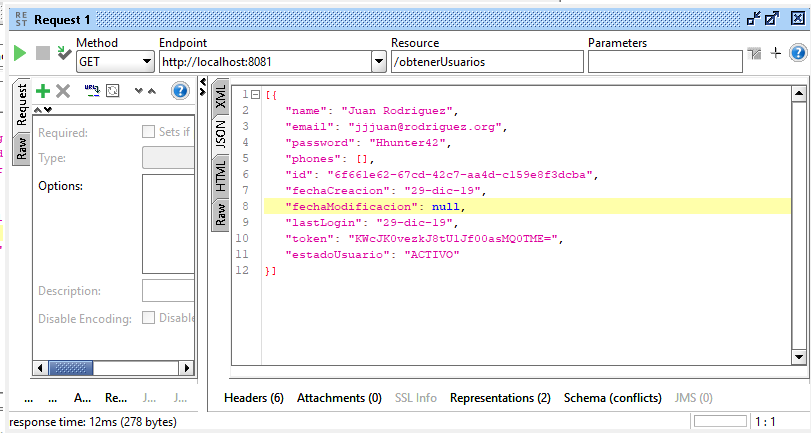
Diagrama de Secuencia



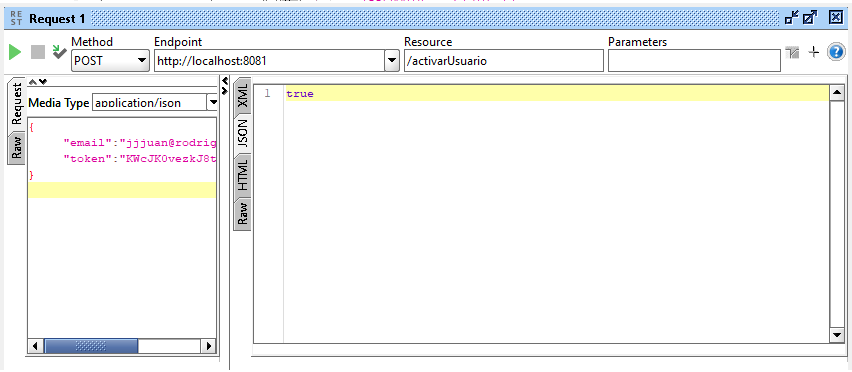
Servicio RegistrarUsuario



Servicio ObtenerUsuarios



Servico Activarusuario



Resultado Testing with Spock Framework

