

# Karate

**NOMBRE:** Roddy Steeven Zamora Rodríguez

## EJERCICIO:

Utilizando un software para pruebas de servicios REST) realizar una prueba de cada uno de estos servicios:

Signup: <https://api.demoblaze.com/signup>

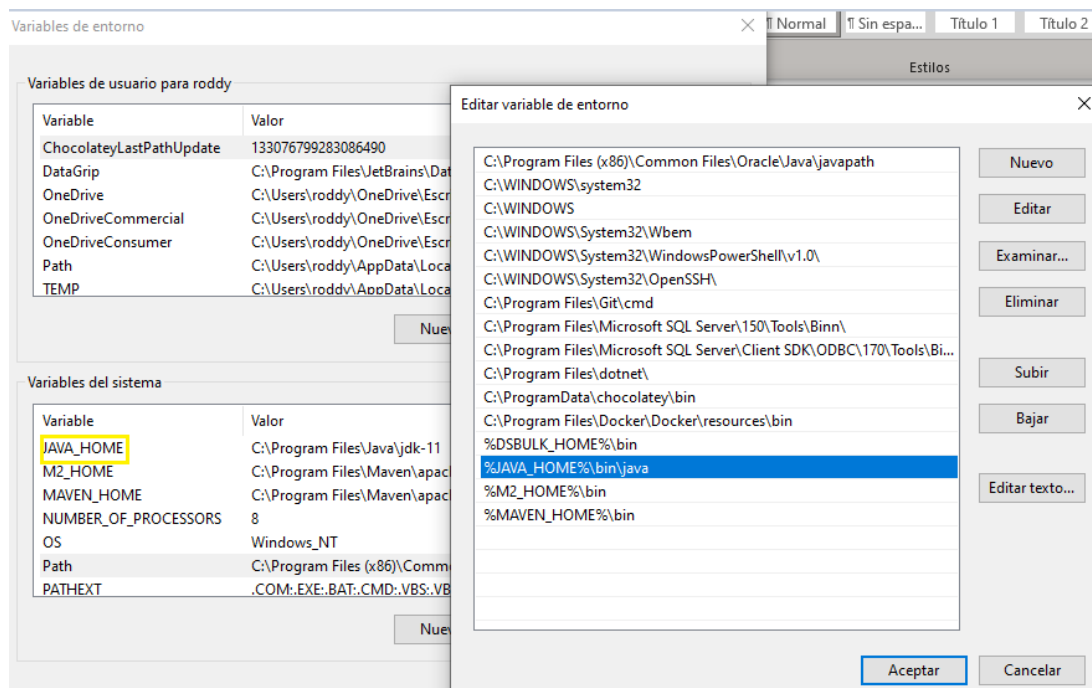
Login: <https://api.demoblaze.com/login>

Identificar las entradas y capturar las salidas en cada uno de los siguientes casos:

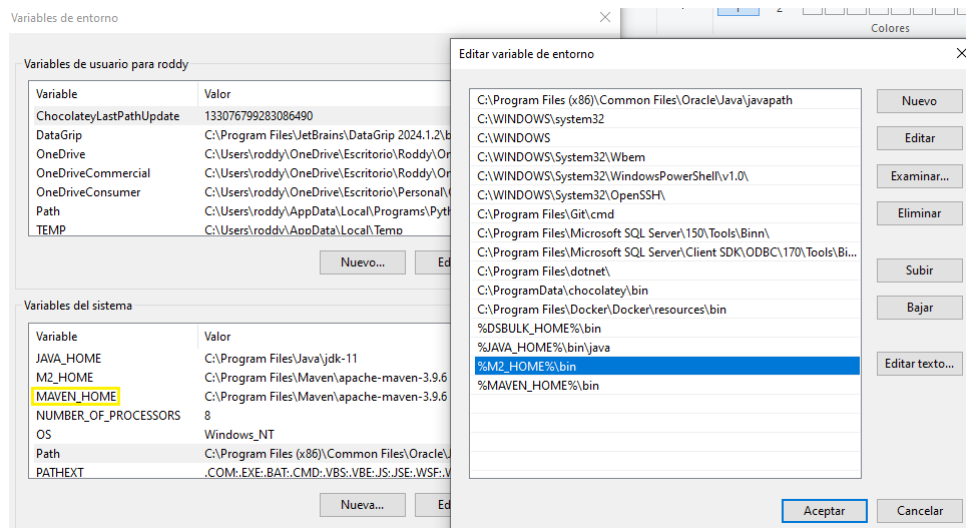
- Crear un nuevo usuario en signup
- Intentar crear un usuario ya existente
- Usuario y password correcto en login
- Usuario y password incorrecto en login

## PRERREQUISITOS:

1.- Descargar e instalar Java 11 y configurar su variable de entorno.



2.- Descargar e instalar Maven y configurar su variable de entorno  
<https://maven.apache.org/download.cgi>.



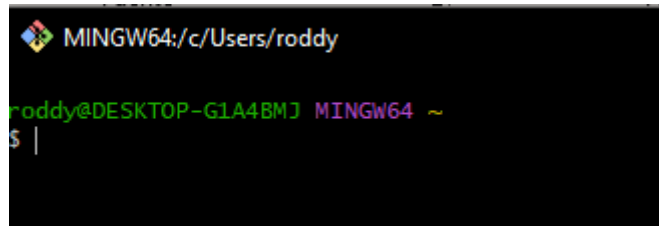
```
C:\Users\roddy>mvn -v
Apache Maven 3.9.6 (bc0240f3c744dd6b6ec2920b3cd08dcc295161ae)
Maven home: C:\Program Files\Maven\apache-maven-3.9.6
Java version: 11, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-11
Default locale: es_EC, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```

3. Descargar un IDE para gestionar las pruebas automatizadas, en este caso IntelliJ-Idea <https://www.jetbrains.com/idea/download/>

4. Generar un proyecto gradle y agregar las dependencias respectivas para usar el Framework karate

```
8  <v> repositories {
9  <v>     maven {
10     |     url 'https://repo.maven.apache.org/maven2'
11     |
12     }
13 }
14 <v> dependencies {
15 |
16     testImplementation 'com.intuit.karate:karate-junit5:1.2.0.RC4'
17 }
18
19 <v> test {
20     useJUnitPlatform()
21 }
22
23 <v> sourceSets {
24 <v>     test{
25     <v>         resources{
26         |             srcDir file('src/test/java')
27         |             exclude '**/*.java'
28         |
29         }
30     }
31 }
```

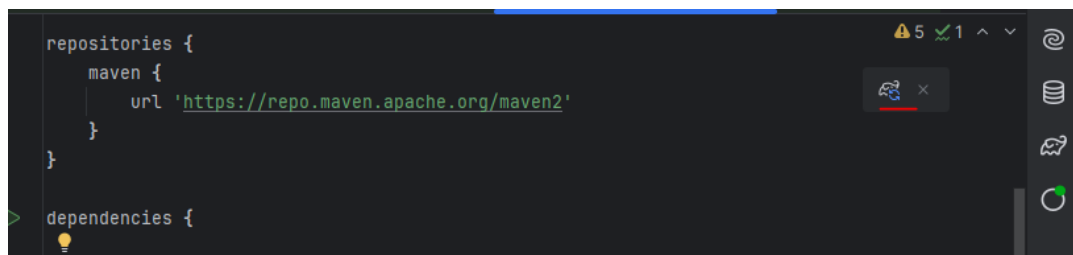
5. Descargar e instalar git <https://git-scm.com/downloads>.

A terminal window with a black background. The title bar reads "MINGW64:/c/Users/roddy". The prompt is "roddy@DESKTOP-G1A4BMJ MINGW64 ~". The cursor is on a new line after a space and a vertical bar.

```
MINGW64:/c/Users/roddy  
roddy@DESKTOP-G1A4BMJ MINGW64 ~  
$ |
```

6. Instalar las dependencias del proyecto base.

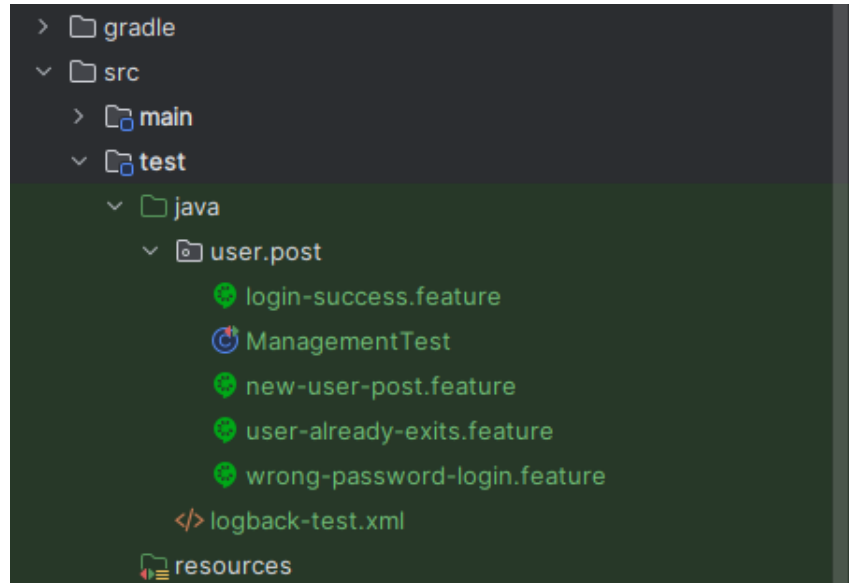
Con el gradle configurado procedemos a descargar las dependencias.

A screenshot of an IDE showing a Gradle configuration snippet. The code defines a Maven repository and a dependencies block. The IDE interface includes a top bar with icons for errors, warnings, and a search icon, and a right sidebar with icons for a file explorer, a run and debug console, and a settings menu.

```
repositories {  
    maven {  
        url 'https://repo.maven.apache.org/maven2'  
    }  
}  
  
> dependencies {  
    ⚡
```

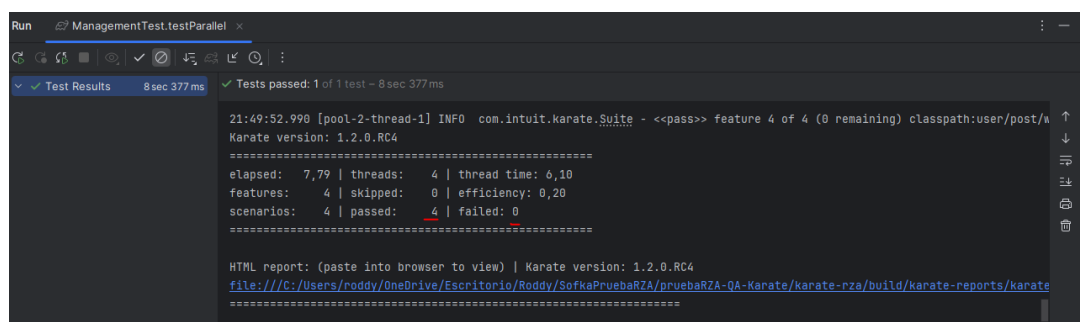
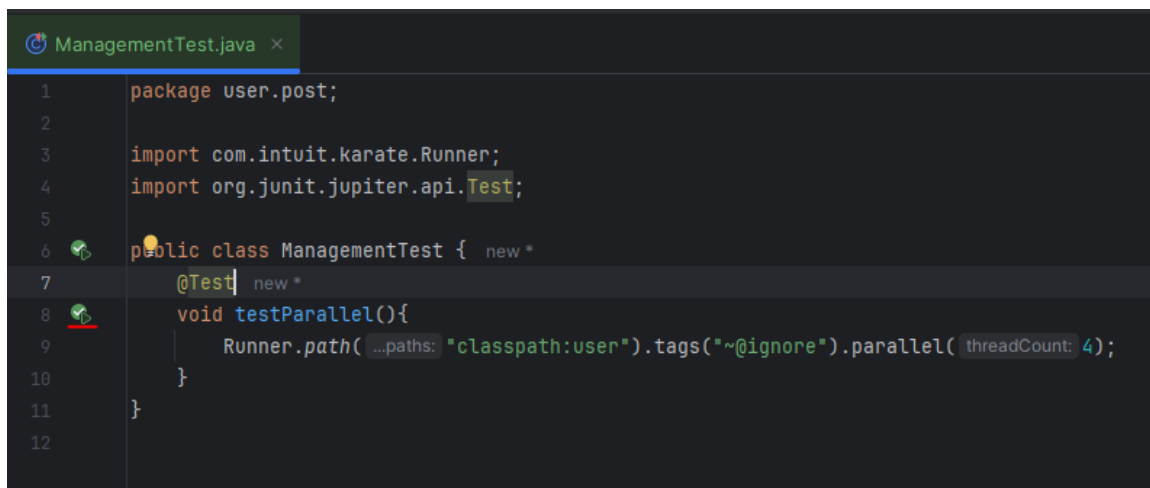
## DESARROLLO:

Con los requisitos instalados se procede a crear los feautres respectivos con los escenarios.




## EJECUCIÓN:

Para agrupar la ejecución de los features en una sola ejecución nos ayudamos de la clase 'ManagementTest'.



Al final se nos mostrara un reporte cual podremos ver el detalle de la ejecución.



4

0

Features  
2024-05-22 09:49:53 p. m.

Tags | Timeline

Feature	Title	Passed	Failed	Scenarios	Time (ms)
user/post/login-success.feature	Metodo login	1	0	1	1486
user/post/new-user-post.feature	Metodo post signup	1	0	1	1470
user/post/wrong-password-login.feature	Metodo login	1	0	1	1570
user/post/user-already-exits.feature	Metodo post signup	1	0	1	1570

