1- Utilizando Unit test

¿En el proyecto spring-boot para qué está esta dependencia en el pom.xml?

```
<dependency>
                  <groupId>org.springframework.boot</groupId>
                  <artifactId>spring-boot-starter-test</artifactId>
                  <scope>test</scope>
</dependency>
```

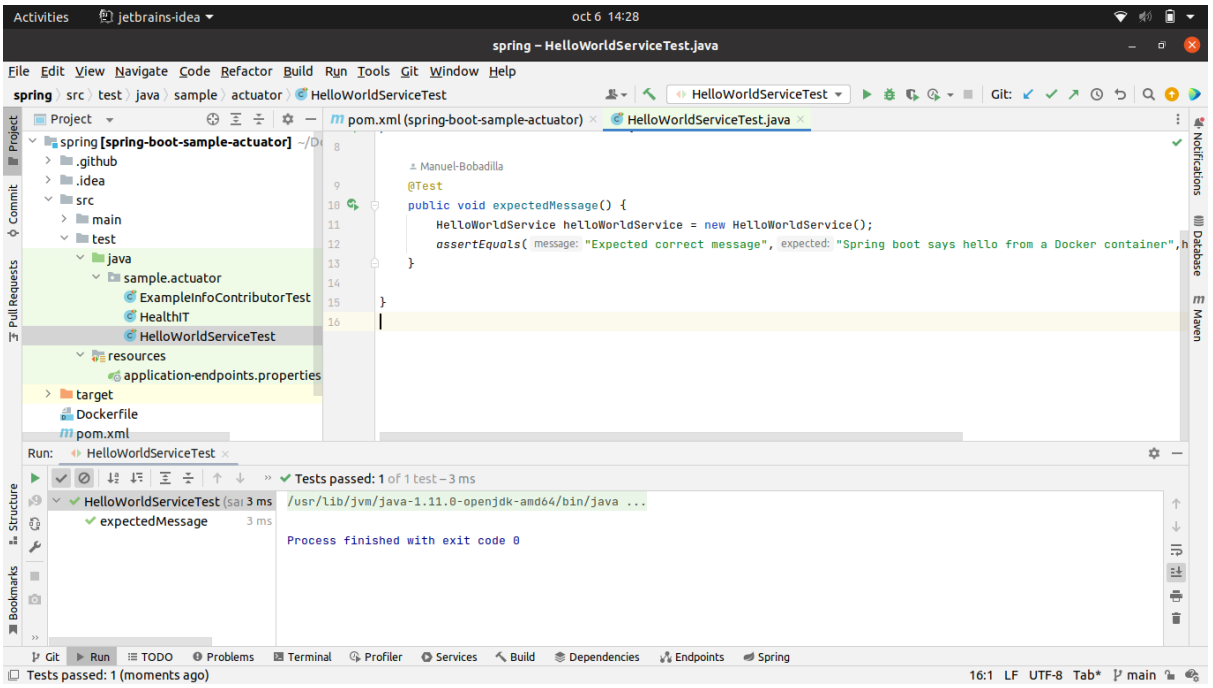
El scope test implica que la dependencia no es requerida para el uso normal de la aplicación, y solo está disponible para el testeo de la compilación y ejecución de pruebas

Analizar y ejecutar el método de unit test:

```
public class HelloWorldServiceTest {

      @Test
      public void expectedMessage() {
            HelloWorldService helloWorldService = new HelloWorldService();
             assertEquals("Expected correct message","Spring boot says hello
from a Docker container",helloWorldService.getHelloMessage());
      }
}
```

@Test Identifica un método como un método de prueba, a grandes rasgos, crea una instancia de una clase HelloWorldService, y mediante el método assertEquals compara el mensaje obtenido de llamar un método de la clase con el mensaje esperado, si es así imprime que fue correcto

Ejecutar los tests utilizando la IDE



Ejecutando el test desde la consola

```
manuel@manuel-Aspire-A315-54K:~/Documents/spring$ mvn test
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by
com.google.inject.internal.cglib.core.$ReflectUtils$1
(file:/usr/share/maven/lib/guice.jar) to method
java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,java.security.Protecti
onDomain)
WARNING: Please consider reporting this to the maintainers of
com.google.inject.internal.cglib.core.$ReflectUtils$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective
access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Scanning for projects...
[INFO]
[INFO] --------< org.springframework.boot:spring-boot-sample-actuator >--------
[INFO] Building Spring Boot Actuator Sample 2.0.2
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:3.0.1:resources (default-resources) @
```

```
spring-boot-sample-actuator ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.7.0:compile (default-compile) @
spring-boot-sample-actuator ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 6 source files to /home/manuel/Documents/spring/target/classes
[INFO]
[INFO] --- maven-resources-plugin:3.0.1:testResources (default-testResources) @
spring-boot-sample-actuator ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.7.0:testCompile (default-testCompile) @
spring-boot-sample-actuator ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 3 source files to /home/manuel/Documents/spring/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.9:test (default-test) @ spring-boot-sample-actuator
---
[INFO] Surefire report directory: /home/manuel/Documents/spring/target/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running sample.actuator.HelloWorldServiceTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.102 sec
Running sample.actuator.ExampleInfoContributorTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.616 sec

Results :

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  5.395 s
[INFO] Finished at: 2022-10-06T14:30:12-03:00
[INFO] ------------------------------------------------------------------------
```

3- Familiarizarse con algunos conceptos de Mockito

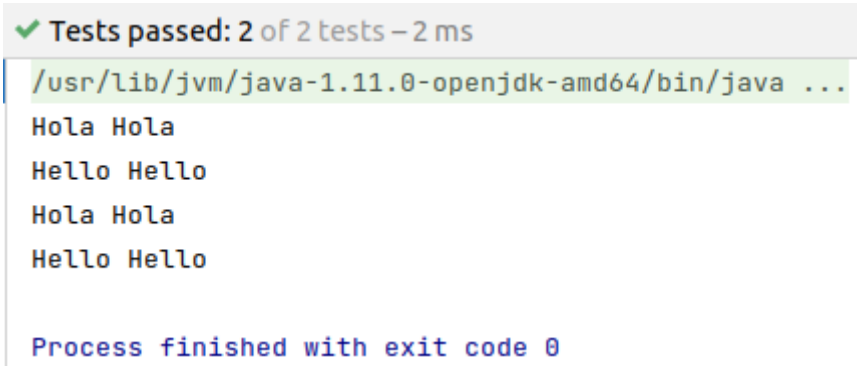
Analizar el código del test

```
public class ExampleInfoContributorTest {

        @Test
        public void infoMap() {
                Info.Builder builder = mock(Info.Builder.class);

                ExampleInfoContributor exampleInfoContributor = new ExampleInfoContributor();
                exampleInfoContributor.contribute(builder);

                verify(builder).withDetail(any(),any());
        }
}
```

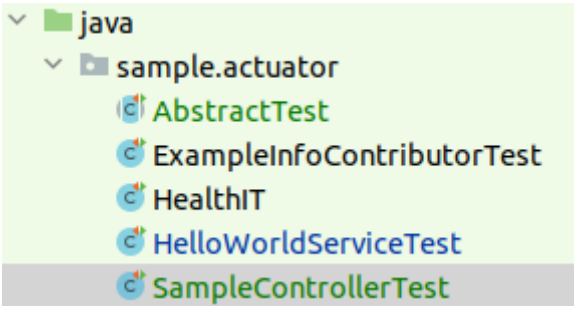
La creación de un objeto mock permite simular un objeto real y así elimina dependencia y permite aislar los test. Mockito Verify methods are used to check that certain behavior happened. We can use Mockito verify methods at the end of the testing method code to make sure that specified methods are called.

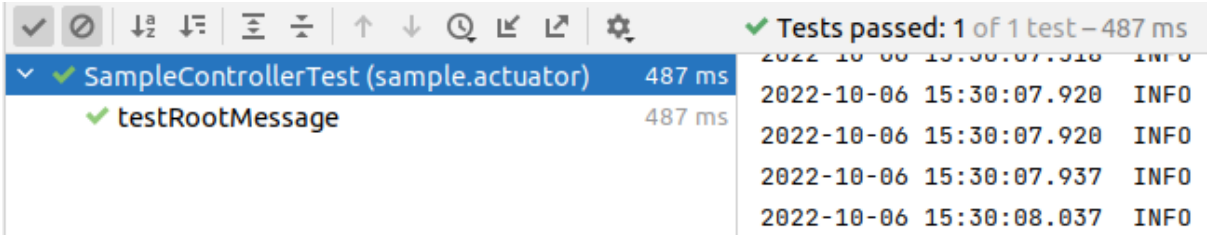
4- Utilizando Mocks

Agregar un unit test a la clase HelloWorldServiceTest




```
@Test
	public void printHelloMessage(){
		for (int i = 0; i < 4; i++){
			System.out.println(getHelloMessage());
		}
	}

	public String getHelloMessage() {
		String helloMessage;
		if (tested){
			helloMessage = helloHello();
		}else {
			helloMessage = holaHola();
		}
		tested = !tested;
		return helloMessage;
	}

	public String helloHello() {
		return "Hello Hello";
	}

	public String holaHola() {
		return "Hola Hola";
	}
```





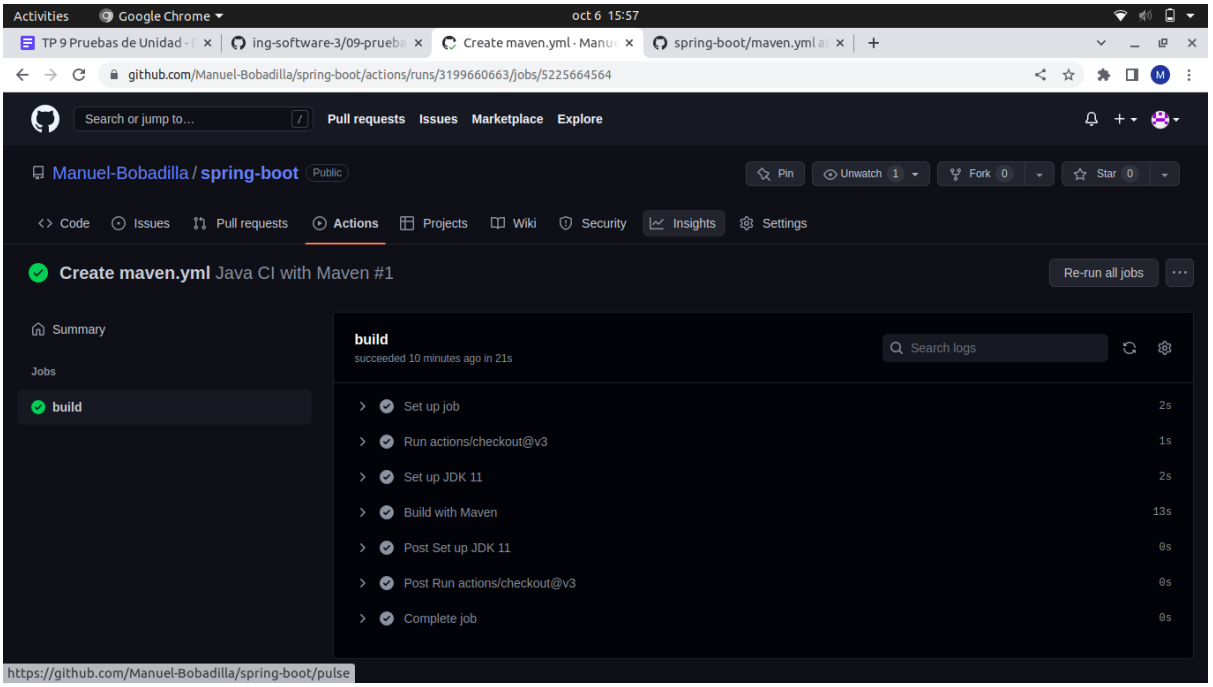
El test comprueba que el mensaje obtenido en el directorio raíz de Spring boot sea hello from a Docker container y que el status de la página sea 200 (ok)

6- Capturar los unit tests como parte del proceso de CI/CD

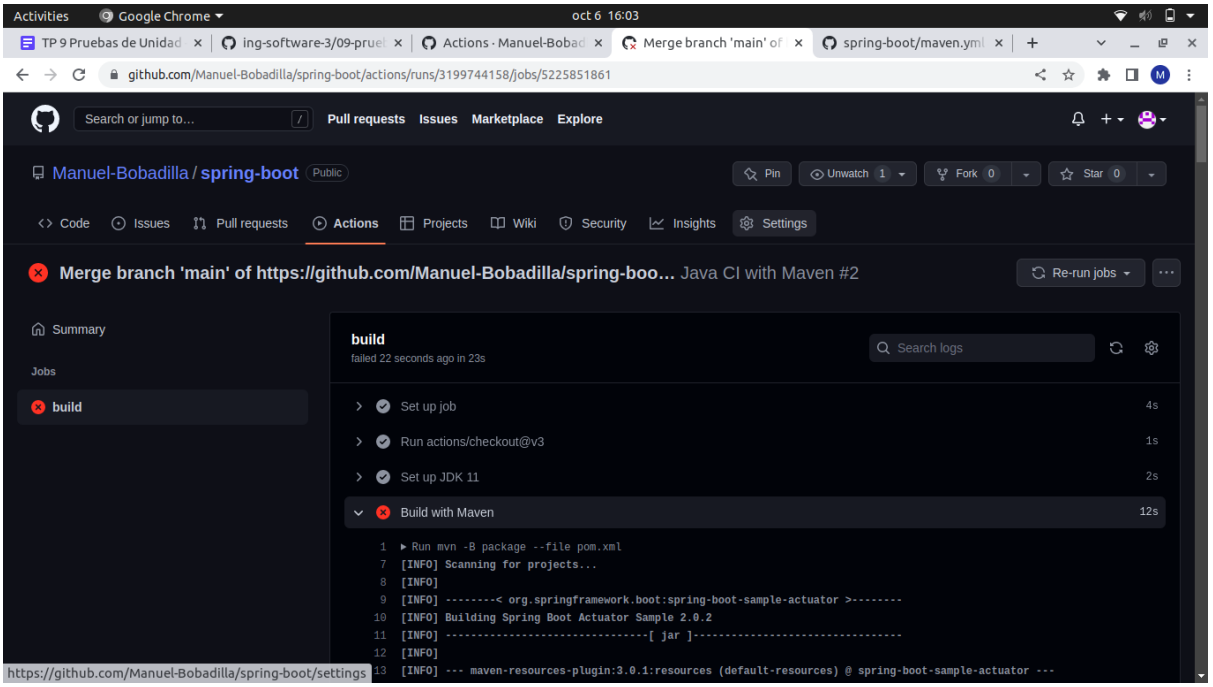
Código de github actions

```yaml
name: Java CI with Maven

on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]

jobs:
  build:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v3
    - name: Set up JDK 11
      uses: actions/setup-java@v3
      with:
        java-version: '11'
        distribution: 'temurin'
        cache: maven
    - name: Build with Maven
      run: mvn -B package --file pom.xml
```

Corremos el test de integración



Modifico el código de los test para que falle a propósito, para observar si en el github actions se muestra que fallan los tests



Vemos que el fallo en los tests se reflejan en github actions, ahora correjimos el error y pusheamos de vuelta