Marina Madrid García

https://github.com/MarinaMadrid/m1-08-jersey-coche-test-maven

SpotBugs

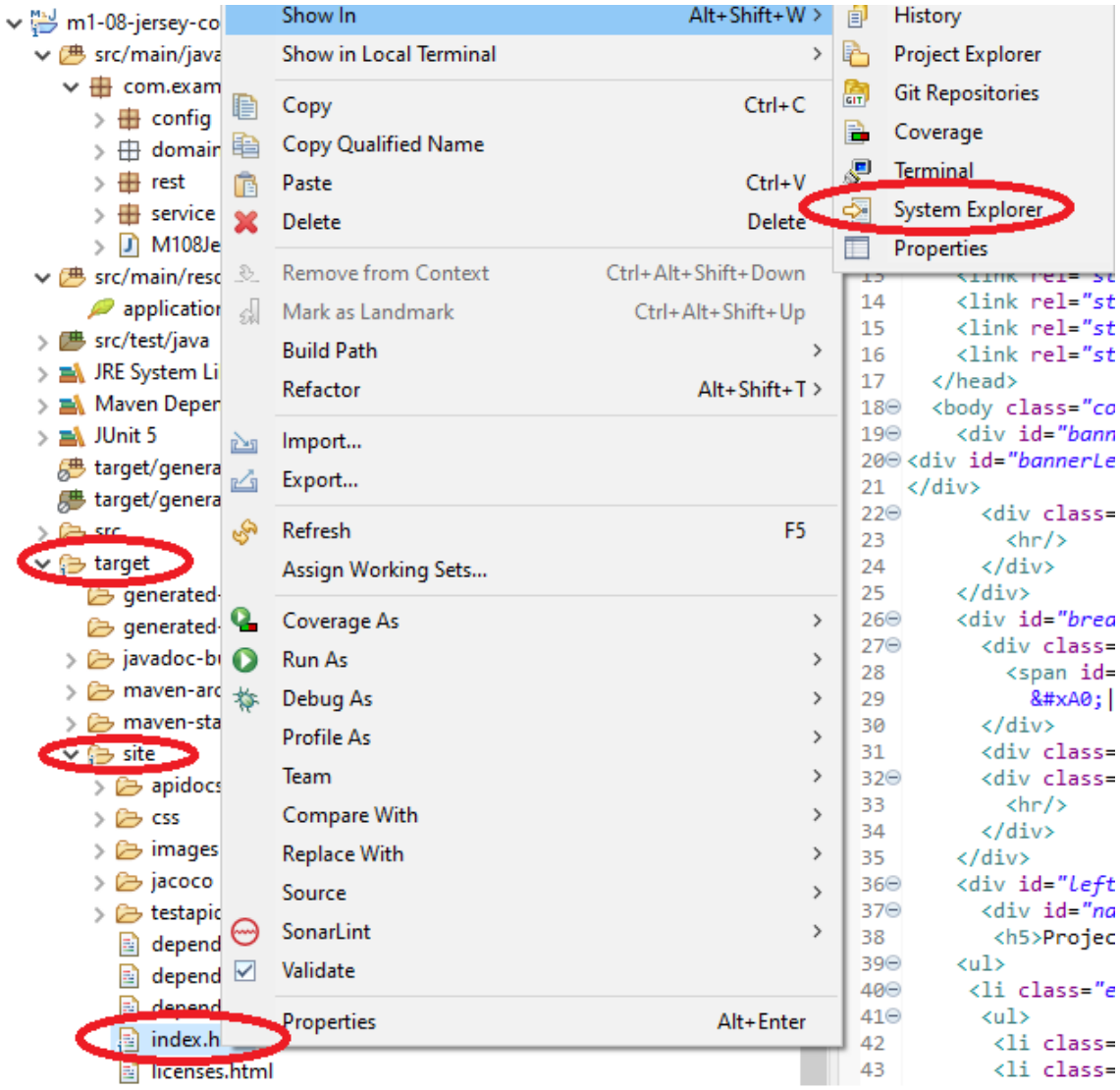
1. He usado el proyecto entregado de Maven porque es la versión que presenta menos errores, y además ya tiene algunas dependencias instaladas. La lista de los plugins necesarios en el pom.xml son:

   - maven-clean-plugin (build)
   - maven-compiler-plugin (build)
   - maven-surefire-plugin (build)
   - maven-site-plugin (build)
   - maven-surefire-report-plugin (reporting)
   - maven-javadoc-plugin (en build con executions y en reporting sin executions)
   - jacoco-maven-plugin (en build con executions y en reporting sin executions)
   - maven-checkstyle-plugin (reporting)

   Y los comandos que se ejecutan para clean y site son:

   - `mvn clean`
   - `mvn install`
   - `mvn clean install`
   - `mvn site`
   - `mvn compile site`

2. Una vez que se haya creado todo, se abre el archivo index.html que está dentro de la carpeta site de target.


m1-08-jersey-co
src/main/java
com.exam
config
domain
rest
service
M108Je
src/main/resc
applicatior
src/test/java
JRE System Li
Maven Deper
JUnit 5
target/genera
target/genera
src
target
generated-
generated-
javadoc-b
maven-arc
maven-sta
site
apidocs
css
images
jacoco
testapic
depend
depend
depend
index.h
licenses.html
Show In Alt+Shift+W
Show in Local Terminal
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Mark as Landmark Ctrl+Alt+Shift+Up
Build Path
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Assign Working Sets...
Coverage As
Run As
Debug As
Profile As
Team
Compare With
Replace With
Source
SonarLint
Validate
Properties Alt+Enter
History
Project Explorer
Git Repositories
Coverage
Terminal
System Explorer
Properties

Nombre	Fecha de modificación	Tipo	Tamaño
apidocs	14/09/2021 12:00	Carpeta de archivos	
css	14/09/2021 11:57	Carpeta de archivos	
images	14/09/2021 11:58	Carpeta de archivos	
jacoco	14/09/2021 11:56	Carpeta de archivos	
jacoco-aggregate	14/09/2021 11:57	Carpeta de archivos	
testapidocs	14/09/2021 12:00	Carpeta de archivos	
checkstyle.html	14/09/2021 11:57	Firefox HTML Doc...	0 KB
cpd.html	14/09/2021 12:00	Firefox HTML Doc...	4 KB
dependencies.html	14/09/2021 12:00	Firefox HTML Doc...	143 KB
dependency-info.html	14/09/2021 12:00	Firefox HTML Doc...	5 KB
dependency-management.html	14/09/2021 12:00	Firefox HTML Doc...	411 KB
index.html	14/09/2021 12:00	Firefox HTML Doc...	4 KB
licenses.html	14/09/2021 12:00	Firefox HTML Doc...	42 KB
plugin-management.html	14/09/2021 12:00	Firefox HTML Doc...	9 KB
plugins.html	14/09/2021 12:00	Firefox HTML Doc...	8 KB
pmd.html	14/09/2021 12:00	Firefox HTML Doc...	4 KB
project-info.html	14/09/2021 12:00	Firefox HTML Doc...	6 KB
project-reports.html	14/09/2021 12:00	Firefox HTML Doc...	5 KB
scm.html	14/09/2021 12:00	Firefox HTML Doc...	4 KB
spotbugs.html	14/09/2021 12:00	Firefox HTML Doc...	23 KB
summary.html	14/09/2021 12:00	Firefox HTML Doc...	5 KB
surefire-report.html	14/09/2021 12:00	Firefox HTML Doc...	20 KB
team.html	14/09/2021 12:00	Firefox HTML Doc...	5 KB

3. Dentro de index.html, se abre projects reports, y ahí a SpotBugs.

m1-08-jersey-coche

Last Published: 2021-09-14 | Version: 0.0.1-SNAPSHOT m1-08-jersey-coche

Project Documentation
- Project Information
  - Dependencies
  - Dependency Information
  - Dependency Management
  - **About**
  - Licenses
  - Plugin Management
  - Plugins
  - Source Code Management
  - Summary
  - Team
- Project Reports

Built by: maven

About m1-08-jersey-coche

Demo project for Spring Boot

Copyright © 2021..

Generated Reports

This document provides an overview of the various reports that are automatically generated by Maven . Each report is briefly described below.

Overview

Document	Description
Javadoc	Javadoc API documentation.
Test Javadoc	Test Javadoc API documentation.
Surefire Report	Report on the test results of the project.
JaCoCo	JaCoCo Coverage Report.
JaCoCo Aggregate	JaCoCo Aggregate Coverage Report.
CPD	Duplicate code detection.
PMD	Verification of coding rules.
SpotBugs	Generates a source code report with the SpotBugs Library.

Copyright © 2021..

4. Y este es el reporte que ha emitido SpotBugs sobre el proyecto:

SpotBugs Bug Detector Report

The following document contains the results of SpotBugs

SpotBugs Version is *4.3.0*

Threshold is *medium*

Effort is *default*

Summary

Classes	Bugs	Errors	Missing Classes
25	37	0	0

Files

Class	Bugs
com.example.demo.domain.vehicles.CombustionVehicle	3
com.example.demo.domain.vehicles.ElectricVehicle	3
com.example.demo.domain.vehicles.HybridVehicle	3
com.example.demo.domain.vehicles.Vehicle	9
com.example.demo.rest.CombustionVehicleController	1
com.example.demo.rest.ElectricVehicleController	1
com.example.demo.rest.HybridVehicleController	1
com.example.demo.service.CombustionVehicleServiceImpl	5
com.example.demo.service.ElectricVehicleServiceImpl	6
com.example.demo.service.HybridVehicleServiceImpl	5

com.example.demo.domain.vehicles.CombustionVehicle

Bug	Category	Details	Line	Priority
com.example.demo.domain.vehicles.CombustionVehicle.getInjection() may expose internal representation by returning CombustionVehicle.injection	MALICIOUS_CODE	EI_EXPOSE_REP	20	Medium
new com.example.demo.domain.vehicles.CombustionVehicle(Long, String, String, Integer, AirConditioning, Battery, Engine, Injection, Boolean) may expose internal representation by storing an externally mutable object into CombustionVehicle.injection	MALICIOUS_CODE	EI_EXPOSE_REP2	16	Medium
com.example.demo.domain.vehicles.CombustionVehicle.setInjection(Injection) may expose internal representation by storing an externally mutable object into CombustionVehicle.injection	MALICIOUS_CODE	EI_EXPOSE_REP2	24	Medium

com.example.demo.domain.vehicles.ElectricVehicle

Bug	Category	Details	Line	Priority
com.example.demo.domain.vehicles.ElectricVehicle.getCharges() may expose internal representation by returning ElectricVehicle.charges	MALICIOUS_CODE	EI_EXPOSE_REP	30	Medium
new com.example.demo.domain.vehicles.ElectricVehicle(Long, String, String, Integer, AirConditioning, Battery, Engine, Integer, Charges, Boolean) may expose internal representation by storing an externally mutable object into ElectricVehicle.charges	MALICIOUS_CODE	EI_EXPOSE_REP2	18	Medium
com.example.demo.domain.vehicles.ElectricVehicle.setCharges(Charges) may expose internal representation by storing an externally mutable object into ElectricVehicle.charges	MALICIOUS_CODE	EI_EXPOSE_REP2	34	Medium

com.example.demo.domain.vehicles.HybridVehicle

Bug	Category	Details	Line	Priority
com.example.demo.domain.vehicles.HybridVehicle.getHydrogenTank() may expose internal representation by returning HybridVehicle.hydrogenTank	MALICIOUS_CODE	EI_EXPOSE_REP	22	Medium
new com.example.demo.domain.vehicles.HybridVehicle(Long, String, String, Integer, AirConditioning, Battery, Engine, HydrogenTank, Boolean) may expose internal representation by storing an externally mutable object into HybridVehicle.hydrogenTank	MALICIOUS_CODE	EI_EXPOSE_REP2	18	Medium
com.example.demo.domain.vehicles.HybridVehicle.setHydrogenTank(HydrogenTank) may expose internal representation by storing an externally mutable object into HybridVehicle.hydrogenTank	MALICIOUS_CODE	EI_EXPOSE_REP2	26	Medium

com.example.demo.domain.vehicles.Vehicle

Bug	Category	Details	Line	Priority
com.example.demo.domain.vehicles.Vehicle.getAirAc() may expose internal representation by returning Vehicle.airAc	MALICIOUS_CODE	EI_EXPOSE_REP	51	Medium
com.example.demo.domain.vehicles.Vehicle.getBattery() may expose internal representation by returning Vehicle.battery	MALICIOUS_CODE	EI_EXPOSE_REP	55	Medium
com.example.demo.domain.vehicles.Vehicle.getEngine() may expose internal representation by returning Vehicle.engine	MALICIOUS_CODE	EI_EXPOSE_REP	59	Medium
new com.example.demo.domain.vehicles.Vehicle(Long, String, String, Integer, AirConditioning, Battery, Engine, Boolean) may expose internal representation by storing an externally mutable object into Vehicle.airAc	MALICIOUS_CODE	EI_EXPOSE_REP2	27	Medium
new com.example.demo.domain.vehicles.Vehicle(Long, String, String, Integer, AirConditioning, Battery, Engine, Boolean) may expose internal representation by storing an externally mutable object into Vehicle.battery	MALICIOUS_CODE	EI_EXPOSE_REP2	28	Medium
new com.example.demo.domain.vehicles.Vehicle(Long, String, String, Integer, AirConditioning, Battery, Engine, Boolean) may expose internal representation by storing an externally mutable object into Vehicle.engine	MALICIOUS_CODE	EI_EXPOSE_REP2	29	Medium
com.example.demo.domain.vehicles.Vehicle.setAirAc(AirConditioning) may expose internal representation by storing an externally mutable object into Vehicle.airAc	MALICIOUS_CODE	EI_EXPOSE_REP2	79	Medium
com.example.demo.domain.vehicles.Vehicle.setBattery(Battery) may expose internal representation by storing an externally mutable object into Vehicle.battery	MALICIOUS_CODE	EI_EXPOSE_REP2	83	Medium
com.example.demo.domain.vehicles.Vehicle.setEngine(Engine) may expose internal representation by storing an externally mutable object into Vehicle.engine	MALICIOUS_CODE	EI_EXPOSE_REP2	87	Medium

com.example.demo.rest.CombustionVehicleController

Bug	Category	Details	Line	Priority
new com.example.demo.rest.CombustionVehicleController(CombustionVehicleService) may expose internal representation by storing an externally mutable object into CombustionVehicleController.combustionVehicleService	MALICIOUS_CODE	EI_EXPOSE_REP2	30	Medium

com.example.demo.rest.ElectricVehicleController

Bug	Category	Details	Line	Priority
new com.example.demo.rest.ElectricVehicleController(ElectricVehicleService) may expose internal representation by storing an externally mutable object into ElectricVehicleController.electricVehicleService	MALICIOUS_CODE	EI_EXPOSE_REP2	30	Medium

com.example.demo.rest.HybridVehicleController

Bug	Category	Details	Line	Priority
new com.example.demo.rest.HybridVehicleController(HybridVehicleService) may expose internal representation by storing an externally mutable object into HybridVehicleController.hybridVehicleService	MALICIOUS_CODE	EI_EXPOSE_REP2	30	Medium

com.example.demo.service.CombustionVehicleServiceImpl

Bug	Category	Details	Line	Priority
Comparison of String parameter using == or != in com.example.demo.service.CombustionVehicleServiceImpl.findByColour(String)	BAD_PRACTICE	ES_COMPARING_PARAMETER_STRING_WITH_EQ	114	High
Comparison of String parameter using == or != in com.example.demo.service.CombustionVehicleServiceImpl.findByName(String)	BAD_PRACTICE	ES_COMPARING_PARAMETER_STRING_WITH_EQ	106	High
Suspicious comparison of Long references in com.example.demo.service.CombustionVehicleServiceImpl.findById(Long)	CORRECTNESS	RC_REF_COMPARISON	98	High
Suspicious comparison of Integer references in com.example.demo.service.CombustionVehicleServiceImpl.findByNumDoors(Integer)	CORRECTNESS	RC_REF_COMPARISON	122	High
Suspicious comparison of Boolean references in com.example.demo.service.CombustionVehicleServiceImpl.findByOnOff(Boolean)	BAD_PRACTICE	RC_REF_COMPARISON_BAD_PRACTICE_BOOLEAN	146	Medium

com.example.demo.service.ElectricVehicleServiceImpl

Bug	Category	Details	Line	Priority
Comparison of String parameter using == or != in com.example.demo.service.ElectricVehicleServiceImpl.findByColour(String)	BAD_PRACTICE	ES_COMPARING_PARAMETER_STRING_WITH_EQ	115	High
Comparison of String parameter using == or != in com.example.demo.service.ElectricVehicleServiceImpl.findByName(String)	BAD_PRACTICE	ES_COMPARING_PARAMETER_STRING_WITH_EQ	107	High
Suspicious comparison of Long references in com.example.demo.service.ElectricVehicleServiceImpl.findById(Long)	CORRECTNESS	RC_REF_COMPARISON	99	High
Suspicious comparison of Integer references in com.example.demo.service.ElectricVehicleServiceImpl.findByLoadingTime(Integer)	CORRECTNESS	RC_REF_COMPARISON	155	High
Suspicious comparison of Integer references in com.example.demo.service.ElectricVehicleServiceImpl.findByNumDoors(Integer)	CORRECTNESS	RC_REF_COMPARISON	123	High
Suspicious comparison of Boolean references in com.example.demo.service.ElectricVehicleServiceImpl.findByOnOff(Boolean)	BAD_PRACTICE	RC_REF_COMPARISON_BAD_PRACTICE_BOOLEAN	147	Medium

com.example.demo.service.HybridVehicleServiceImpl

Bug	Category	Details	Line	Priority
Comparison of String parameter using == or != in com.example.demo.service.HybridVehicleServiceImpl.findByColour(String)	BAD_PRACTICE	ES_COMPARING_PARAMETER_STRING_WITH_EQ	113	High
Comparison of String parameter using == or != in com.example.demo.service.HybridVehicleServiceImpl.findByName(String)	BAD_PRACTICE	ES_COMPARING_PARAMETER_STRING_WITH_EQ	105	High
Suspicious comparison of Long references in com.example.demo.service.HybridVehicleServiceImpl.findById(Long)	CORRECTNESS	RC_REF_COMPARISON	97	High
Suspicious comparison of Integer references in com.example.demo.service.HybridVehicleServiceImpl.findByNumDoors(Integer)	CORRECTNESS	RC_REF_COMPARISON	121	High
Suspicious comparison of Boolean references in com.example.demo.service.HybridVehicleServiceImpl.findByOnOff(Boolean)	BAD_PRACTICE	RC_REF_COMPARISON_BAD_PRACTICE_BOOLEAN	145	Medium

Jenkins

1. Se crea un contenedor de Docker de Jenkins, con los siguientes comandos:

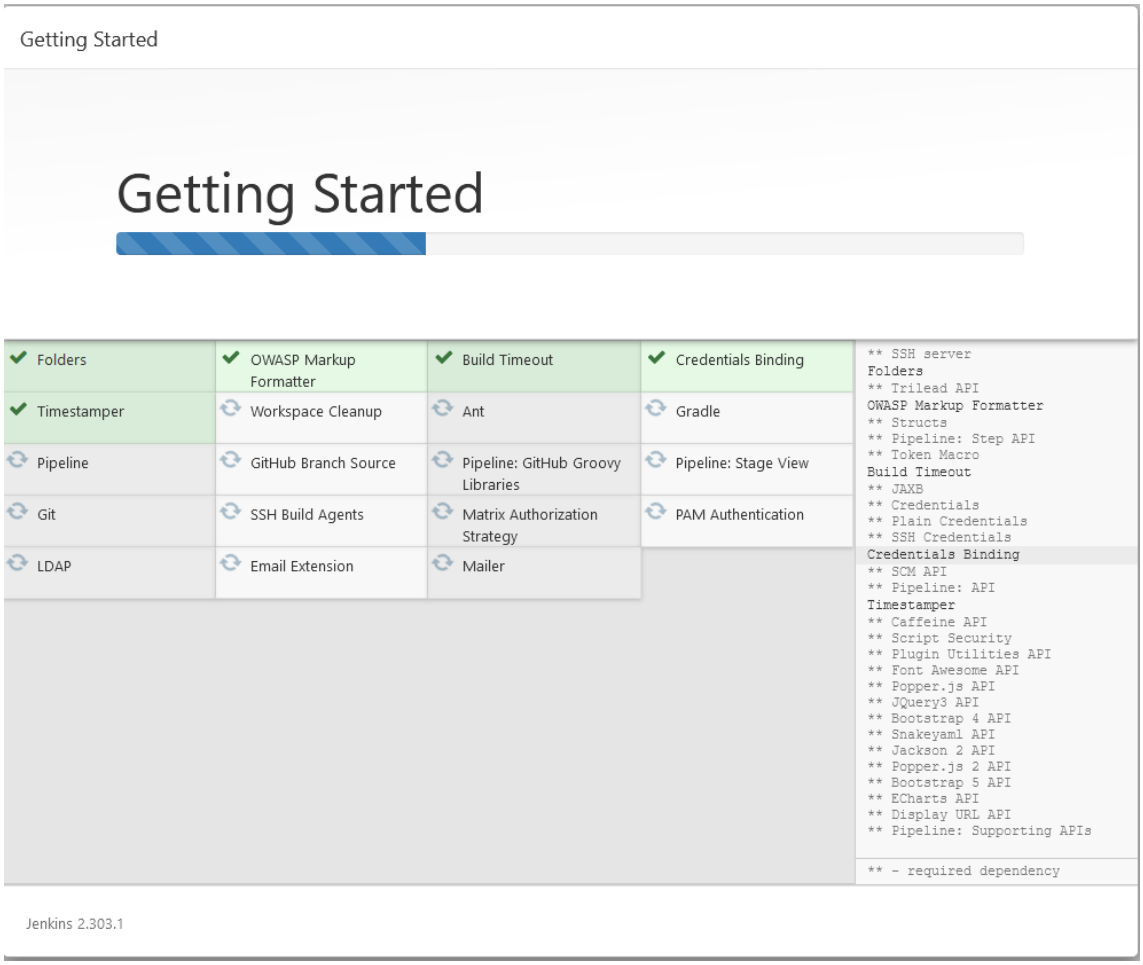
```
docker run --name jenkins -p 8080:8080 -d jenkins/jenkins:lts
```

```
docker logs -f jenkins
```

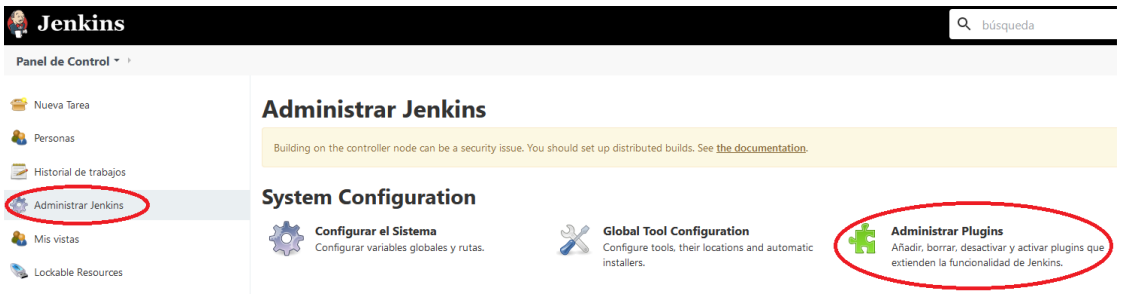
2. Se obtiene la contraseña con el siguiente comando:

```
docker exec -it jenkins cat
/var/jenkins_home/secrets/initialAdminPassword
```

3. En localhost:8080, se introduce la contraseña anterior y se instala lo que viene por defecto. Después se crea una cuenta de admin y se configura la instancia con el localhost escogido.

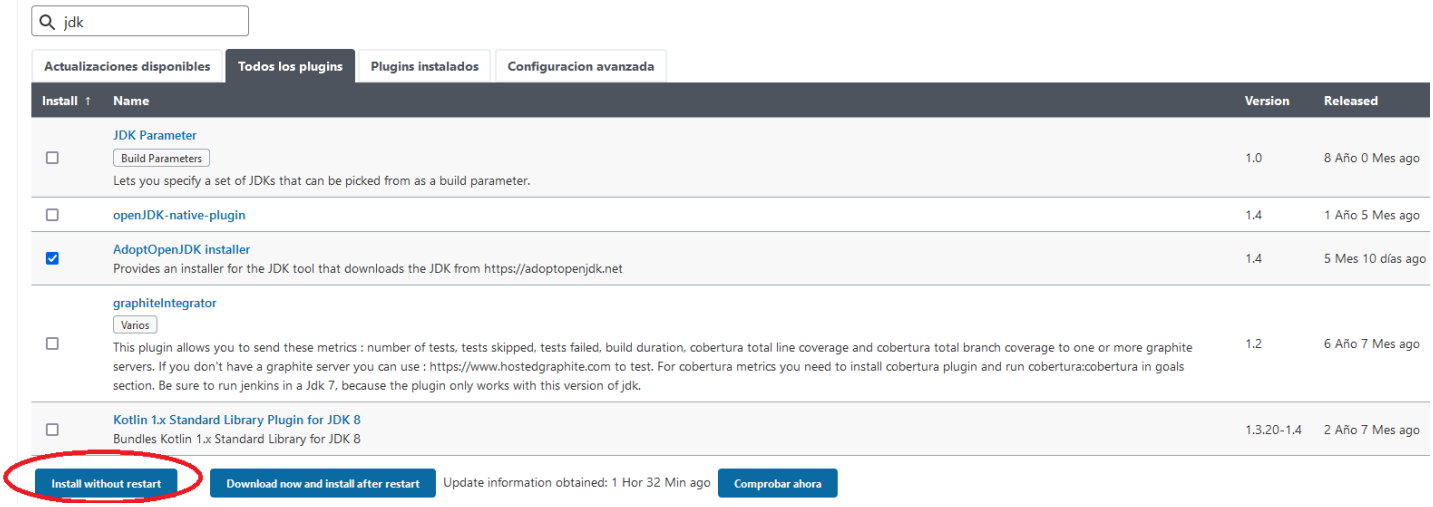


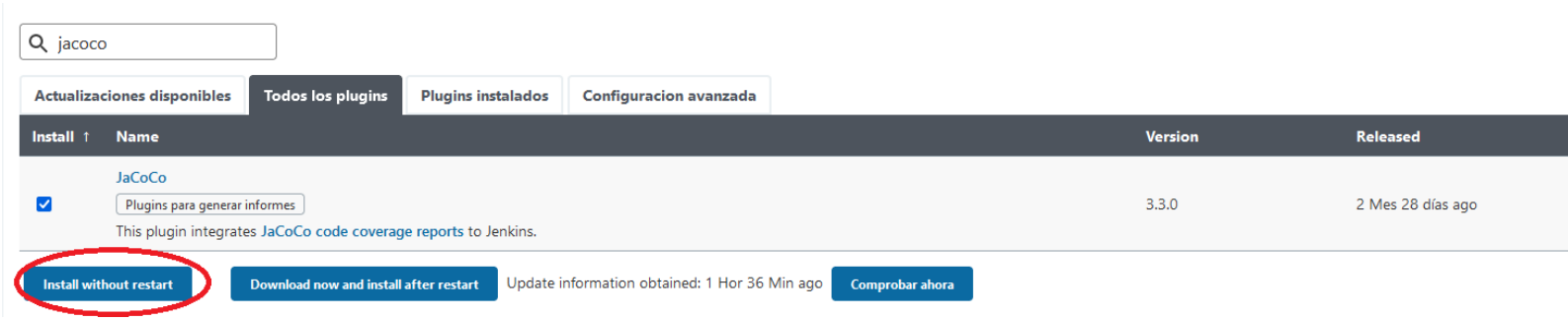
4. En administrar Jenkins, se pincha en administrar plugins:



Se busca en todos los plugins, los siguientes plugins:

- AdoptOpenJDK installer: escribimos jdk en la búsqueda.
- JaCoCo





5. Se reinicia Jenkins, con el siguiente comando en Ubuntu:

   Para comprobar la lista de imágenes: `docker ps` y se pulsa enter.

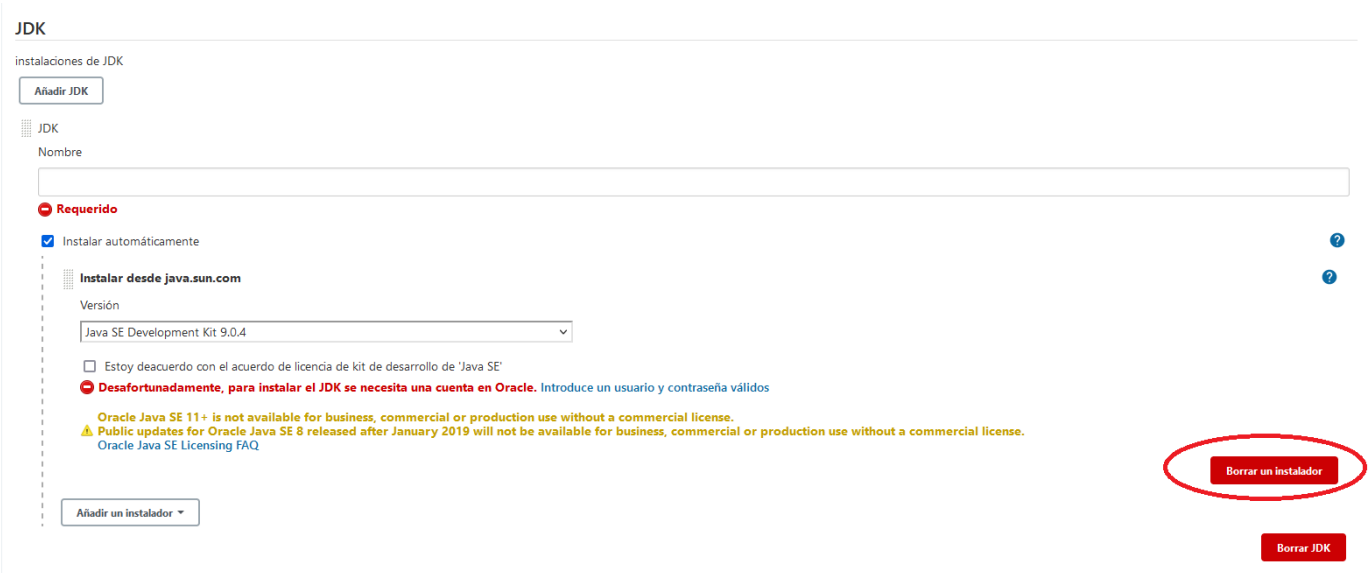
   `docker restart Jenkins`

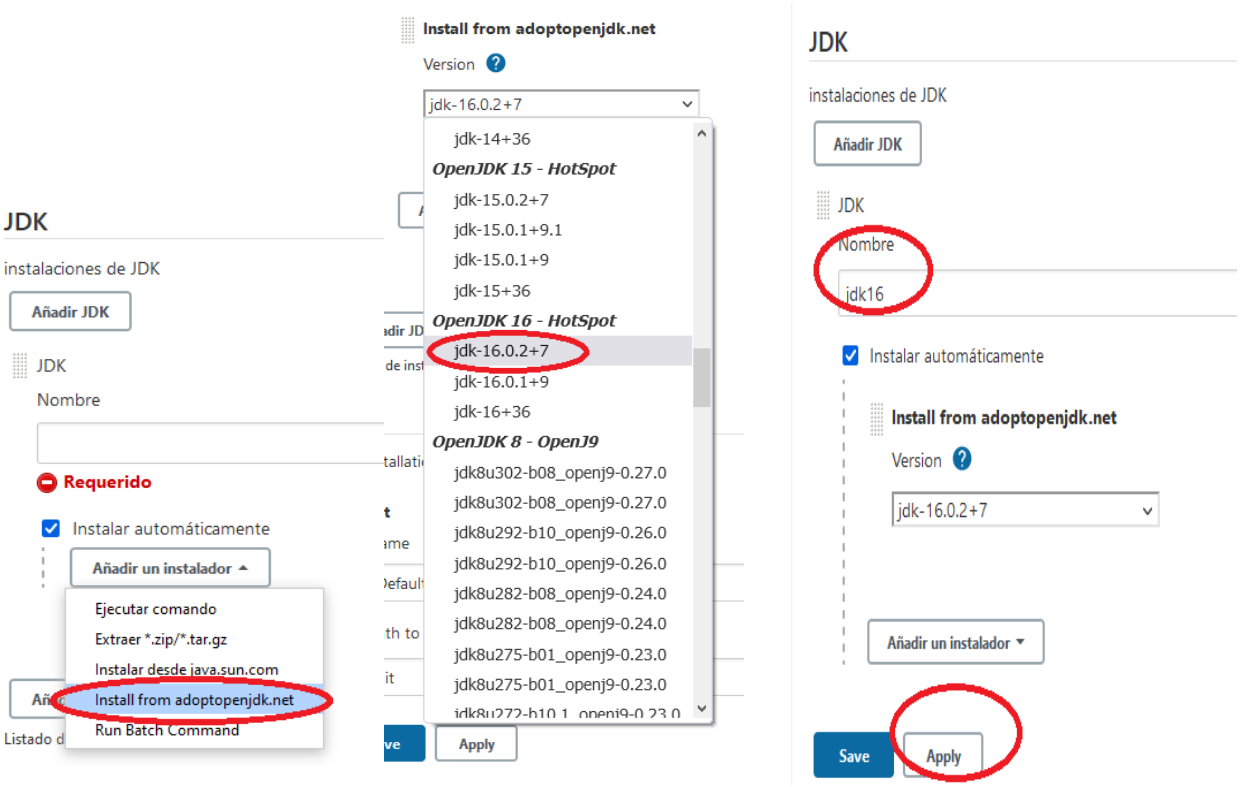
```
marinamadrid@DESKTOP-HP061D8:~$ docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED      STATUS       PORTS
                  NAMES
8ea66229c2e4   jenkins/jenkins:lts   "/sbin/tini -- /usr/…"   4 days ago   Up 2 hours   0.0.0.0:8080->8080/tcp, :::8080-
>8080/tcp, 50000/tcp   jenkins
marinamadrid@DESKTOP-HP061D8:~$ docker restart jenkins
```

Y volvemos a iniciar sesión en Jenkins.

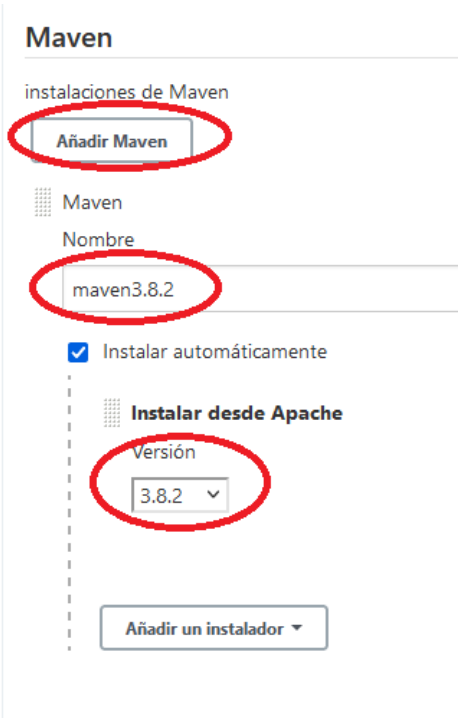
6. En administrar Jenkins, se pincha en Global Tool Configuration, y se configura Maven y JDK, siguiendo los siguientes pasos:

- Configuración de JDK





- Configuración de Maven

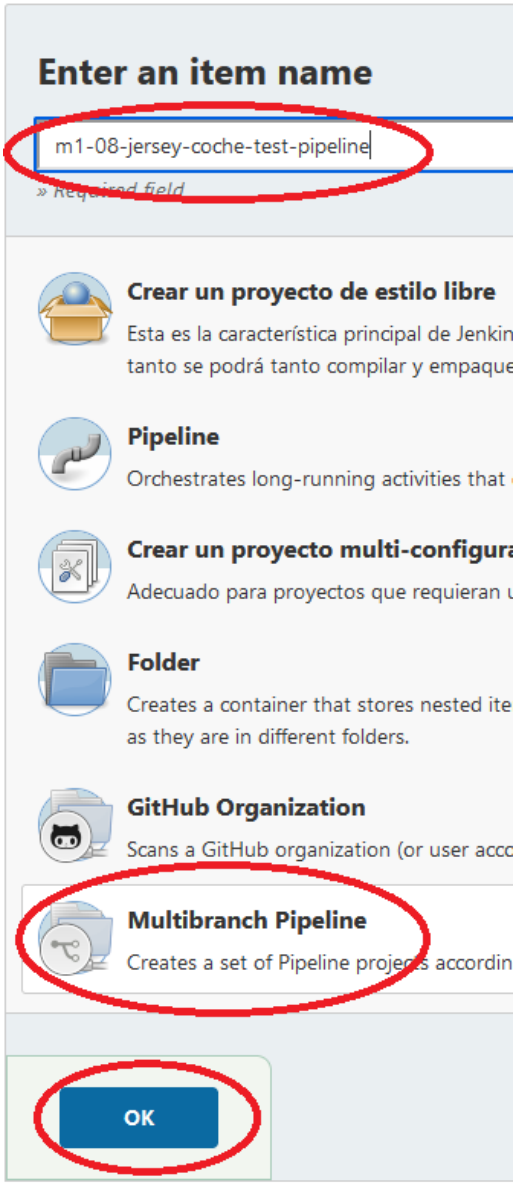


Y para finalizar se da a Save.

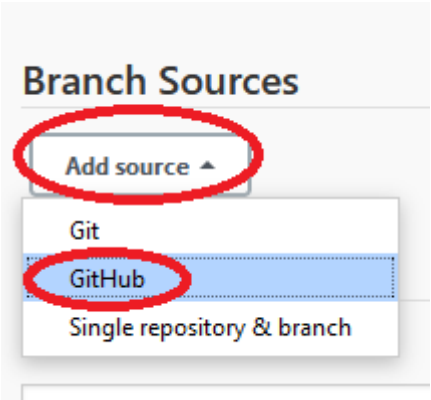
7. Se crea el archivo de Jenkinsfile, para ello se clica con el botón derecho del ratón sobre la carpeta del proyecto en Eclipse, y se da New/File, y se le llama Jenkinsfile. Y se añade el siguiente texto:

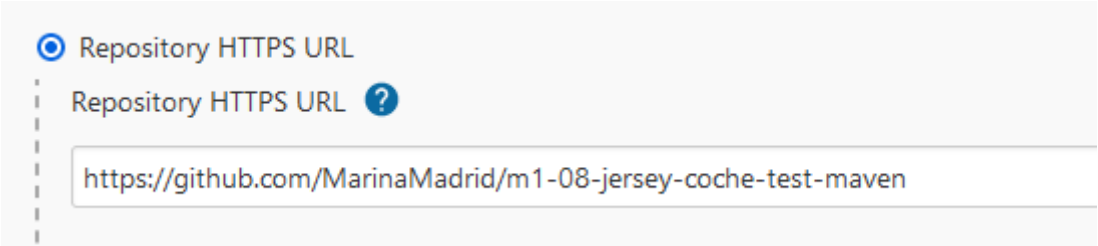
```
pipeline {
    agent any
    tools {
        maven "maven3.8.2"
        jdk "jdk16"
    }
    stages {
        stage("Env Variables") {
            steps {
                sh "printenv"
            }
        }
        stage('Build') {
            steps {
                sh 'mvn -B -DskipTests clean package'
            }
        }
        stage('Test') {
            steps {
                sh 'mvn test'
            }
            post {
                always {
                    junit 'target/surefire-reports/*.xml'
                    archiveArtifacts 'target/*.jar'
                }
            }
        }
    }
}
```

8. Se crea el pipeline, dando a Nueva tarea, y luego Multibranch pipeline.



Y luego se configura la parte de git de la siguiente forma:





9. Se escanea y construye automáticamente.

Scan Repository Log

```
Started
[Tue Sep 14 20:21:46 UTC 2021] Starting branch indexing...
20:21:48 Connecting to https://api.github.com with no credentials, anonymous access
Examining MarinaMadrid/m1-08-jersey-coche-test-maven

  Checking branches...

  Getting remote branches...

    Checking branch master

  Getting remote pull requests...
      'Jenkinsfile' found
    Met criteria
Scheduled build for branch: master

  1 branches were processed

  Checking pull-requests...

  0 pull requests were processed

Finished examining MarinaMadrid/m1-08-jersey-coche-test-maven

[Tue Sep 14 20:21:49 UTC 2021] Finished branch indexing. Indexing took 2.6 sec
Finished: SUCCESS
```

m1-08-jersey-coche-test-pipeline

Disable Multibranch Pipeline

Branches (1) | Pull Requests (0)

S	W	Name ↓	Último Éxito	Último Fallo	Última Duración	
✓	☼	master	1 Min 58 Seg - #1	N/D	22 Seg	

Icono: S M L

Guía de iconos | Atom feed para todos | Atom feed para fallas | Atom feed para los más recientes

Up
Status
Changes
Construir ahora
Ver Configuración
Full Stage View
GitHub
Pipeline Syntax

Historia de tareas Tendencia

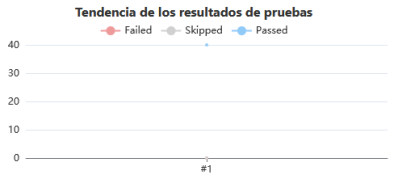
Atom feed Para Todos
Atom feed para los errores

Branch master

Full project name: m1-08-jersey-coche-test-pipeline/master

Last Successful Artifacts
m1-08-jersey-coche-0.0.1-SNAPSHOT-javadoc.jar 246.32 KB view
m1-08-jersey-coche-0.0.1-SNAPSHOT.jar 22.03 MB view

Recent Changes



Stage View

	Declarative: Checkout SCM	Declarative: Tool Install	Env Variables	Build	Test
Average stage times: (Average full run time: ~22s)	1s	221ms	547ms	10s	9s
#1 Sep 14 22:52 No Changes	1s	221ms	547ms	10s	9s

Últimos resultados de tests (Sin fallas)

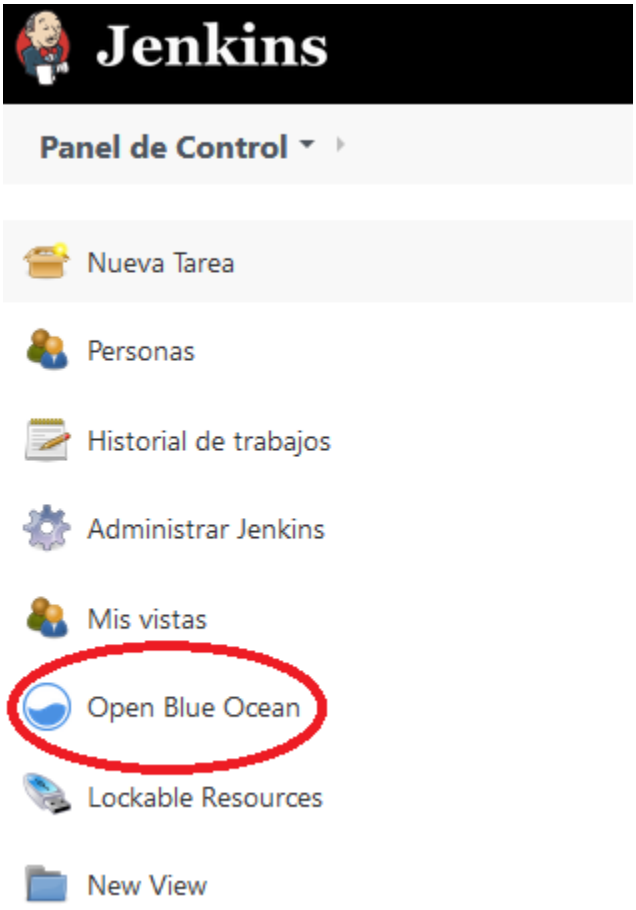
Enlaces permanentes

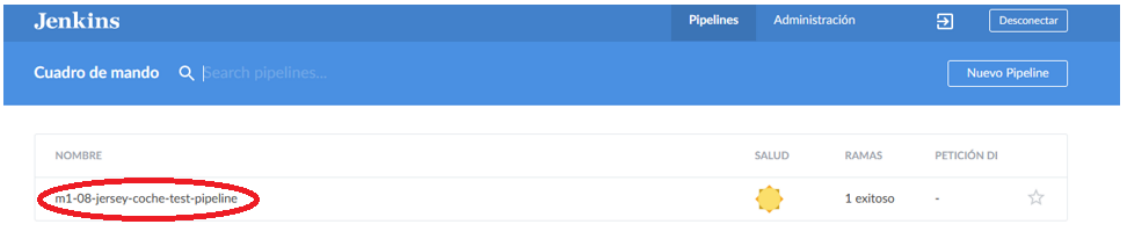
- "Última ejecución (#1) hace 2 Min 18 Seg"
- "Última ejecución estable (#1) hace 2 Min 18 Seg"
- "Última ejecución correcta (#1) hace 2 Min 18 Seg"
- "Last completed build (#1) hace 2 Min 18 Seg"

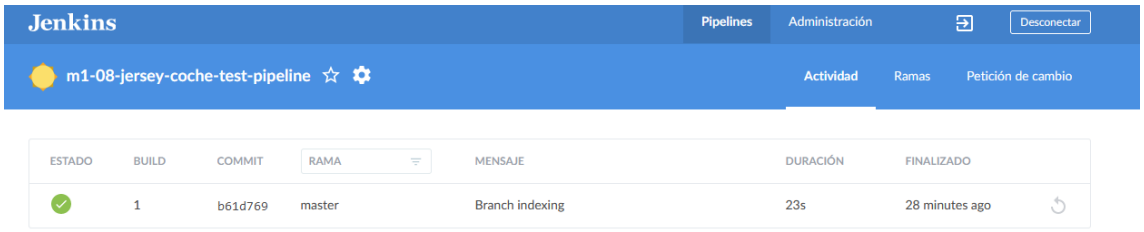
Blue Ocean

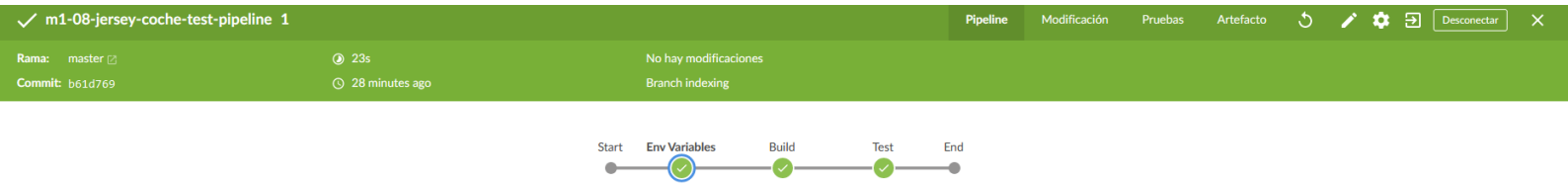
1. Se instala el plugin de Blue Ocean y luego se reinicia Jenkins.

2. Se pincha en Open Blue Ocean, que aparece en la barra lateral izquierda.









Env Variables - <1s

Restart Env Variables

✓	Check out from version control	<1s
✓	maven3.8.2 — Use a tool from a predefined Tool Installation	<1s
✓	Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.	<1s
✓	jdk16 — Use a tool from a predefined Tool Installation	<1s
✓	Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.	<1s
✓	maven3.8.2 — Use a tool from a predefined Tool Installation	<1s
✓	Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.	<1s
✓	jdk16 — Use a tool from a predefined Tool Installation	<1s
✓	Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.	<1s
✓	printenv — Shell Script	<1s

Build - 10s

Restart Build

✓	maven3.8.2 — Use a tool from a predefined Tool Installation
✓	Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.
✓	jdk16 — Use a tool from a predefined Tool Installation
✓	Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.
✓	mvn -B -DskipTests clean package — Shell Script

Test - 9s

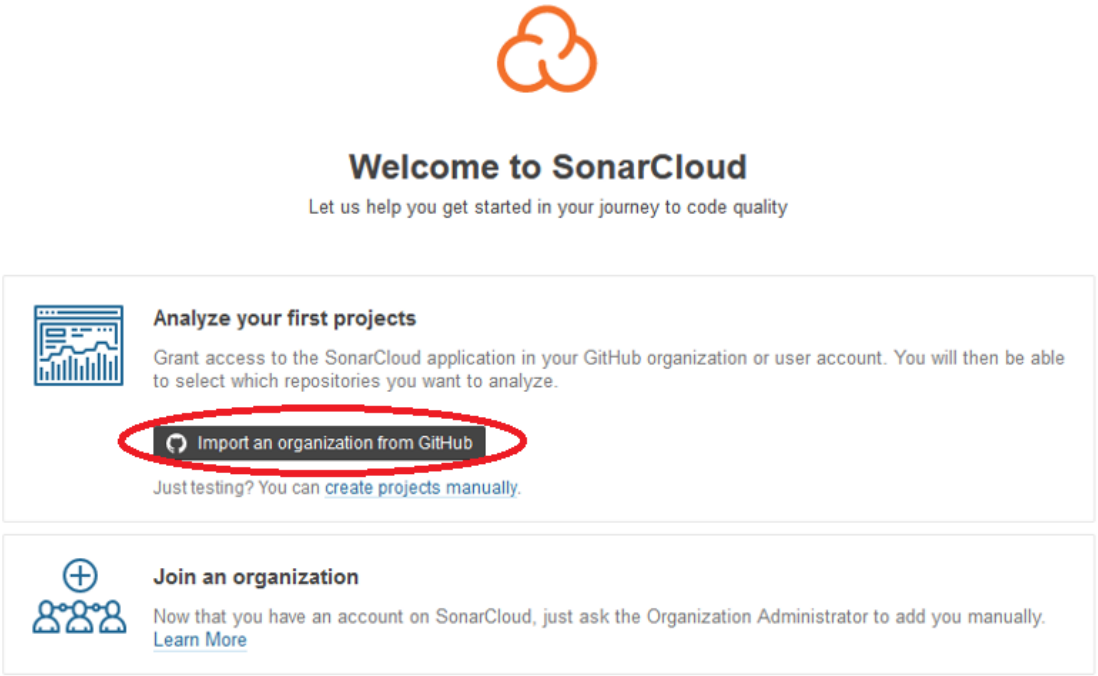
Restart Test

✓	maven3.8.2 — Use a tool from a predefined Tool Installation	<1s
✓	Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.	<1s
✓	jdk16 — Use a tool from a predefined Tool Installation	<1s
✓	Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.	<1s
✓	mvn test — Shell Script	9s
✓	target/surefire-reports/*.xml — Archive JUnit-formatted test results	<1s
✓	target/*.jar — Guardar los archivos generados	<1s

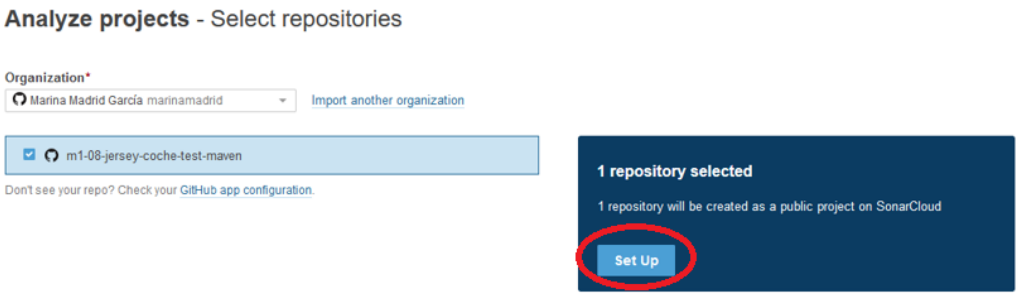
SonarCloud

1. Se entra en https://sonarcloud.io/welcome y inicia sesión con nuestra cuenta de Github (hay que darle permisos)

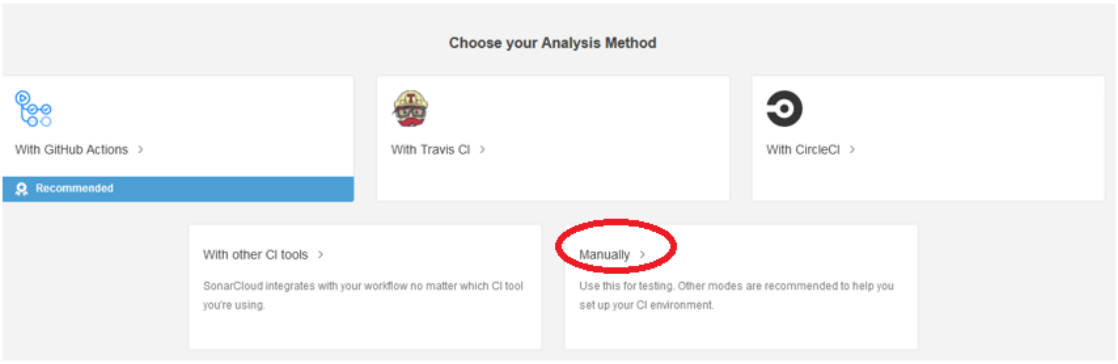
2. Se pincha en “Import an organization from GitHub”, se selecciona solamente el repositorio donde se está trabajando (o si se desea se puede importar todos los repositorios de Github) y se crea una organización eligiendo la key y el plan gratuito.



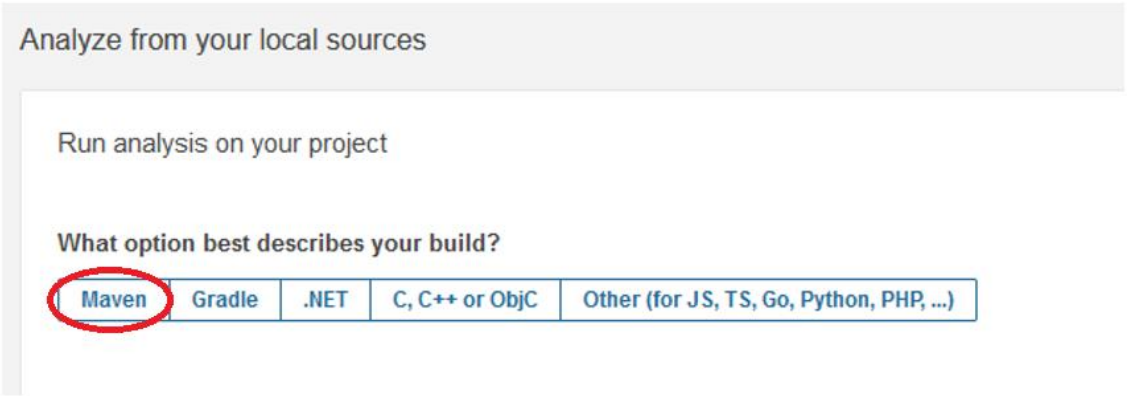
3. Se selecciona Set up.



4. Se pincha en “Manually”.



5. Y se elige la opción de Maven.



Execute the SonarScanner for Maven from your computer

Update your `pom.xml` file with the following properties:

```
<properties>
  <sonar.organization>marinamadrid</sonar.organization>
  <sonar.host.url>https://sonarcloud.io</sonar.host.url>
</properties>
```

Run the following command in the project folder:

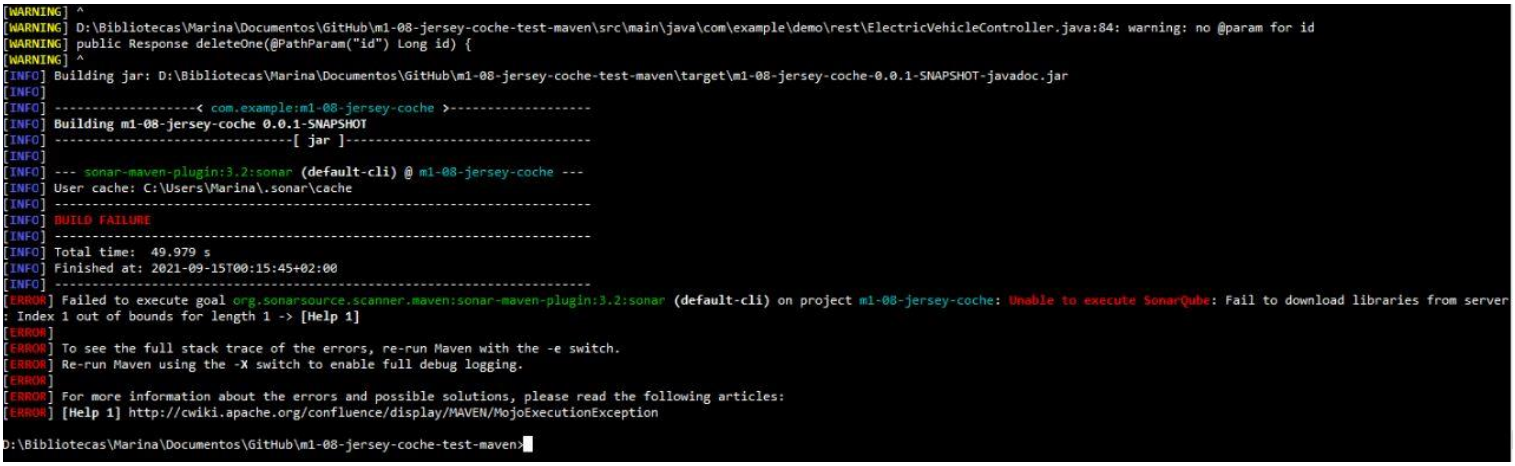
```
mvn verify org.sonarsource.scanner.maven:sonar-maven-plugin:sonar -Dsonar.projectKey=MarinaMadrid_m1-08-jersey-coche-test-maven
```

If you wish, you can shorten this command (to `mvn verify sonar:sonar`, for example) by specifying a prefix for the plugin.

See the SonarScanner for Maven documentation for more details.

6. Se coloca la propiedad anterior en el pom.xml, y se ejecuta el comando de arriba desde la terminal de dicho proyecto en Eclipse.

- Me ha aparecido el siguiente error:



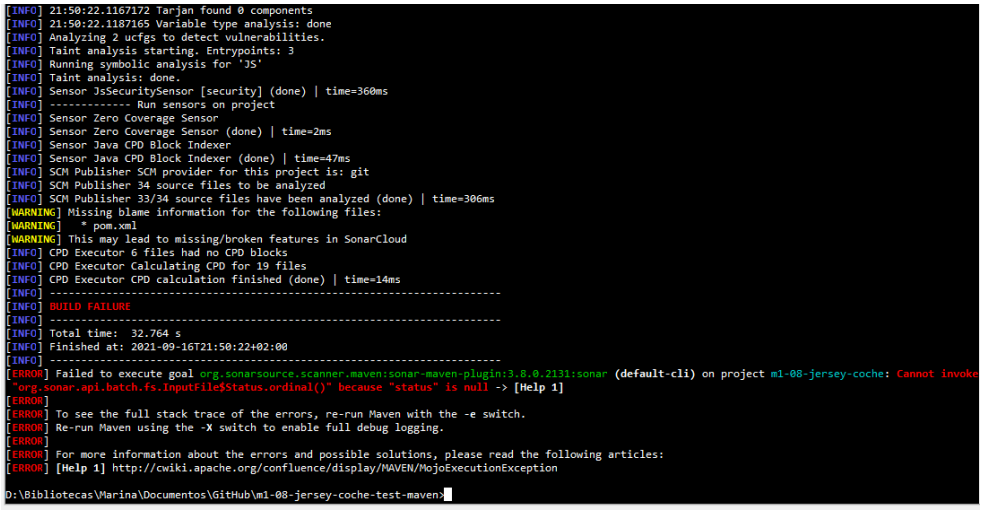
Parece que se ha arreglado añadiendo y actualizando las versiones de algunos plugins y dependencias en el pom.xml:

```
    <!-- https://mvnrepository.com/artifact/org.sonarsource.scanner.maven/sonar-maven-plugin -->
    <dependency>
        <groupId>org.sonarsource.scanner.maven</groupId>
        <artifactId>sonar-maven-plugin</artifactId>
        <version>3.8.0.2131</version>
    </dependency>
</dependencies>
```

```
<plugin>
    <groupId>org.sonarsource.scanner.maven</groupId>
    <artifactId>sonar-maven-plugin</artifactId>
    <version>3.8.0.2131</version>
</plugin>
```

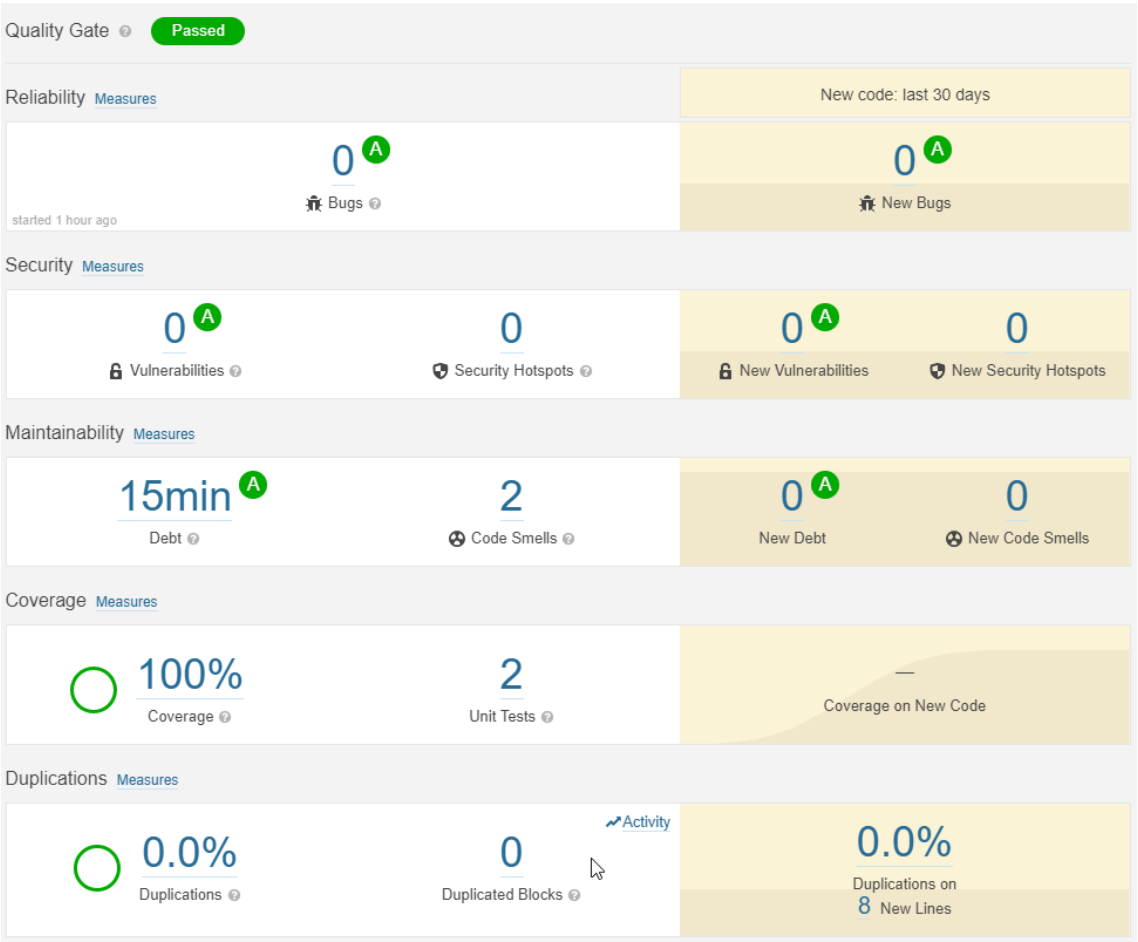
```
<plugin>
    <groupId>org.codehaus.mojo</groupId>
    <artifactId>sonar-maven-plugin</artifactId>
    <version>3.0.2</version>
</plugin>
```

- Ahora se está construyendo --- sonar-maven-plugin:3.8.0.2131:sonar, pero vuelve a salir otro error.



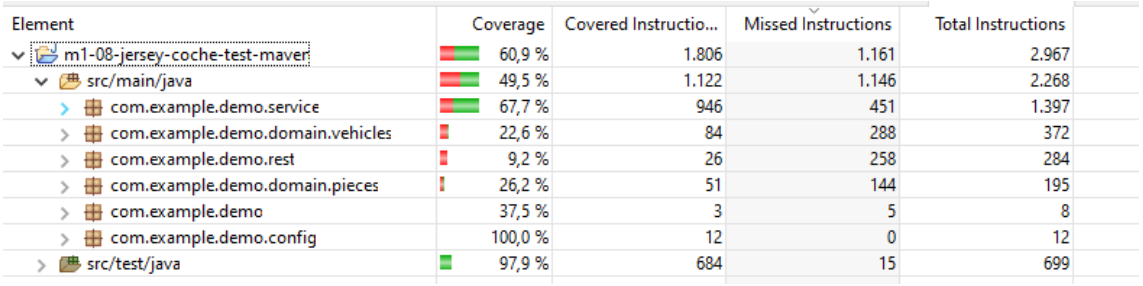
No se ha podido solucionar este problema.

7. Si se hubiese podido completar, se tendría que verificar que se analiza en sonarcloud.io y se obtendría un informe con la siguiente estructura, donde se puede observar los fallos, vulnerabilidades, la cobertura de los tests unitarios o hediondeces del código (code smells).



Corrección de errores

1. La corrección de errores la he realizado basándome en los resultados de SpotBugs y de la cobertura de Junit, ya que, SonarCloud no funciona. SpotBugs ha detectado 37 bugs, ningún error ni ningún “missing class”, y la cobertura de test es de 60,9%

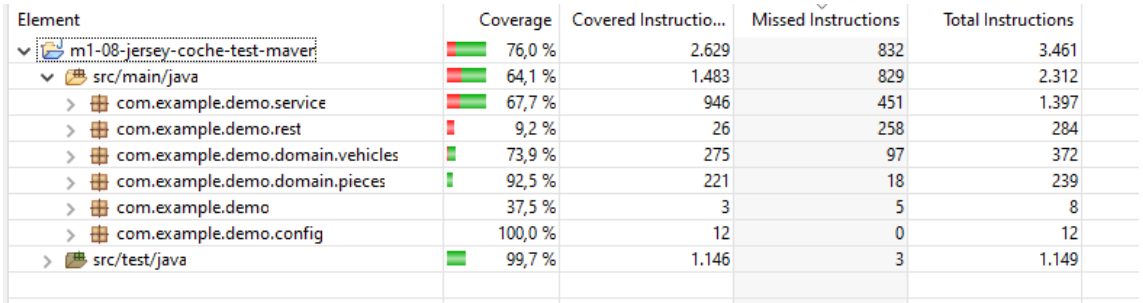


Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
m1-08-jersey-coche-test-maver	60,9 %	1.806	1.161	2.967
src/main/java	49,5 %	1.122	1.146	2.268
com.example.demo.service	67,7 %	946	451	1.397
com.example.demo.domain.vehicles	22,6 %	84	288	372
com.example.demo.rest	9,2 %	26	258	284
com.example.demo.domain.pieces	26,2 %	51	144	195
com.example.demo	37,5 %	3	5	8
com.example.demo.config	100,0 %	12	0	12
src/test/java	97,9 %	684	15	699

2. Los bugs detectados en SpotBugs son los siguientes, sean optado por dejarlos porque no son graves:

   - May expose internal representation by returning…
   - May expose internal representation by storing an externally mutable object into…
   - Se aconseja el uso de equals en lugar de == o !:
     - Comparison of String parameter using == or
     - Suspicious comparison of Boolean references.
     - Suspicious comparison of Long references.
     - Suspicious comparison of Integer references.

3. También he realizado nuevos tests unitarios en la parte de “domain”, tanto en “pieces” como en “vehicles”. Quedando la cobertura final de test en 76%.



Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
m1-08-jersey-coche-test-maver	76,0 %	2.629	832	3.461
src/main/java	64,1 %	1.483	829	2.312
com.example.demo.service	67,7 %	946	451	1.397
com.example.demo.rest	9,2 %	26	258	284
com.example.demo.domain.vehicles	73,9 %	275	97	372
com.example.demo.domain.pieces	92,5 %	221	18	239
com.example.demo	37,5 %	3	5	8
com.example.demo.config	100,0 %	12	0	12
src/test/java	99,7 %	1.146	3	1.149

4. Otros errores que son falsos positivos que pueden aparecer en SonarCloud son:

   - Los constructores superan el máximo de 7 parámetros, pero hay algunos constructores que requieren de más, como 8 o 9.
   - La eliminación de modificadores public de algunas clases, pero esto no puede realizarse porque javadoc necesita esos modificadores para crear correctamente su documentación.

- Cambiar el modificador de visibilidad de alguna clase abstracta a protected, no se ha realizado porque supondría refactorizar gran parte del proyecto, y se realizó así porque la primera implementación se hacían llamadas desde clases externas a esta clase, y es cierto que no es una buena práctica.

5. Una vez, solucionado los errores, se tendría que modificar el archivo Jenkinsfile para incluir Sonar, quedando el documento así:

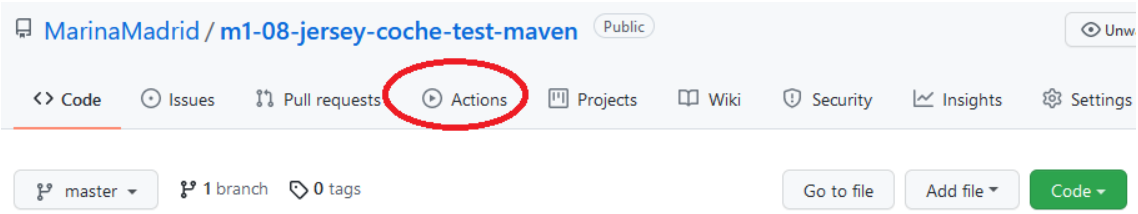
```
pipeline {
  agent any
  tools {
    maven "maven3.8.1"
    jdk "jdk16"
  }
  stages {
    stage("Env Variables") {
      steps {
        sh "printenv"
      }
    }
    stage('Build') {
      steps {
        sh 'mvn -B -DskipTests clean package'
      }
    }
    stage('Test') {
      steps {
        sh 'mvn test'
      }
      post {
        always {
          junit 'target/surefire-reports/*.xml'
          archiveArtifacts 'target/*.jar'
        }
      }
    }
    stage('Site') {
      steps {
        sh 'mvn site'
      }
    }
    stage('Sonar') {
      steps {
        sh 'mvn verify sonar:sonar -Dsonar.projectKey= MarinaMadrid_m1-08-jersey-coche-test-maven -Dsonar.organization= marinamadrid
 -Dsonar.host.url=https://sonarcloud.io -Dsonar.login= 2aded93a4e947401040d0a36433ae6425427685f -Dsonar.branch.name=master'
      }
```

```
        }
    }
}
```

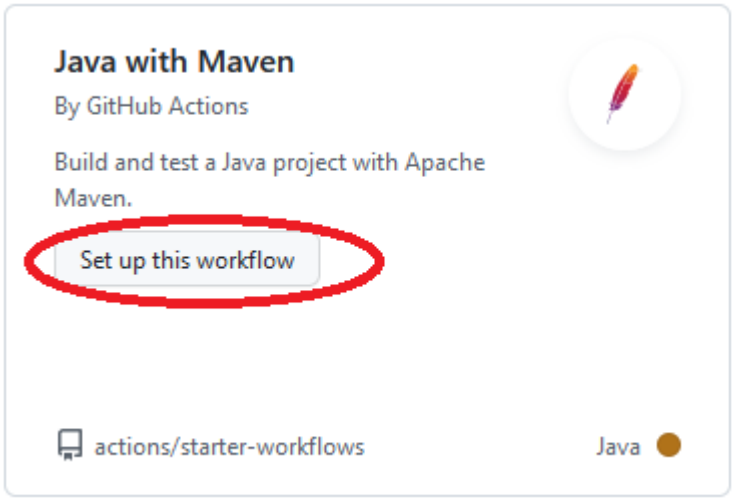
6. Y luego, se hubiese vuelto a ejecutar tanto Jenkins como SonarCloud. Y se hubiese comparado los nuevos informes y cobertura con los de antes. En mi caso, se ha vuelto a calcular la cobertura con Junit, y ha pasado de 60,9% a 76%.

Github actions

1. Se abre el repositorio donde está el proyecto en GitHub, y se pincha en Actions.



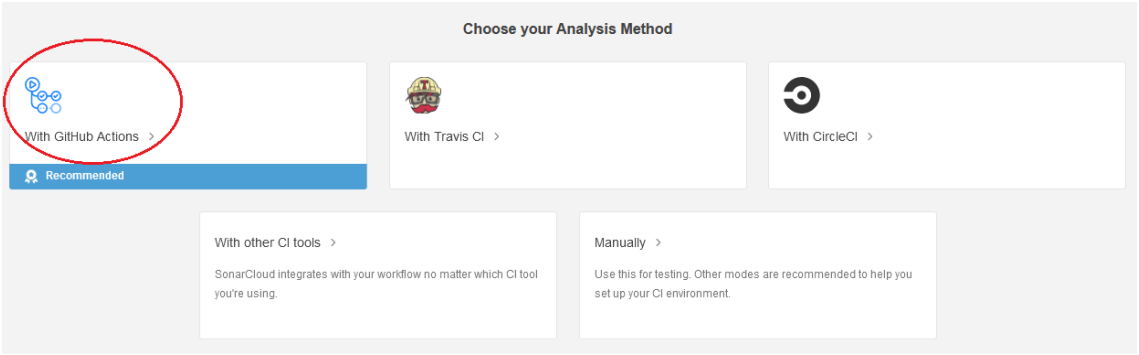
2. Y luego se busca la opción de Java with Maven, y se pincha en “Set up this workflow”

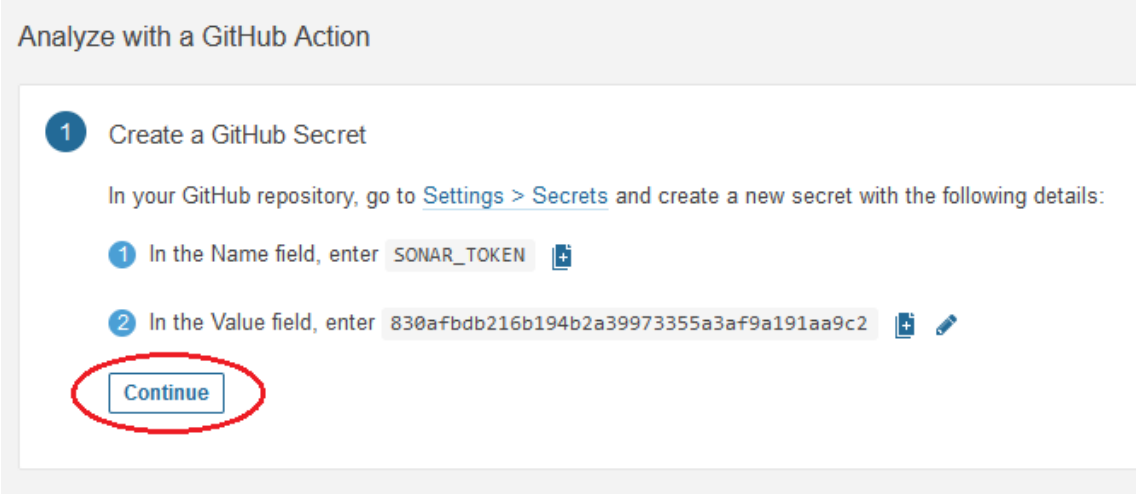


3. Se ajusta a la versión 16 de Java, y se da a Start commit, generándose el archivo maven.yml en la carpeta workflows.

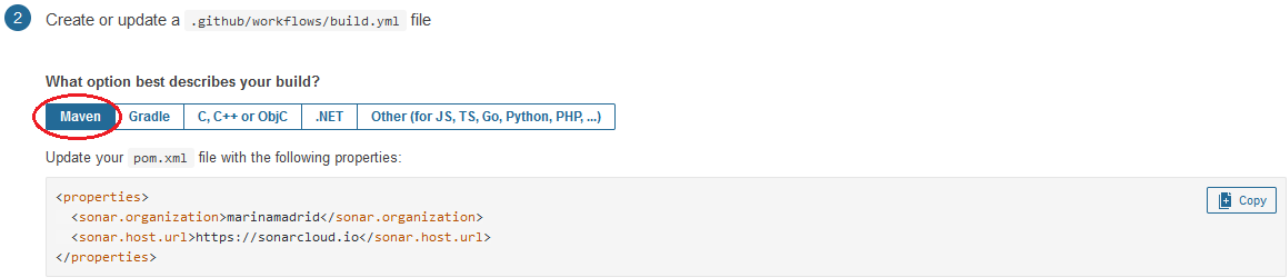
```yaml
name: Java CI with Maven

on:
  push:
    branches: [ master ]
  pull_request:
    branches: [ master ]

jobs:
  build:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v2
    - name: Set up JDK 16
      uses: actions/setup-java@v2
      with:
        java-version: '16'
        distribution: 'adopt'
        cache: maven
    - name: Build with Maven
      run: mvn -B package --file pom.xml

```

4. En SonarCloud, se elige la opción de “GitHub Actions”, y luego a continuar.





5. Se elige Maven, se añade en el archivo pom.xml las propiedades de sonar.organization y sonar.host.url, se crea un archivo build.yml con la configuración de la foto de abajo, cambiándole solamente la versión de java a 16, y se pincha en Start commit.



Create or update your `.github/workflows/build.yml`

Here is a base configuration to run a SonarCloud analysis on your master branch and Pull Requests. If you already have some GitHub Actions, you might want to just add some of these new steps to an existing one.

Copy

```yaml
name: Build
on:
  push:
    branches:
      - master
  pull_request:
    types: [opened, synchronize, reopened]
jobs:
  build:
    name: Build
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
        with:
          fetch-depth: 0  # Shallow clones should be disabled for a better relevancy of analysis
      - name: Set up JDK 11
        uses: actions/setup-java@v1
        with:
          java-version: 11
      - name: Cache SonarCloud packages
        uses: actions/cache@v1
        with:
          path: ~/.sonar/cache
          key: ${{ runner.os }}-sonar
          restore-keys: ${{ runner.os }}-sonar
      - name: Cache Maven packages
        uses: actions/cache@v1
        with:
          path: ~/.m2
          key: ${{ runner.os }}-m2-${{ hashFiles('**/pom.xml') }}
          restore-keys: ${{ runner.os }}-m2
      - name: Build and analyze
        env:
          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}  # Needed to get PR information, if any
          SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
        run: mvn -B verify org.sonarsource.scanner.maven:sonar-maven-plugin:sonar -Dsonar.projectKey=MarinaMadrid_m1-08-jersey-coche-test-maven
```

```xml
<!-- Propiedades -->
<properties>
    <java.version>16</java.version>
    <maven.compiler.source>14</maven.compiler.source>
    <maven.compiler.target>14</maven.compiler.target>
    <sonar.projectKey>MarinaMadrid_m1-08-jersey-coche-test-maven</sonar.projectKey>
    <sonar.organization>marinamadrid</sonar.organization>
    <sonar.host.url>https://sonarcloud.io</sonar.host.url>
</properties>
```

m1-08-jersey-coche-test-maven / .github / workflows / build.yml in master

Cancel changes | Start commit

Edit file | Preview changes | Spaces | 2 | No wrap

```yaml
name: Build
on:
  push:
    branches:
      - master
  pull_request:
    types: [opened, synchronize, reopened]
jobs:
  build:
    name: Build
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
        with:
          fetch-depth: 0  # Shallow clones should be disabled for a better relevancy of analysis
      - name: Set up JDK 16
        uses: actions/setup-java@v1
        with:
          java-version: 16
      - name: Cache SonarCloud packages
        uses: actions/cache@v1
        with:
          path: ~/.sonar/cache
          key: ${{ runner.os }}-sonar
```

Marketplace | Documentation

Search Marketplace for Actions

Featured Actions

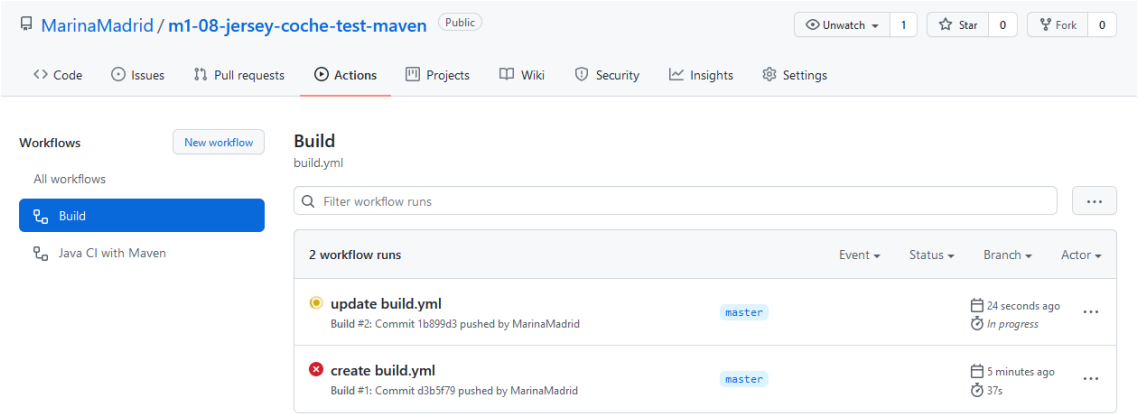
Upload a Build Artifact — By actions — 1.2k
Upload a build artifact that can be used by subsequent workflow steps

Setup Go environment — By actions — 531
Setup a Go environment and add it to the PATH

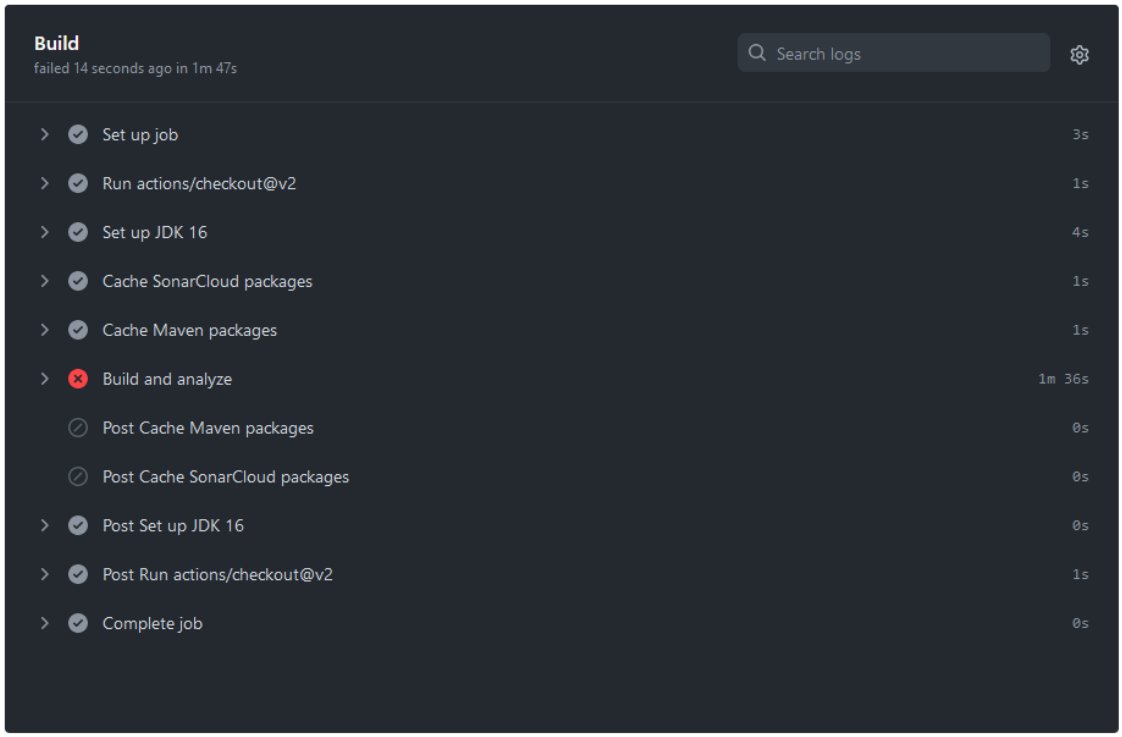
Setup Java JDK — By actions — 502
Set up a specific version of the Java JDK and add the command-line tools to the PATH

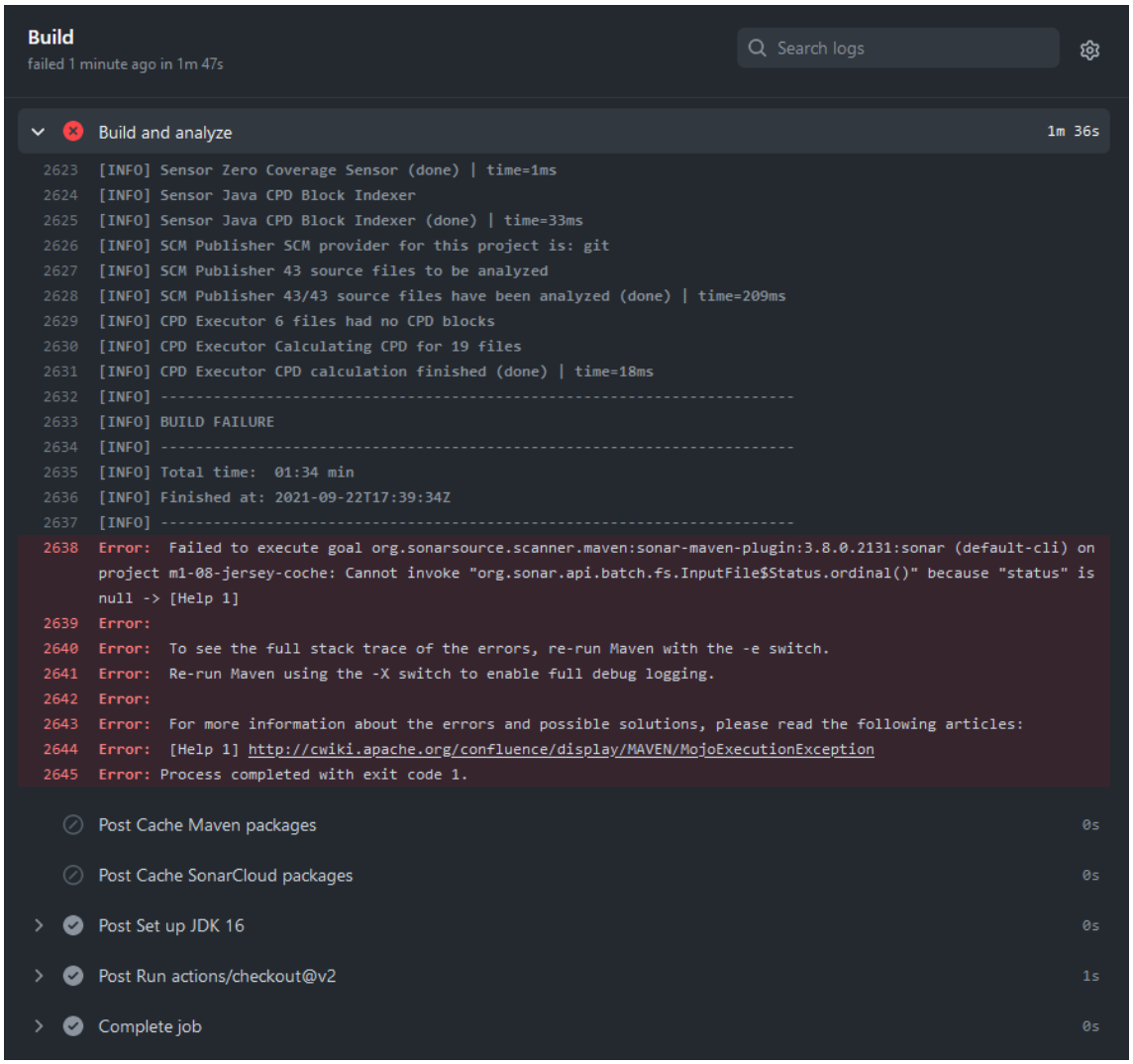
Close Stale Issues — By actions — 476

6. Y ya se puede ejecutar, en Actions y eligiendo el workflow creado.



7. Y como se esperaba no ha podido terminar, por el fallo que no se ha podido solucionar que aparece en SonarCloud.




Build
failed 1 minute ago in 1m 47s
Search logs
Build and analyze
1m 36s
2623 [INFO] Sensor Zero Coverage Sensor (done) | time=1ms
2624 [INFO] Sensor Java CPD Block Indexer
2625 [INFO] Sensor Java CPD Block Indexer (done) | time=33ms
2626 [INFO] SCM Publisher SCM provider for this project is: git
2627 [INFO] SCM Publisher 43 source files to be analyzed
2628 [INFO] SCM Publisher 43/43 source files have been analyzed (done) | time=209ms
2629 [INFO] CPD Executor 6 files had no CPD blocks
2630 [INFO] CPD Executor Calculating CPD for 19 files
2631 [INFO] CPD Executor CPD calculation finished (done) | time=18ms
2632 [INFO] --
2633 [INFO] BUILD FAILURE
2634 [INFO] --
2635 [INFO] Total time: 01:34 min
2636 [INFO] Finished at: 2021-09-22T17:39:34Z
2637 [INFO] --
2638 Error: Failed to execute goal org.sonarsource.scanner.maven:sonar-maven-plugin:3.8.0.2131:sonar (default-cli) on project m1-08-jersey-coche: Cannot invoke "org.sonar.api.batch.fs.InputFile$Status.ordinal()" because "status" is null -> [Help 1]
2639 Error:
2640 Error: To see the full stack trace of the errors, re-run Maven with the -e switch.
2641 Error: Re-run Maven using the -X switch to enable full debug logging.
2642 Error:
2643 Error: For more information about the errors and possible solutions, please read the following articles:
2644 Error: [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoExecutionException
2645 Error: Process completed with exit code 1.
Post Cache Maven packages
0s
Post Cache SonarCloud packages
0s
Post Set up JDK 16
0s
Post Run actions/checkout@v2
1s
Complete job
0s