

Material de Trabajo Sesión 03

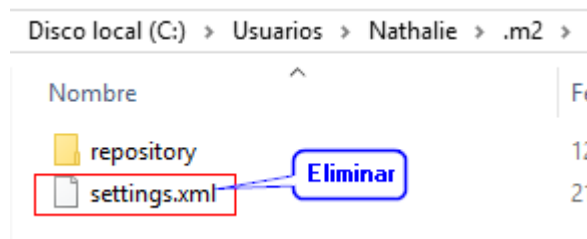
El material de trabajo de la sesión 03 contiene toda la información de la sesión 03 del curso “devops: integración y entrega continua”. Se recomienda seguir paso a paso lo indicado para tener éxito en la ejecución de las actividades indicadas.

Nota: Antes de continuar con las actividades se recomienda leer y ejecutar las instrucciones del documento “00.configuracion-entorno-trabajo.docx” para tener las herramientas configuradas y no tener inconveniente alguno.

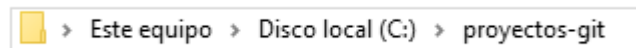
1. Configurar Proyecto “Facturación Web” en Eclipse

- 1.1. Eliminar el archivo “settings.xml” que se encuentra en la siguiente ruta
“C:\usuarios\\$miusuario\.m2\”

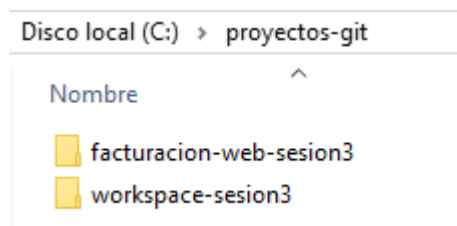
Nota: \$miusuario debe ser reemplazado por el usuario con el que iniciaron sesión en Windows.



- 1.2. Crear la carpeta “proyectos-git” en la unidad C de su PC.

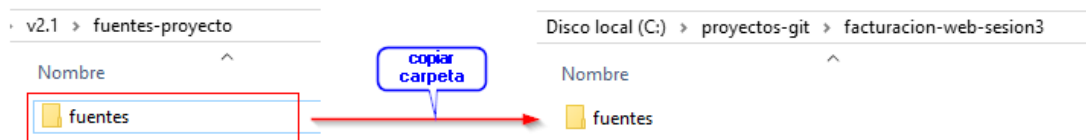


- 1.3. Crear la carpeta “facturacion-web-sesion3” y “workspace-sesion3” en la carpeta “proyectos-git”



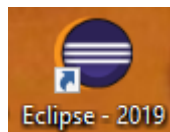
- 1.4. Copiar la carpeta “fuentes” que se encuentra ubicada en “v2.1 > fuentes-proyectos” y pegarla en la carpeta “C:\proyectos-git\facturacion-web-sesion3”

Nota: Se ha compartido un nuevo proyecto “fuentes.rar” que se encuentra en la misma ubicación donde se descargo este manual

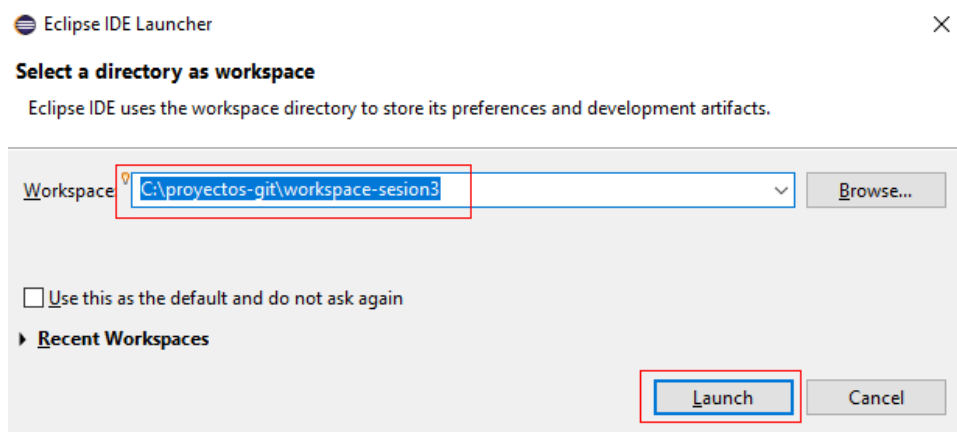


- 1.5. Abrir el “Eclipse” haciendo doble-clic en el icono “Eclipse – 2019” que se encuentra en el escritorio.

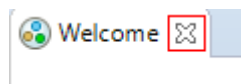
Nota: En la sección “[2.3. Instalación Eclipse](#)” del archivo “[00.configuracion-entorno-local.docx](#)” se indica como configurar e instalar el entorno de desarrollo “Eclipse”



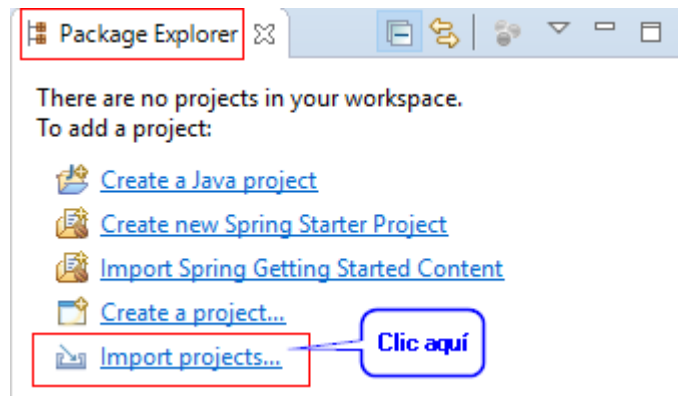
- 1.6. En la siguiente ventana debemos indicar la ruta del workspace “C:\proyectos-git\workspace-sesion3” y clic en el botón “Launch”



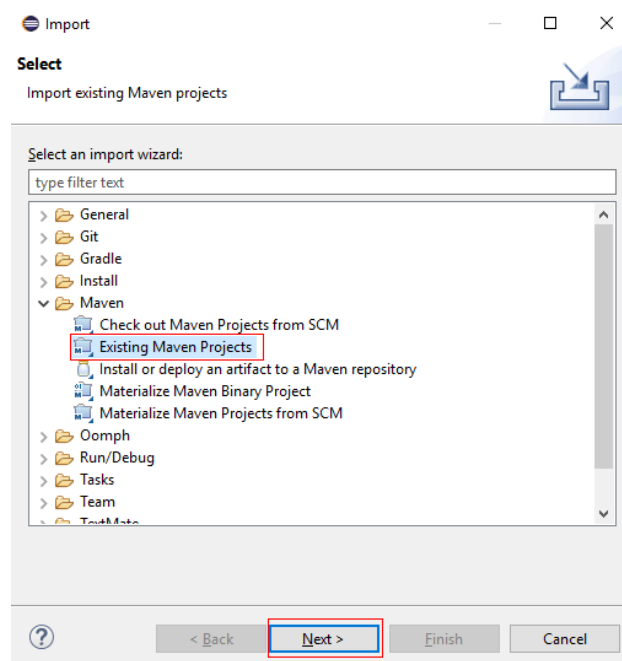
- 1.7. Cerrar la ventana “Welcome” haciendo clic en el botón “x”



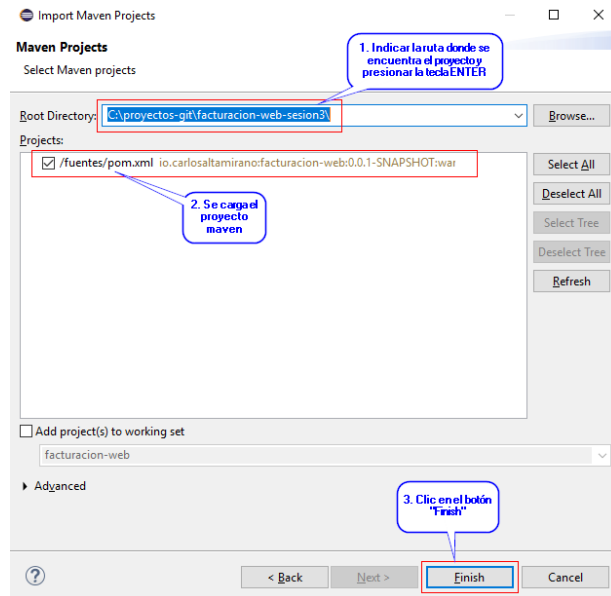
- 1.8. Clic en la opción “Import Project” del panel “Package Explorer” que se encuentra en el lado izquierda de la ventana.



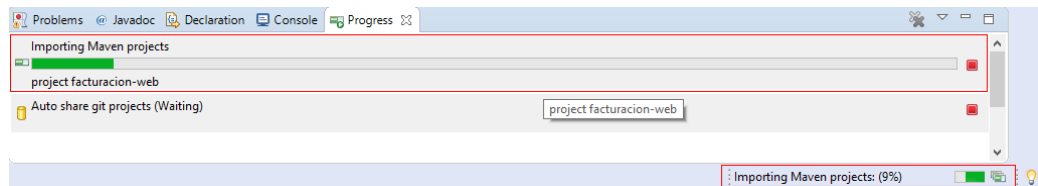
1.9. Seleccionar “Maven > Existing Maven Projects” y clic en el botón “Next >”



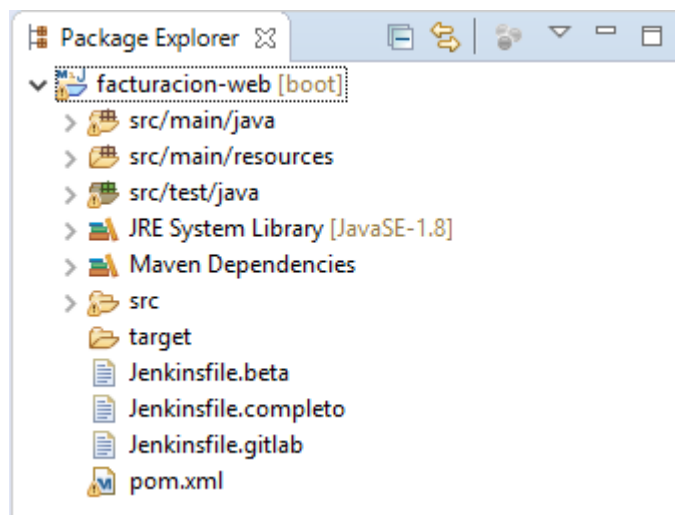
1.10. Copiar la ruta “C:\proyectos-git\facturacion-web-sesion3\” y presionar la tecla ENTER



- 1.11. En la esquina inferior derecha se mostrará el mensaje “Importing Maven projects” que significa que se está descargando las dependencias necesarias para que el proyecto cargue sin errores.

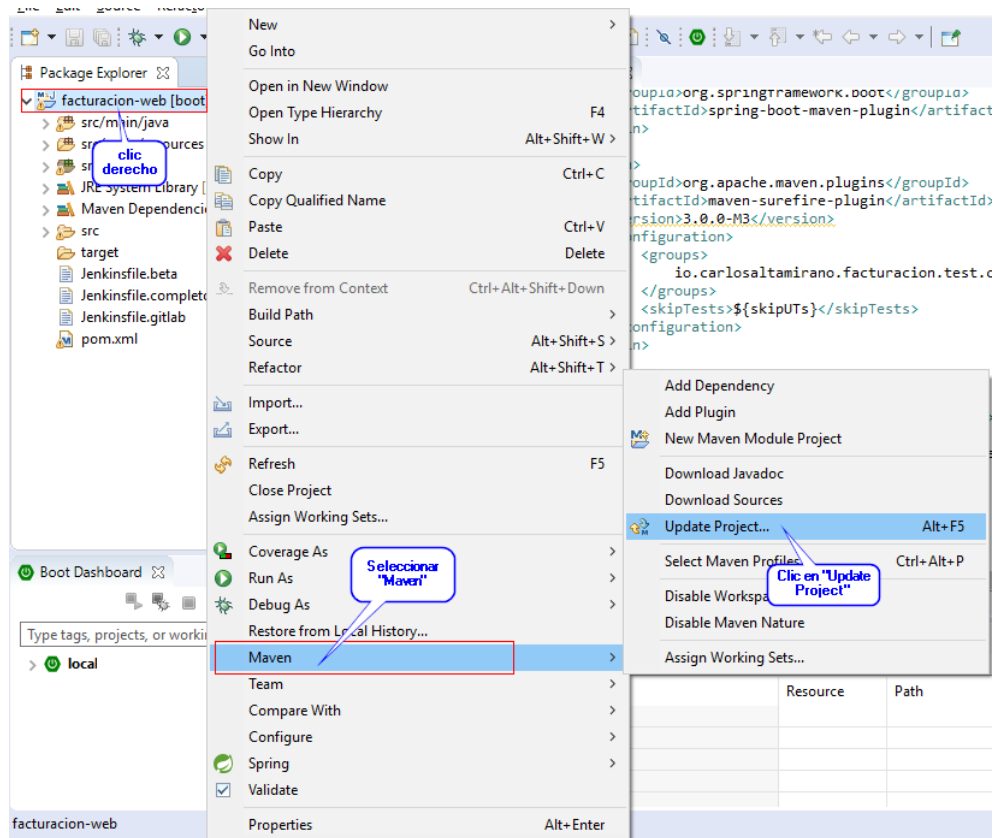


- 1.12. El proyecto empezará a cargar y se debe mostrar en el panel del “Package Explorer”

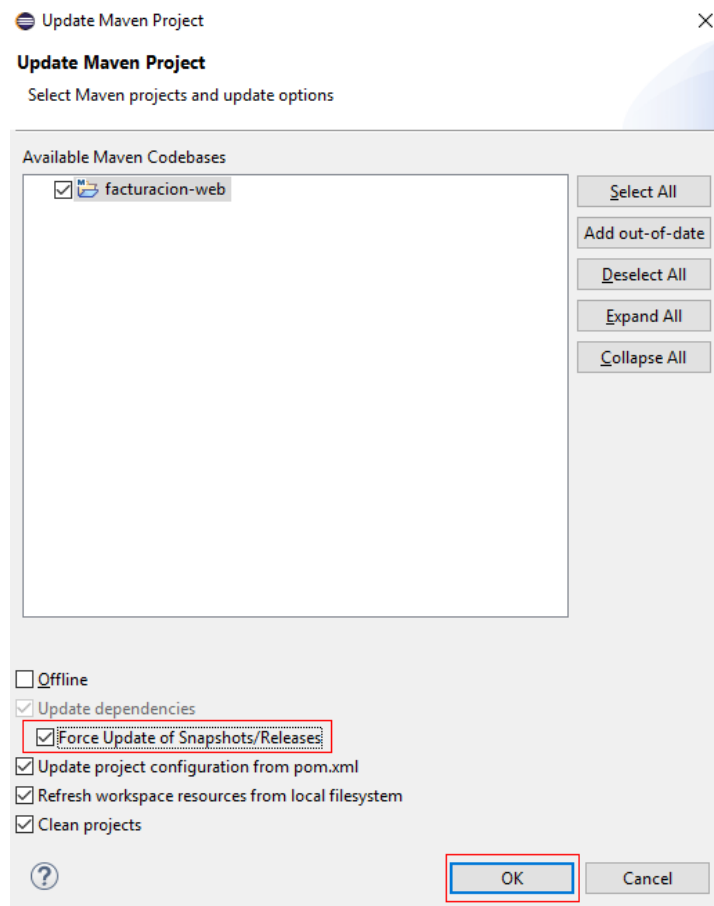


- 1.13. En caso ocurra algún problema al cargar el proyecto, realizar lo siguiente

- 1.13.1. Clic derecho en el proyecto “facturacion-web”, seleccionar “Maven” y hacer clic en “Update Project”



1.13.2. Marcar la opción “Force Update of Snapshots/Releases” y clic en el botón “OK” para descargar las dependencias.



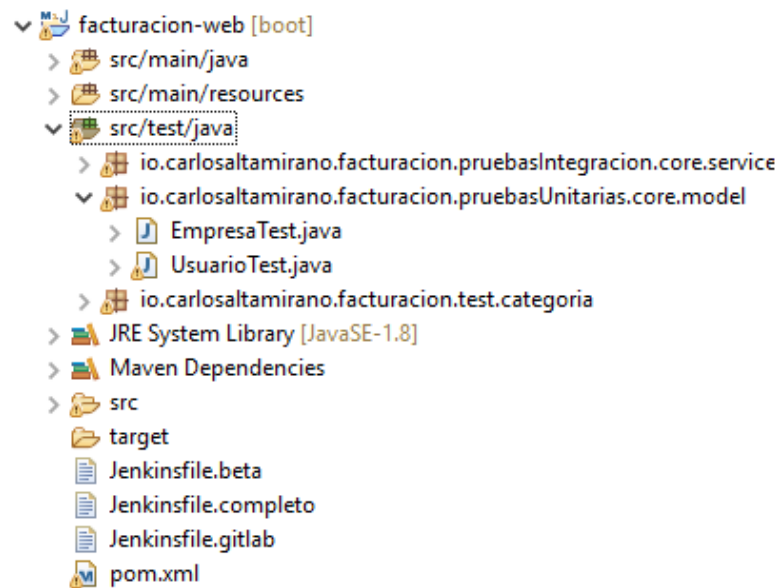
2. Configurar Pruebas Unitarias

2.1. Requisitos

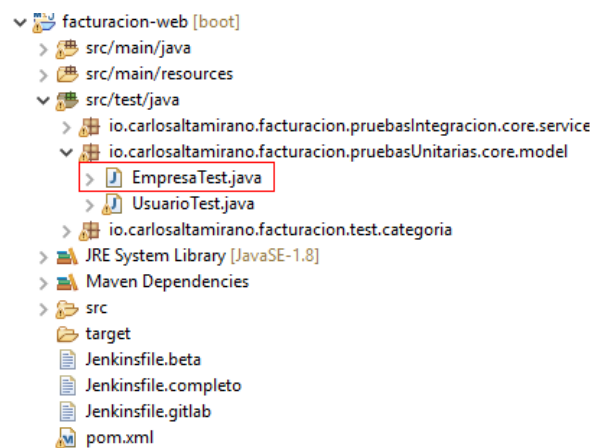
Ninguno

2.2. Ejecutar Pruebas Unitarias desde Eclipse

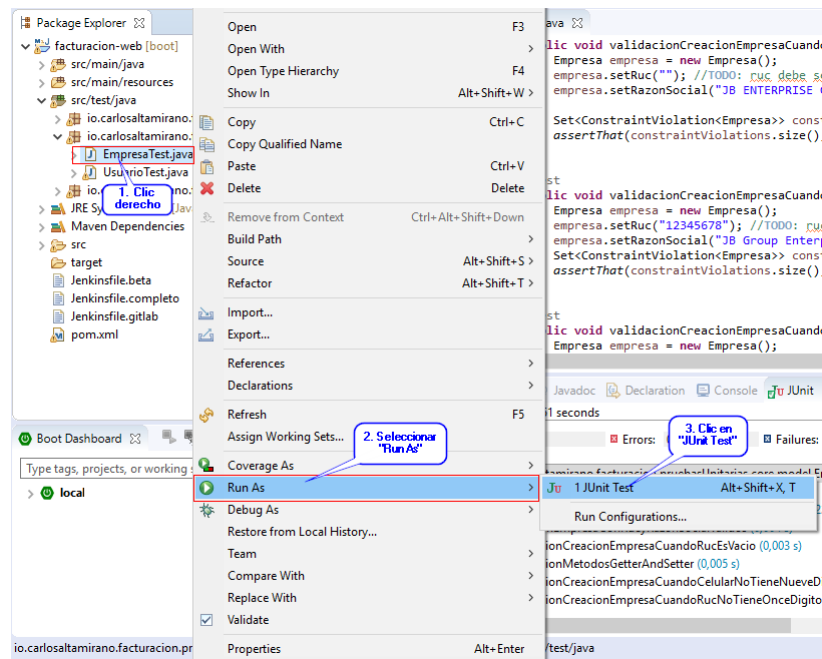
2.2.1. Expandir el proyecto *"facturacion-web"*, la carpeta *"src/test/java"* y el paquete *"io.carlosaltamirano.facturacion.pruebasUnitarias.core.model"*



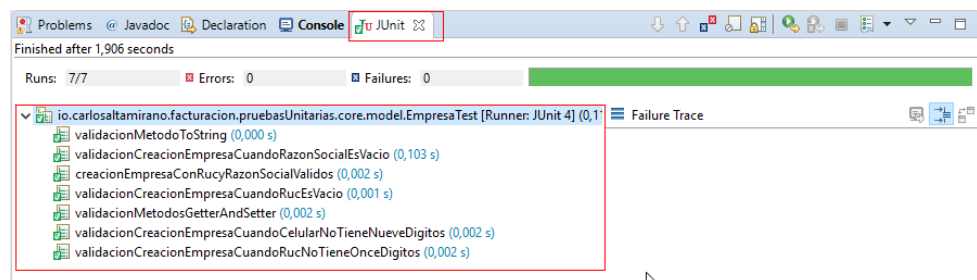
2.2.2. Seleccionar la clase “EmpresaTest”



2.2.3. Clic derecho, seleccionar la opción “Run As” y clic en “JUnit Test”



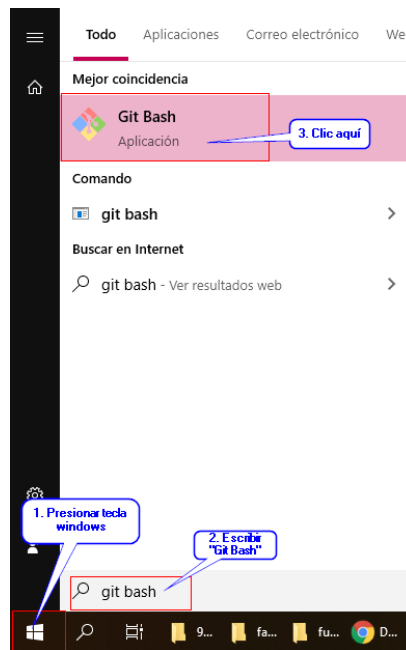
2.2.4. En la parte inferior del eclipse se abrirá el panel “JUnit” donde se mostrará el estado de la ejecución de las pruebas unitarias.



2.3. Ejecutar Pruebas Unitarias desde Línea de Comandos

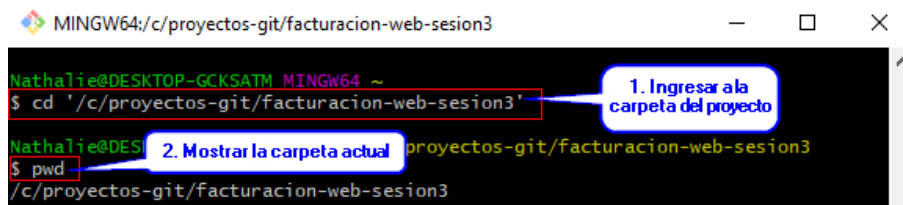
El proyecto “facturacion-web” fue creado usando “Maven”, eso quiere decir que podemos controlar el ciclo de vida del proyecto (compilar, test, test de integración, empaquetado y despliegue) a través de la línea de comandos.

2.3.1. Abrir el programa “Git Bash” (Presionar la tecla Windows, luego escribir “git bash”) y hacer clic en el programa.



2.3.2. Ir a la carpeta “C:\proyectos-git\facturacion-web-sesion3” usando el siguiente comando:

```
$ cd '/c/proyectos-git/facturacion-web-sesion3'
```



2.3.3. Para ejecutar las pruebas unitarias con maven se deberá ejecutar el siguiente comando:

```
$ mvn -f fuentes/pom.xml test
```

Donde:

- mvn: commando maven
- -f fuentes/pom.xml: ubicación donde se encuentra el archivo pom.xml
- test: goal para ejecutar pruebas unitarias

```
Nathalie@DESKTOP-GCKSATM MINGW64 /c/proyectos-git/facturacion-web-sesion3
$ mvn -f fuentes/pom.xml test
[INFO] Scanning for projects...
[INFO] -----< io.carlosaltamirano.facturacion-web >-----
[INFO] Building facturacion-web 0.0.1-SNAPSHOT
[INFO] -----[war]-----
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-war-plugin/3.1.0/maven-war-plugin-3.1.0.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-war-plugin/3.1.0/maven-war-plugin-3.1.0.jar
[INFO] --- jacoco-maven-plugin:0.8.4:prepare-agent (prepare-ut-agent) @ facturacion-web ---
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-jacoco-plugin/0.8.4/maven-jacoco-plugin-0.8.4.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-jacoco-plugin/0.8.4/maven-jacoco-plugin-0.8.4.jar
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ facturacion-web ---
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.454 s - in io.carlosaltamirano.facturacion.pruebasUnitarias.core.model.EmpresaTest
[INFO] Running io.carlosaltamirano.facturacion.pruebasUnitarias.core.model.UsuarioTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 s - in io.carlosaltamirano.facturacion.pruebasUnitarias.core.model.UsuarioTest
[INFO] Results:
[INFO] Tests run: 9, Failures: 0, Errors: 0, Skipped: 0
[INFO] --- jacoco-maven-plugin:0.8.4:report (report) @ facturacion-web ---
[INFO] Loading execution data file C:\proyectos-git\facturacion-web-sesion3\facturacion-web-sesion3\target\jacoco.exec
[INFO] Analyzed bundle 'facturacion-web' with 4 classes
[INFO] BUILD SUCCESS
[INFO] Total time: 01:11 min
[INFO] Finished at: 2019-10-28T02:56:15-05:00
[INFO] -----
```

1. Ejecutar comando maven

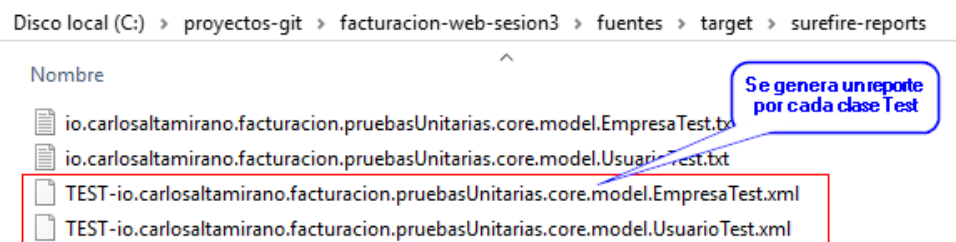
2. Automáticamente se empezará a descargar las dependencias necesarias para ejecutar las pruebas unitarias

2.3.4. Finalmente, se debe mostrar el resultado de la ejecución de los tests.

```
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.454 s - in io.carlosaltamirano.facturacion.pruebasUnitarias.core.model.EmpresaTest
[INFO] Running io.carlosaltamirano.facturacion.pruebasUnitarias.core.model.UsuarioTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 s - in io.carlosaltamirano.facturacion.pruebasUnitarias.core.model.UsuarioTest
[INFO] Results:
[INFO] Tests run: 9, Failures: 0, Errors: 0, Skipped: 0
[INFO] --- jacoco-maven-plugin:0.8.4:report (report) @ facturacion-web ---
[INFO] Loading execution data file C:\proyectos-git\facturacion-web-sesion3\facturacion-web-sesion3\target\jacoco.exec
[INFO] Analyzed bundle 'facturacion-web' with 4 classes
[INFO] BUILD SUCCESS
[INFO] Total time: 01:11 min
[INFO] Finished at: 2019-10-28T02:56:15-05:00
[INFO] -----
```

Ejecución satisfactoria

2.3.5. Los resultados de los test son almacenados en formato xml y se encuentran en la siguiente ruta “fuentes\target\surefire-reports\”



Se genera un reporte por cada clase Test

3. Pruebas de Integración

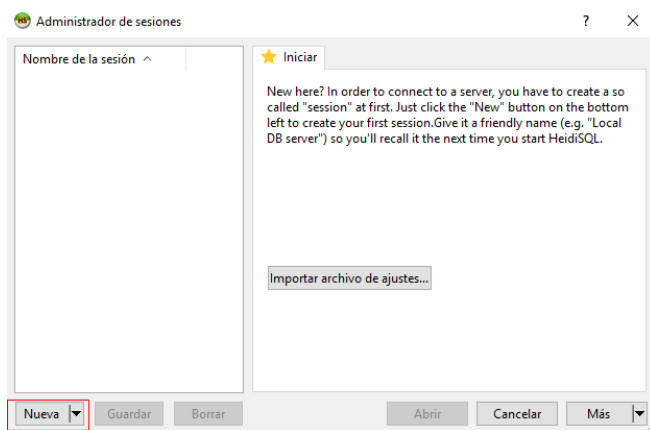
3.1. Requisitos

3.1.1. Crear la Base de Datos “facturacion-web”

- Ingresar al aplicativo “HeidiSQL” que se encuentra en el escritorio

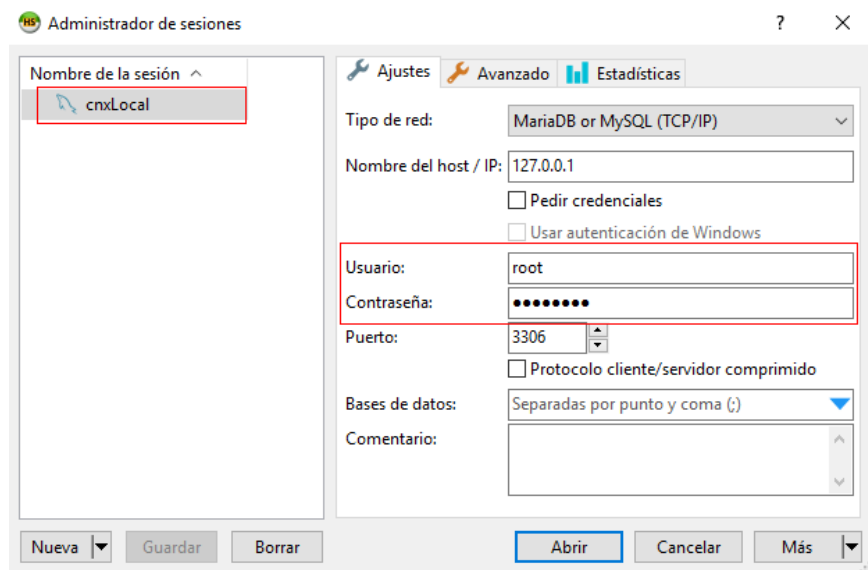


- Clic en el botón “Nueva”

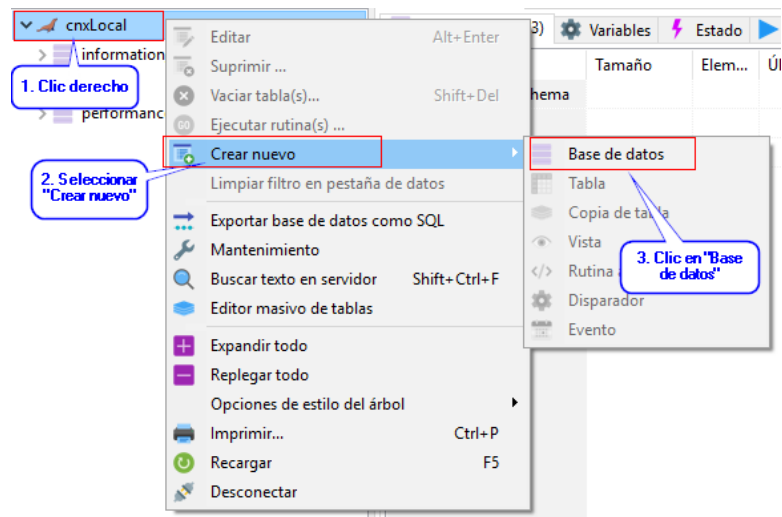


- c. Completar los datos de la conexión, clic en el botón “Guardar” y luego clic en el botón “Abrir”

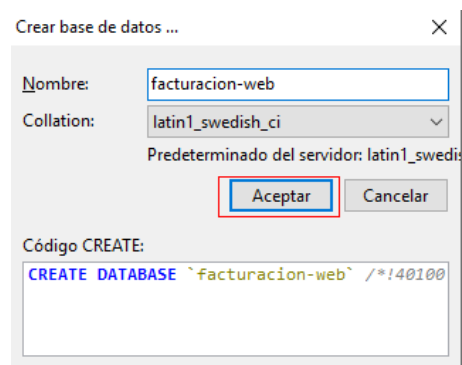
Nombre de la sesión: cnxLocal
 Usuario: root
 Clave: admin123



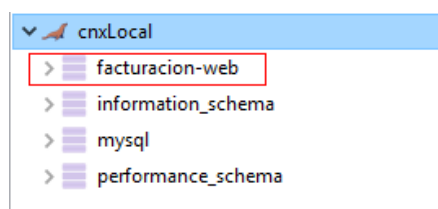
- d. En el panel izquierdo seleccionar el nombre de la sesión; luego, hacer clic derecho, seleccionar la opción “Crear nuevo” y hacer clic en “Base de datos”



- e. En el campo nombre de la ventana “Crear base de datos...” escribir “facturacion-web” y hacer clic en el botón “Aceptar”

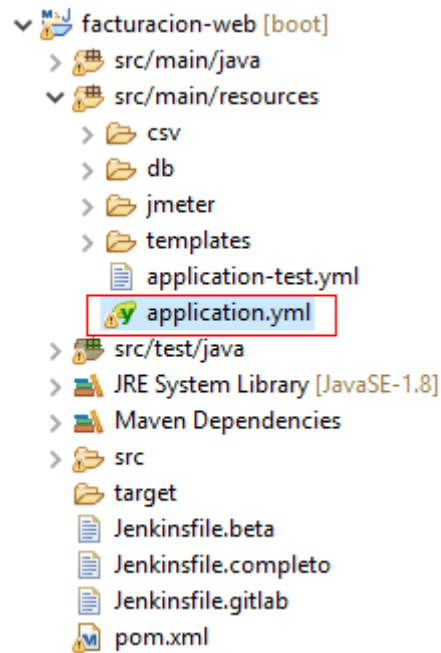


- f. Luego, verificar que la base de datos ha sido creada satisfactoriamente.

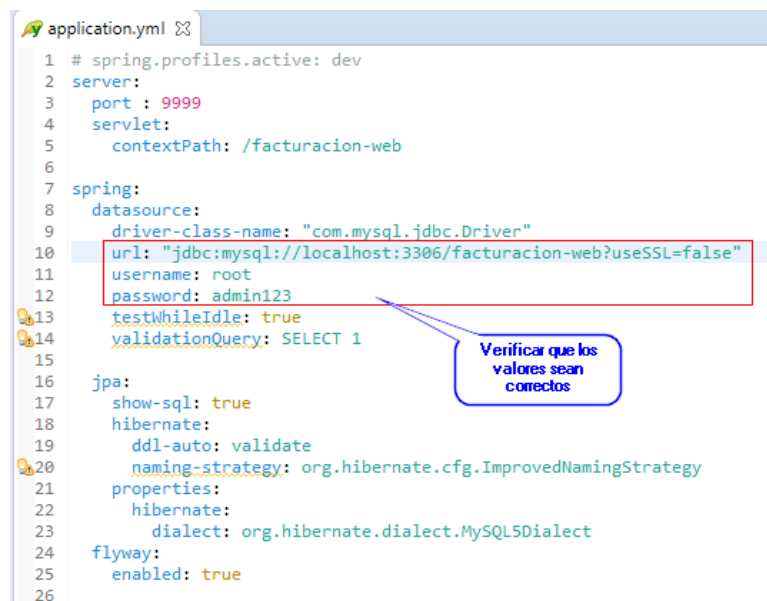


3.2. Ejecutar Pruebas de Integración desde Eclipse

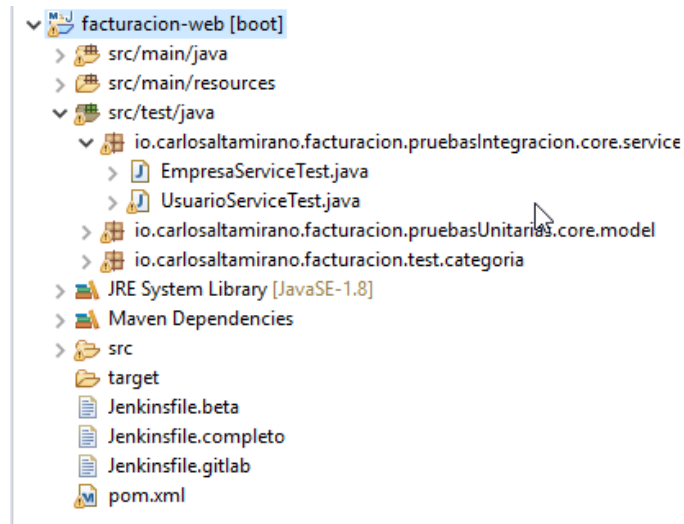
- 3.2.1. Abrir el archivo “application.yml” que se encuentra en la ruta “src/main/resources”



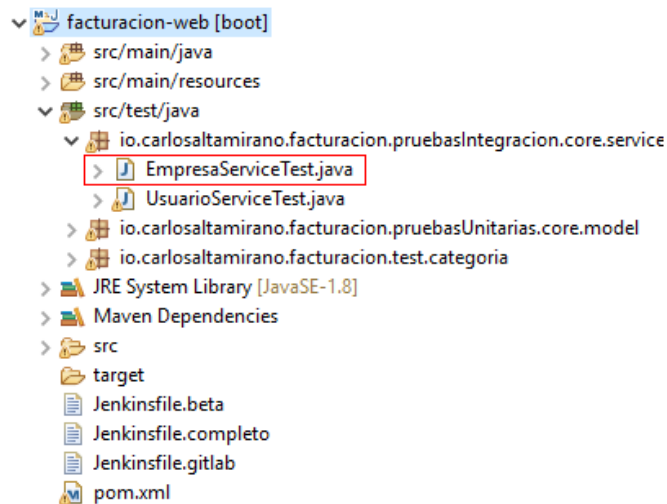
3.2.2. Verificar que los valores de los atributos “url, username y password” sean correctos.



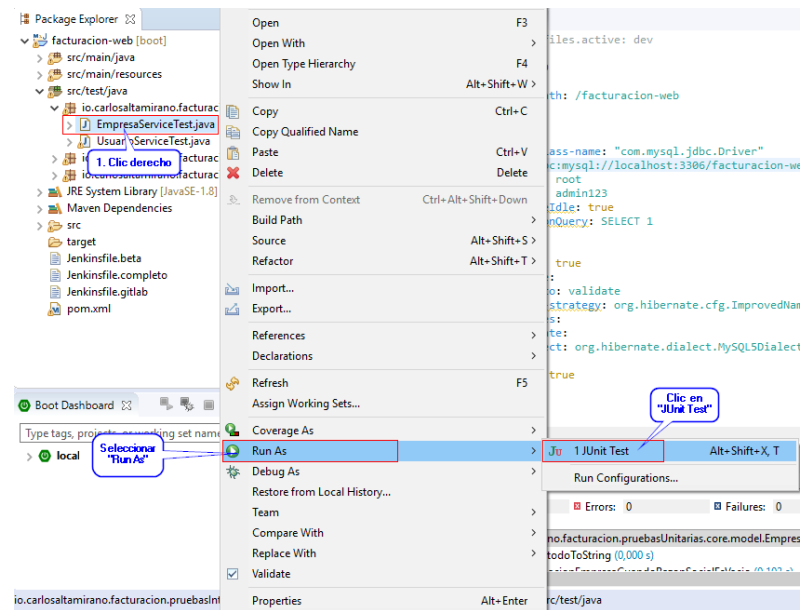
3.2.3. Expandir el proyecto “facturacion-web”, la carpeta “src/test/java” y el paquete “io.carlosaltamirano.facturacion.pruebasIntegracion.core.model”



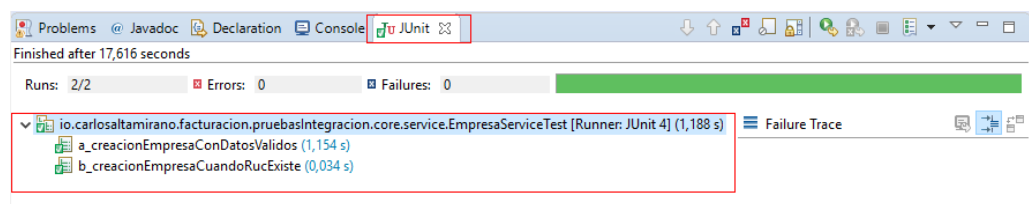
3.2.4. Seleccionar la clase EmpresaServiceTest



3.2.5. Clic derecho, seleccionar la opción “Run As” y clic en “JUnit Test”



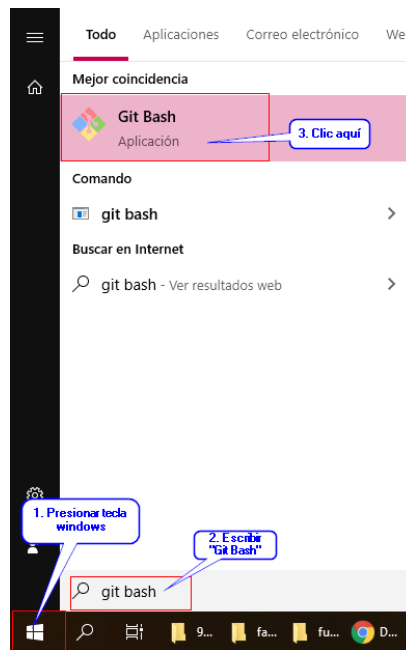
3.2.6. En la parte inferior del eclipse se abrirá el panel “JUnit” donde se mostrará el estado de la ejecución de las pruebas de integración.



3.3. Ejecutar Pruebas de Integración desde Línea de Comandos

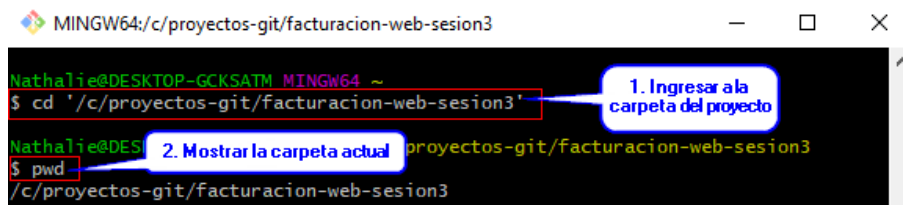
El proyecto “facturacion-web” fue creado usando “Maven”, eso quiere decir que podemos controlar el ciclo de vida del proyecto (compilar, test, test de integración, empaquetado y despliegue) a través de la línea de comandos.

3.3.1. Abrir el programa “Git Bash” (Presionar la tecla Windows, luego escribir “git bash”) y hacer clic en el programa.



3.3.2. Ir a la carpeta “C:\proyectos-git\facturacion-web-sesion3” usando el siguiente comando:

```
$ cd '/c/proyectos-git/facturacion-web-sesion3'
```



3.3.3. Para ejecutar las pruebas de integración con maven se deberá ejecutar el siguiente comando:

```
$ mvn -f fuentes/pom.xml verify -DskipUTs=true
```

Donde:

- mvn: commando maven
- -f fuentes/pom.xml: ubicación donde se encuentra el archivo pom.xml
- verify: goal para ejecutar pruebas de integración
- -DskipUTs=true: parámetro que permite omitir la ejecución de pruebas unitarias


```
MINGW64:/c:/proyectos-git/facturacion-web-sesion3
Nathalie@DESKTOP-GCKSATM MINGW64 /c:/proyectos-git/facturacion-web-sesion3
$ mvn -f fuentes/pom.xml verify -DskipUTs=true
[INFO] Scanning for projects...
[INFO] Building facturacion-web 1.0.0-SNAPSHOT
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-war-plugin/3.0.0-M3/maven-war-plugin-3.0.0-M3.jar
[INFO] 0 [sig] sh 1522! sigpacket::process: Suppressing signal 18 to win32 process (pid 0)
[INFO] facturacion-web >-----
[INFO] Building facturacion-web 1.0.0-SNAPSHOT
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-war-plugin/3.0.0-M3/maven-war-plugin-3.0.0-M3.jar
[INFO] 0 [sig] sh 1522! sigpacket::process: Suppressing signal 18 to win32 process (pid 0)
[INFO] facturacion-web >-----
```

1. Ejecutar comando maven

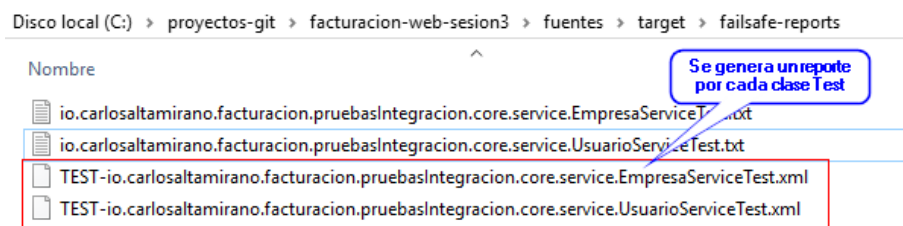
2. Automáticamente se descarga las dependencias para ejecutar las Pruebas de Integración

3.3.4. Finalmente, se debe mostrar el resultado de la ejecución de los tests.

```
[INFO] Results:
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0
[INFO] --- maven-failsafe-plugin:3.0.0-M3:verify (default) @ facturacion-web ---
[INFO] --- jacoco-maven-plugin:0.8.4:report-integration (it-report) @ facturacion-web ---
[INFO] Loading execution data file C:\proyectos-git\facturacion-web-sesion3\Fuentes\target\jacoco-it.exec
[INFO] Analyzed bundle 'facturacion-web' with 4 classes
[INFO] BUILD SUCCESS
[INFO] Total time: 01:06 min
[INFO] Finished at: 2019-10-28T12:19:22-05:00
[INFO] -----
```

Ejecución satisfactoria

3.3.5. Los resultados de los test de integración son almacenados en formato xml y se encuentran en la siguiente ruta “fuentes\target\failsafe-reports\”

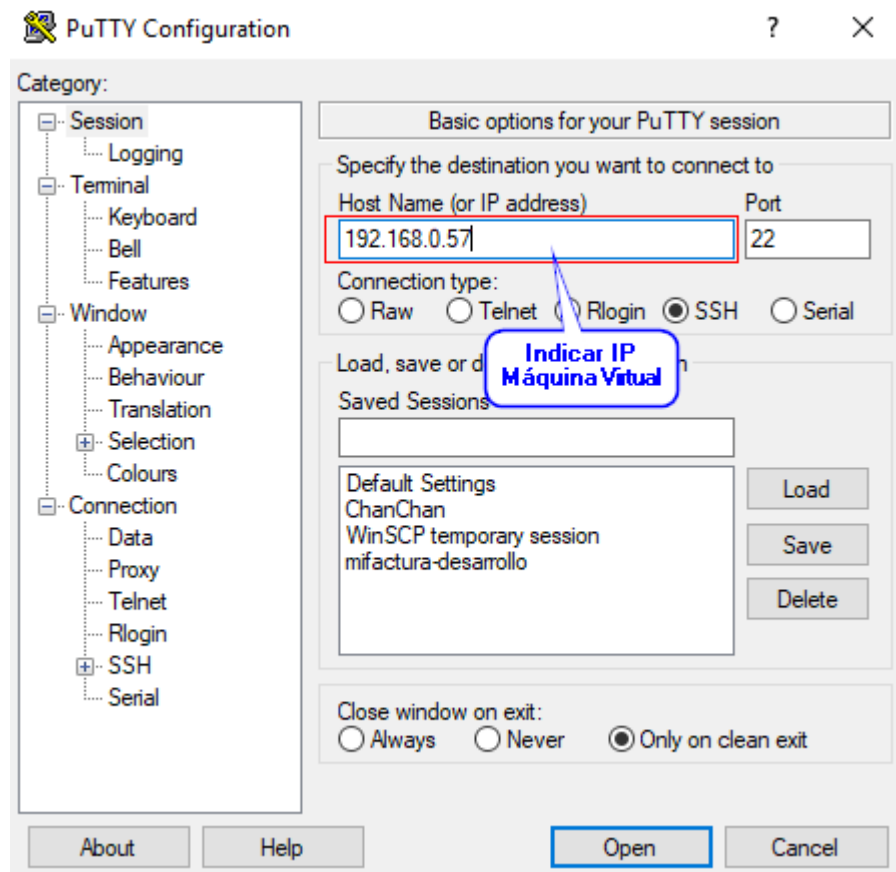


4. Análisis de Código Estático

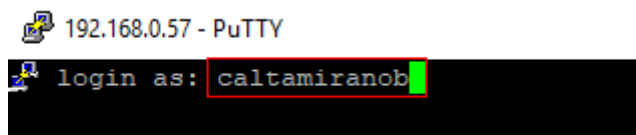
4.1. Requisitos

4.1.1. Iniciar el contenedor SonarQube

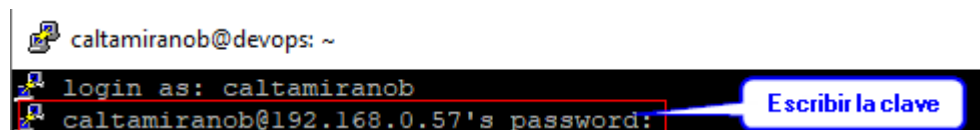
- Ingresar a la máquina virtual a través del putty, indicar la IP de la máquina virtual y hacer clic en el botón “Open”



- b. Indicar el usuario “caltamiranob” y presionar la tecla ENTER



- c. Ingresar la clave “admin123” para iniciar sesión en la máquina virtual.



- d. Finalmente, se mostrará la siguiente ventana donde podemos ejecutar nuestros comandos.

```
caltamiranob@devops: ~  
login as: caltamiranob  
caltamiranob@192.168.0.57's password:  
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-142-generic x86_64)  
  
 * Documentation:  https://help.ubuntu.com  
 * Management:    https://landscape.canonical.com  
 * Support:       https://ubuntu.com/advantage  
  
 * Kata Containers are now fully integrated in Charmed Kubernetes 1.16!  
   Yes, charms take the Krazy out of K8s Kata Kluster Konstruktion.  
  
      https://ubuntu.com/kubernetes/docs/release-notes  
  
23 packages can be updated.  
11 updates are security updates.  
  
New release '18.04.3 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Mon Oct 28 13:35:32 2019  
caltamiranob@devops:~$
```

- e. Iniciar el contenedor sonarqube con el siguiente comando:

```
$ docker container start sonarqube
```

```
caltamiranob@devops: ~  
caltamiranob@devops:~$ docker container start sonarqube  
sonarqube
```

- f. Para ver los logs del contenedor sonarqube ejecutar el siguiente comando

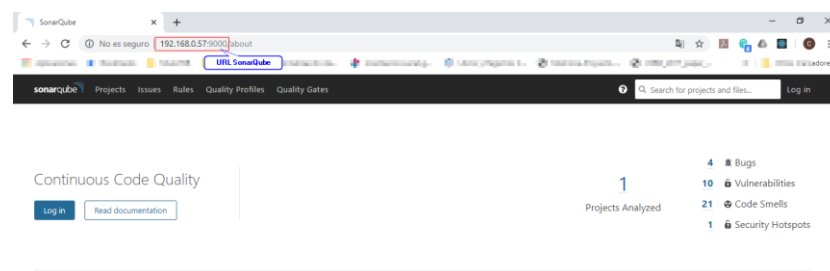
```
$ docker logs sonarqube -f --since=1m
```

```
caltamiranob@devops:~$ docker logs sonarqube -f --since=1m  
tail: './logs/es.log' has been replaced; following new file  
2019.10.28 17:52:07 INFO es[[o.e.e.NodeEnvironment] using [1] data paths, mount  
ts [[/opt/sonarqube/data/1/], net usable_space [15.7gb], total_sp  
ace [217gb], ty  
2019.10.28 17:52:07 INFO es[[o.e.e.NodeEnvironment] heap s  
pressed ordinary object pointers [true]  
2019.10.28 17:52:08 INFO es[[o.e.n.Node] node name [sonarqube], node ID [qQqPJ  
iIcTfaulgwZiKOYA]
```

Logs del contenedor

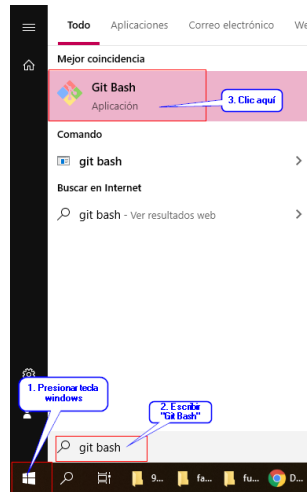
Comando para visualizar logs de contenedor

- g. Ingresar la siguiente url: http://IP_MAQUINA_VIRTUAL:9000 en cualquier navegador para validar que el aplicativo SonarQube ha iniciado exitosamente.



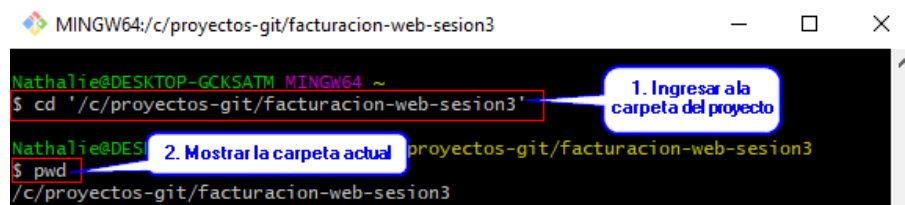
4.2. Ejecutar el análisis de código estático desde línea de comandos

4.2.1. Abrir el programa “Git Bash” (Presionar la tecla Windows, luego escribir “git bash”) y hacer clic en el programa.



4.2.2. Ir a la carpeta “C:\proyectos-git\facturacion-web-sesion3” usando el siguiente comando:

```
$ cd '/c/proyectos-git/facturacion-web-sesion3'
```

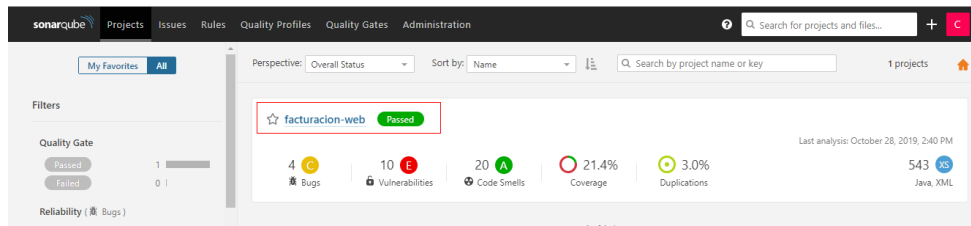


4.2.3. Para ejecutar el análisis de código estático con sonarqube se debe ejecutar el siguiente comando:

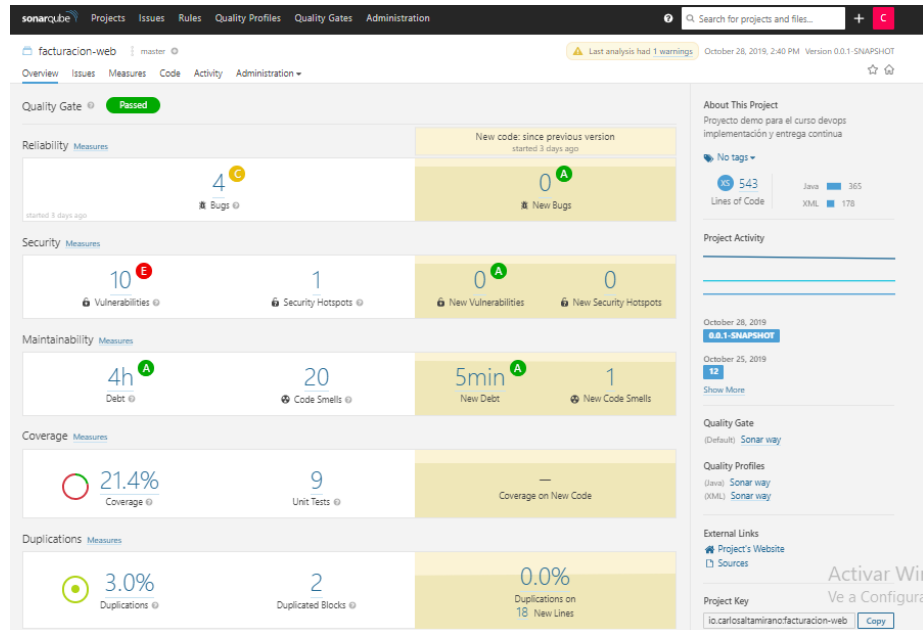
```
$ mvn -f fuentes/pom.xml sonar:sonar -Dsonar.host.url=http://{IP_SERVIDOR}:9000
```

Donde:

- *mvn*: commando maven
- *-f fuentes/pom.xml*: ubicación donde se encuentra el archivo *pom.xml*
- *verify*: goal para ejecutar pruebas de integración
- *-Dsonar.host.url*: IP del servidor sonarqube



4.2.7. Hacer clic en el proyecto “facturacion-web” para mostrar el dashboard del proyecto analizado.



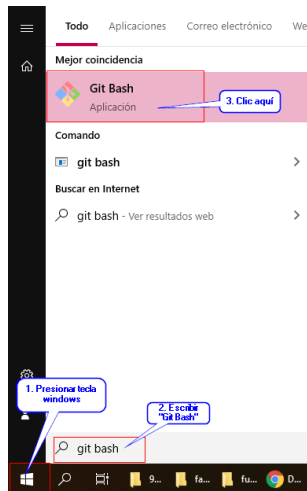
5. Generar Build del proyecto

5.1. Requisitos

Ninguno

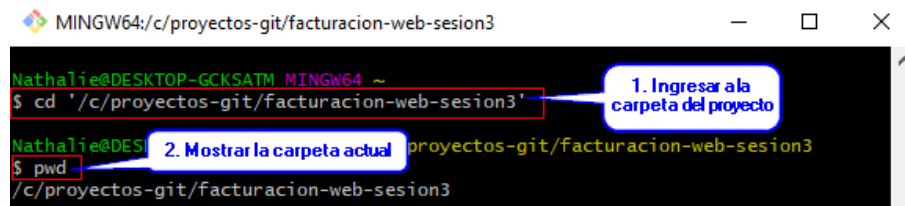
5.2. Generar Build desde la línea de comandos

5.2.1. Abrir el programa “Git Bash” (Presionar la tecla Windows, luego escribir “git bash”) y hacer clic en el programa.



5.2.2. Ir a la carpeta “C:\proyectos-git\facturacion-web-sesion3” usando el siguiente comando:

```
$ cd '/c/proyectos-git/facturacion-web-sesion3'
```

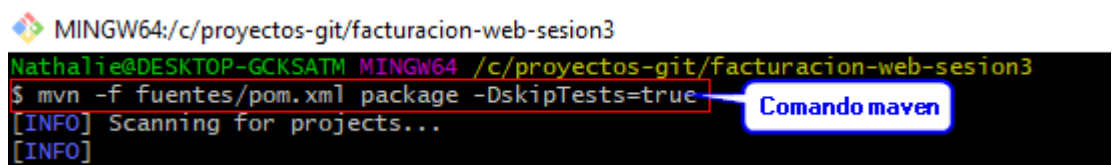


5.2.3. Generar jar del proyecto “facturacion-web” ejecutando el siguiente comando:

```
$ mvn -f fuentes/pom.xml package -DskipTests=true
```

Donde:

- ✓ **mvn**: comando maven
- ✓ **-f fuentes/pom.xml**: lugar donde se encuentra el archivo pom.xml
- ✓ **package**: goal para empaquetar el proyecto y generar el jar
- ✓ **-DskipTests=true**: parámetro que omitirá la ejecución de las pruebas



5.2.4. Al finalizar la ejecución del comando “mvn package” se muestra el siguiente mensaje

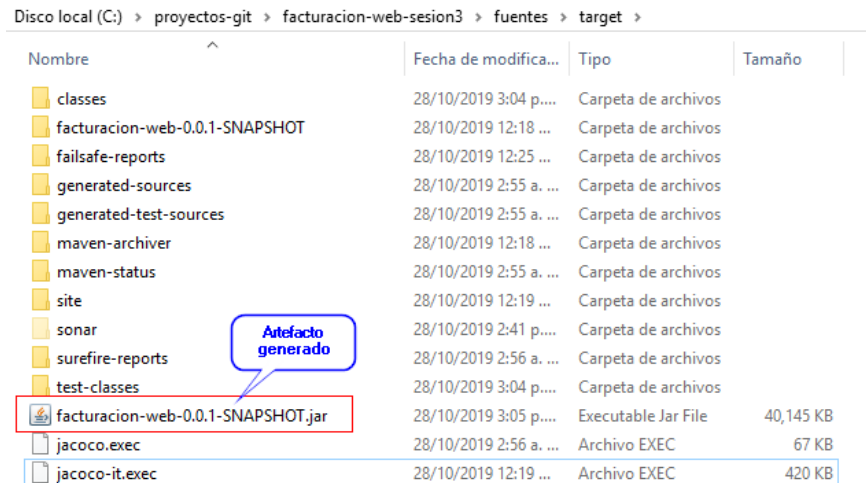
```

[INFO] --- jacoco-maven-plugin:0.8.4:report (it-report) @ facturacion-web ---
[INFO] Loading execution data file C:\proyectos-git\facturacion-web-sesion3\fuentes\target\jacoco.exec
[INFO] Analyzed bundle 'facturacion-web' with 4 classes
[WARNING] Classes in bundle 'facturacion-web' do not match with execution data. For report generation the same class files must be used as at runtime.
[WARNING] Execution data for class io.carlosaltamirano\facturacion\core\model\Empresa does not match.
[INFO] --- maven-jar-plugin:3.0.2:jar (default-jar) @ facturacion-web ---
[INFO] Building jar: C:\proyectos-git\facturacion-web-sesion3\fuentes\target\facturacion-web-0.0.1-SNAPSHOT.jar
[INFO] --- spring-boot-maven-plugin:2.0.4.RELEASE:repackage (default) @ facturacion-web ---
[INFO] BUILD SUCCESS
[INFO] Total time: 23.099 s
[INFO] Finished at: 2019-10-28T15:05:33-05:00
[INFO]

```

Después de ejecutar el comando mvn package se debe generar el componente jar

5.2.5. El comando anterior ha generado un archivo con extensión “.jar” que se encuentra en la carpeta “target” del proyecto “facturacion-web-sesion3”

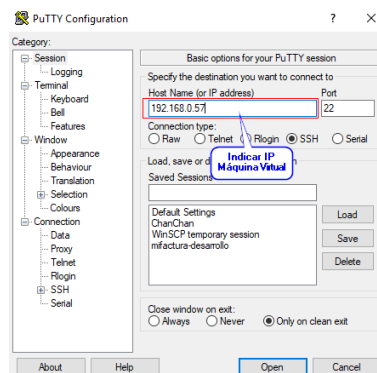


6. Integración con Jenkins

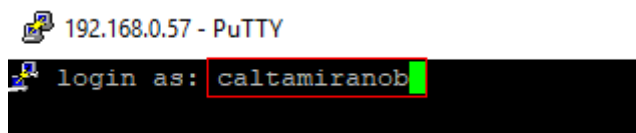
6.1. Pre-Requisitos

6.1.1. Iniciar los contenedores GitLab, Jenkins, SonarQube, Artifactory y MySQL

a. Ingresar a la máquina virtual a través del Putty



b. Indicar el usuario “caltamiranob” y presionar la tecla ENTER



- c. Ingresar la clave “admin123” y presionar ENTER para ingresar al servidor.

```
caltamiranob@devops: ~  
login as: caltamiranob  
caltamiranob@192.168.0.57's password:
```

Escribir la clave

- d. Se deberá mostrar una imagen similar a la siguiente:

```
caltamiranob@devops: ~  
login as: caltamiranob  
caltamiranob@192.168.0.57's password:  
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-142-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
* Kata Containers are now fully integrated in Charmed Kubernetes 1.16!  
  Yes, charms take the Krazy out of K8s Kata Kluster Konstruktion.  
  
  https://ubuntu.com/kubernetes/docs/release-notes  
  
23 packages can be updated.  
11 updates are security updates.  
  
New release '18.04.3 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Mon Oct 28 13:35:32 2019  
caltamiranob@devops:~$
```

- e. Iniciar el contenedor **GitLab** con el siguiente comando:

```
$ docker container start gitlab
```

```
caltamiranob@devops: ~  
caltamiranob@devops:~$ docker container start gitlab  
gitlab
```

- f. Iniciar el contenedor **Jenkins** con el siguiente comando:

```
$ docker container start jenkins
```

```
caltamiranob@devops: ~  
caltamiranob@devops:~$ docker container start jenkins  
jenkins
```

- g. Iniciar el contenedor **Artifactory** con el siguiente comando:

```
$ docker container start artifactory-oss
```

```
caltamiranob@devops: ~  
caltamiranob@devops:~$ docker container start artifactory-oss  
artifactory-oss
```

- h. Iniciar el contenedor **SonarQube** con el siguiente comando:

```
$ docker container start sonarqube
```

```
caltamiranob@devops: ~  
caltamiranob@devops:~$ docker container start sonarqube  
sonarqube
```

- i. Iniciar el contenedor **Mysql** con el siguiente comando:

```
$ docker container start mysql
```

```
caltamiranob@devops: ~  
caltamiranob@devops:~$ docker container start mysql  
mysql
```

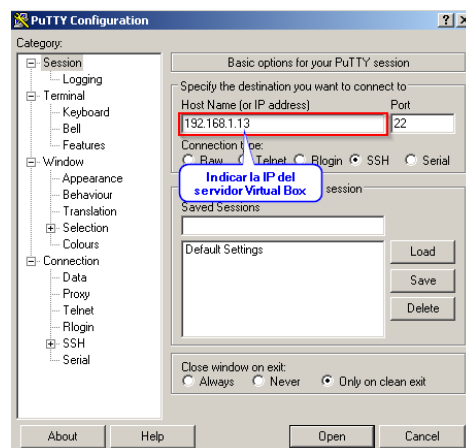
- j. Para verificar que todos los contenedores han sido iniciados, ejecutar el siguiente comando:

```
$ docker container ls
```

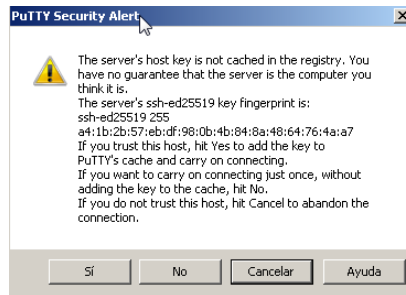
```
caltamiranob@devops: ~$ docker container ls  
CONTAINER ID   IMAGE      NAMES      COMMAND      CREATED      STATUS      PORTS  
48c2799ed019   mysql:5.6   mysql      "docker-entrypoint.s..." 6 days ago   Up 7 minutes   0.0.0.0:3306->3306/tcp  
36394c81921b   docker.bintray.io/jfrog/artifactory-oss:latest "entrypoint-artifac..." 2 weeks ago   Up 10 minutes   0.0.0.0:8081->8081/tcp  
b438bc7f557d   sonarqube   sonarqube   "/bin/run.sh" 2 weeks ago   Up 3 hours     0.0.0.0:9000->9000/tcp  
e44136253430   jenkinsci/blueocean   sonarqube   "/sbin/tini -- /usr..." 2 weeks ago   Up 9 minutes   0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp  
030d8b7a1ee   gishub/gishub-ow:latest   jenkins     "/assets/wrapper" 2 weeks ago   Up 9 minutes (unhealthy)   0.0.0.0:80->80/tcp, 0.0.0.0:4444->4444/tcp
```

6.1.2. Instalar cliente mysql en contenedor Jenkins

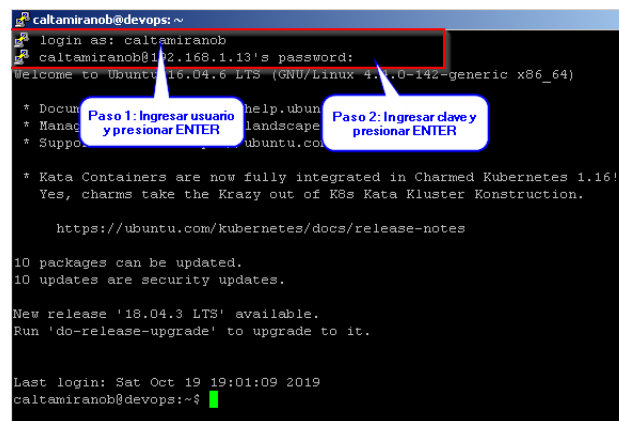
- a. Abrir el programa “Putty” y conectarse a la máquina virtual indicando el hostname (IP de la máquina virtual), el puerto (22 por defecto) y clic en el botón “Open”



- b. Clic en el botón “Sí”, si aparece este mensaje de alerta de seguridad:



- c. Ingresar el usuario (caltamiranob) y clave (admin123)



- d. Iniciar el contenedor Jenkins con el siguiente comando:
- ***docker container start jenkins***

```
caltamiranob@devops:~$ docker container start jenkins
```

- e. Ingresar a la consola del contenedor Jenkins con el siguiente comando:

- ***docker container exec -it jenkins /bin/bash***

```
caltamiranob@devops:~$ docker container exec -it jenkins /bin/bash
bash-4.4#
```

f. Instalar el cliente mysql ejecutando el siguiente comando:

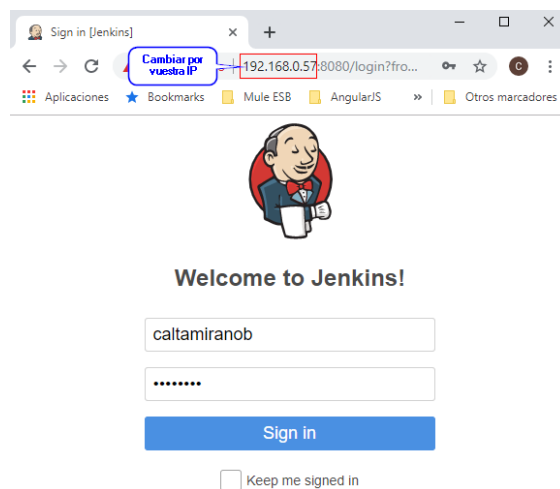
- ***apk add mysql-client***

```
bash-4.4# apk add mysql-client
fetch http://dl-cdn.alpinelinux.org/alpine/v3.9/main/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.9/community/x86_64/APKINDEX.tar.gz
(1/3) Installing mariadb-common (10.3.17-r0)
(2/3) Installing mariadb-client (10.3.17-r0)
(3/3) Installing mysql-client (10.3.17-r0)
Executing busybox-1.29.3-r10.trigger
OK: 444 MiB in 95 packages
```

6.1.3. Instalar Plugin SonarQube en Jenkins

a. Iniciar sesión el portal de Jenkins

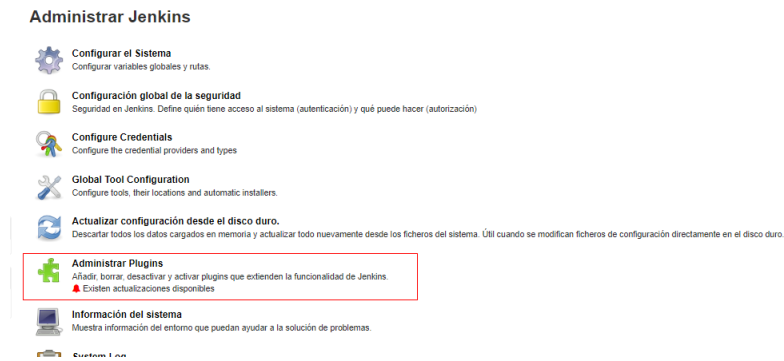
- Url: http://IP_MAQUINA_VIRTUAL:8080/
- Usuario: caltamiranob
- Clave: admin123



b. Ir a la opción “Administrar Jenkins” en el menú del lado izquierdo de nuestra pantalla.



c. Seleccionar la opción “Administrar Plugins”



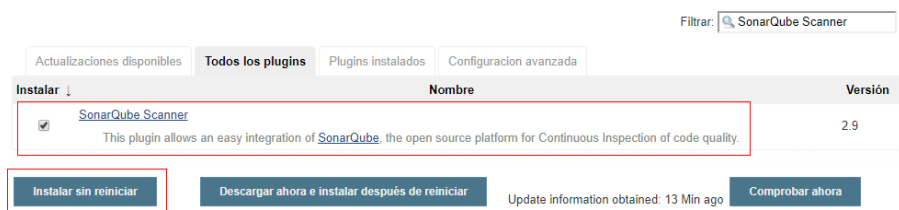
d. Clic en la pestaña “Todos los plugins”



e. En el campo “Filtrar” escribir “SonarQube Scanner” y presionar la tecla Enter.



f. Seleccionar el plugin “SonarQube Scanner” y clic en el botón “Instalar sin reiniciar”



g. Luego, aparecerá el mensaje de instalación exitosa.

Instalando/Actualizando plugins

Preparación

- Checking internet connectivity
- Checking update center connectivity
- Success

SonarQube Scanner

Actualizado

Loading plugin extensions

Success

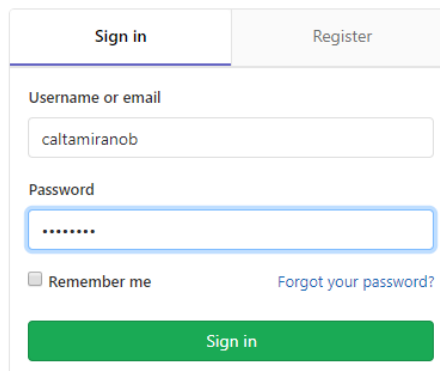
➡ [Volver al inicio de la página](#)
(puedes empezar a usar los plugins instalados inmediatamente)

➡ ☐ Reiniciar Jenkins cuando termine la instalación y no queden trabajos en ejecución

6.2. Creación de Proyecto en GitLab

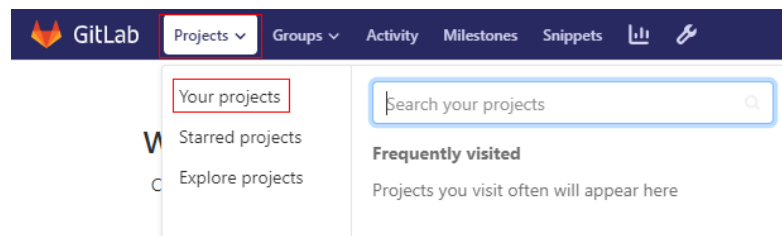
6.2.1. Iniciar sesión en el portal “GitLab”

- Url: http://IP_MAQUINA_VIRTUAL
- Usuario: caltamiranob
- Clave: admin123



The screenshot shows the GitLab sign-in interface. At the top, there are two tabs: 'Sign in' (active) and 'Register'. Below the tabs, there are two input fields: 'Username or email' with the value 'caltamiranob' and 'Password' with masked characters '.....'. Below the password field, there is a 'Remember me' checkbox and a link 'Forgot your password?'. At the bottom, there is a green 'Sign in' button.

6.2.2. Seleccionar la opción “Projects > Your Projects”



6.2.3. Clic en el botón “Create a Project” ó “New Project”

New project

6.2.4. En esta ventana registrar los datos del proyecto y hacer clic en el botón “Create Project”

- Nombre: facturacion-web-sesion03
- Descripción: Proyecto para el curso implementación devops
- Visibilidad: Publico

Blank project Create from template Import project

Project name
facturacion-web

Project URL Project slug
http://192.168.0.57/ facturacion-web

Want to house several dependent projects under the same namespace? [Create a group.](#)

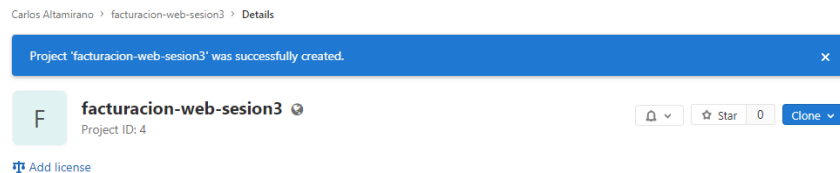
Project description (optional)
Proyecto para el curso implementación devops

Visibility Level ⓘ
☐ Private
Project access must be granted explicitly to each user.
☐ Internal
The project can be accessed by any logged in user.
☒ Public
The project can be accessed without any authentication.

☐ Initialize repository with a README
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

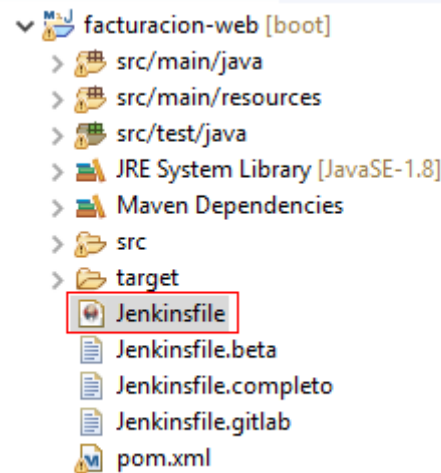
Create project Cancel

6.2.5. Al finalizar, se deberá mostrar la siguiente ventana



6.3. Creación de archivo Jenkinsfile (Pipeline)

6.3.1. Abrir archivo Jenkinsfile del proyecto “facturacion-web” en eclipse



6.3.2. Abrir el archivo Jenkinsfile. Este archivo contiene los pasos para realizar el proceso de integración y entrega continua. Se está usando el tipo “Declarative Pipeline” para la creación de los pasos.

```

1 def IP_SERVIDOR = "192.168.0.57"
2 def URL_GIT_REPOSITORIO = "http://${IP_SERVIDOR}/caltamiranob/facturacion-web-sesion03.git"
3
4 pipeline {
5     agent any
6     environment {
7         ENV_FACTURACION_WEB_BD_TEST_URL = "jdbc:mysql://${IP_SERVIDOR}:3306/facturacion-web-integration-test?useSSL=false"
8     }
9
10     stages {
11         stage('Descargar fuentes') {
12             steps {
13                 echo 'Descargando fuentes...'
14                 git credentialsId: 'caltamiranob', url: "${URL_GIT_REPOSITORIO}"
15             }
16         }
17
18         stage('Compilar') {
19             steps {
20                 echo 'Compilando...'
21                 sh 'mvn -f fuentes/pom.xml clean compile'
22             }
23         }
24
25         stage('Pruebas Unitarias') {
26             steps {
27                 echo 'Ejecutando pruebas unitarias...'
28                 sh 'mvn -f fuentes/pom.xml test'
29                 junit '**/target/surefire-reports/*.xml'
30             }
31         }
32
33         stage('Pruebas Integración') {
34             steps {
35                 echo 'Ejecutando pruebas integración...'
36             }
37         }
38     }
39 }

```

6.3.3. Cambiar el valor de las siguientes variables en el archivo “Jenkinsfile”:

```

def IP_SERVIDOR = "192.168.0.57"

```

Cambiar por la IP de vuestra máquina virtual

```

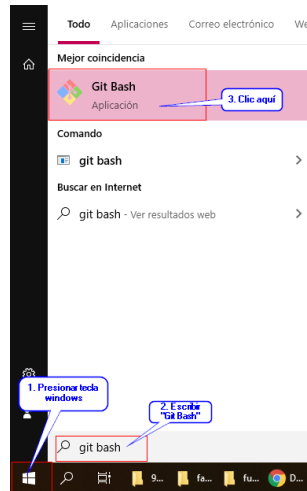
def URL_GIT_REPOSITORIO = "http://${IP_SERVIDOR}/caltamiranob/facturacion-web-sesion03.git"

```

Cambiar por la URL de su proyecto en GitLab

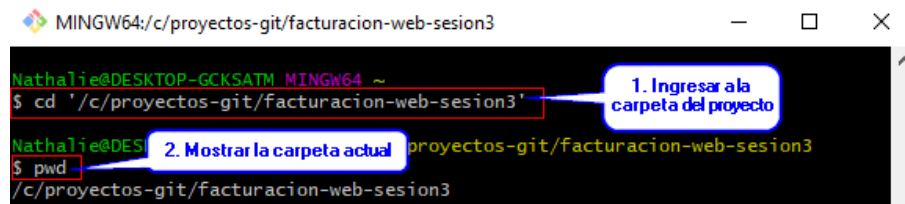
6.4. Sincronizar repositorio local y remoto de Git/GitLab

6.4.1. Abrir el programa “Git Bash” (Presionar la tecla Windows, luego escribir “git bash”) y hacer clic en el programa.



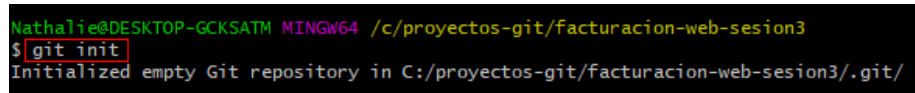
6.4.2. Ir a la carpeta “C:\proyectos-git\facturacion-web-sesion3” usando el siguiente comando:

```
$ cd '/c/proyectos-git/facturacion-web-sesion3'
```

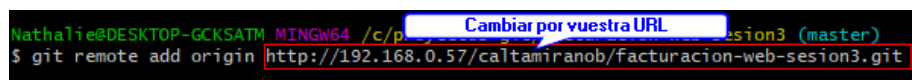


6.4.3. Ejecutar los siguientes comandos para sincronizar el repositorio local con el repositorio remoto

- **git init**



- **git remote add origin** <http://192.168.0.57/caltamiranob/facturacion-web-sesion3.git>



- **git add .**

```
Nathalie@DESKTOP-GCKSATM MINGW64 /c/proyectos-git/facturacion-web-sesion3 (master)
$ git add .
warning: CRLF will be replaced by LF in fuentes/Jenkinsfile.
The file will have its original line endings in your working directory
warning: CRLF will be replaced by LF in fuentes/Jenkinsfile.gitlab.
The file will have its original line endings in your working directory
```

- **git commit -m "Initial commit"**

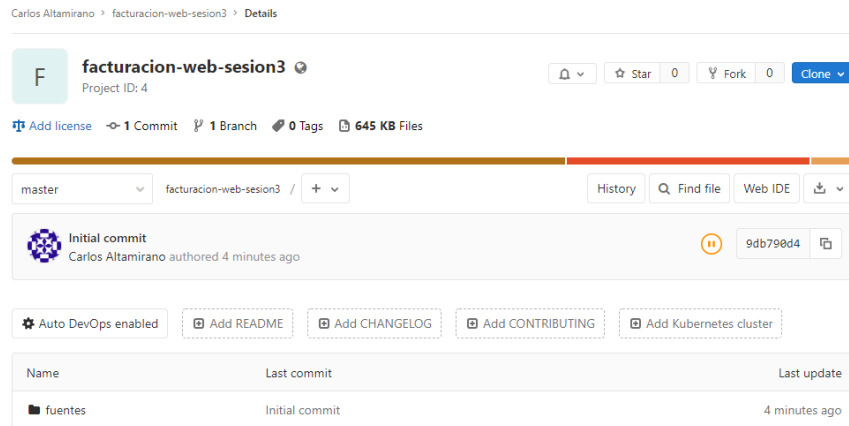
```
Nathalie@DESKTOP-GCKSATM MINGW64 /c/proyectos-git/facturacion-web-sesion3 (master)
$ git commit -m "Initial commit"
[master (root-commit) 9db790d] Initial commit
38 files changed, 2069 insertions(+)
create mode 100644 fuentes/.gitignore
create mode 100644 fuentes/Jenkinsfile
create mode 100644 fuentes/Jenkinsfile.beta
create mode 100644 fuentes/Jenkinsfile.completo
create mode 100644 fuentes/Jenkinsfile.gitlab
create mode 100644 fuentes/pom.xml
create mode 100644 fuentes/src/main/java/io/carlosaltamirano/facturacion/Application.java
create mode 100644 fuentes/src/main/java/io/carlosaltamirano/facturacion/api/EmpresaController.java
create mode 100644 fuentes/src/main/java/io/carlosaltamirano/facturacion/api/UsuarioController.java
create mode 100644 fuentes/src/main/java/io/carlosaltamirano/facturacion/core/exception/FacturacionWebException.java
create mode 100644 fuentes/src/main/java/io/carlosaltamirano/facturacion/core/exception/RucEmpresaExisteException.java
create mode 100644 fuentes/src/main/java/io/carlosaltamirano/facturacion/core/model/Empresa.java
create mode 100644 fuentes/src/main/java/io/carlosaltamirano/facturacion/core/model/Usuario.java
create mode 100644 fuentes/src/main/java/io/carlosaltamirano/facturacion/core/repository/EmpresaRepository.java
create mode 100644 fuentes/src/main/java/io/carlosaltamirano/facturacion/core/repository/UsuarioRepository.java
create mode 100644 fuentes/src/main/java/io/carlosaltamirano/facturacion/core/service/EmpresaService.java
create mode 100644 fuentes/src/main/java/io/carlosaltamirano/facturacion/core/service/UsuarioService.java
create mode 100644 fuentes/src/main/java/io/carlosaltamirano/facturacion/core/service/impl/EmpresaServiceImpl.java
create mode 100644 fuentes/src/main/java/io/carlosaltamirano/facturacion/core/service/impl/UsuarioServiceImpl.java
create mode 100644 fuentes/src/main/resources/application-test.yml
create mode 100644 fuentes/src/main/resources/application.yml
create mode 100644 fuentes/src/main/resources/csv/empresas.csv
create mode 100644 fuentes/src/main/resources/db/create_database_integration_test_mysql.sql
create mode 100644 fuentes/src/main/resources/db/create_database_mysql.sql
create mode 100644 fuentes/src/main/resources/db/delete_database_integration_test_mysql.sql
create mode 100644 fuentes/src/main/resources/db/migration/V1_0__init.sql
create mode 100644 fuentes/src/main/resources/db/migration/V1_1__usuario.sql
create mode 100644 fuentes/src/main/resources/jmeter/plan-pruebas-Facturacion-web.jmx
create mode 100644 fuentes/src/main/resources/jmeter/plan-pruebas-Facturacion-web.jtl
create mode 100644 fuentes/src/main/resources/templates/index-usuario.html
create mode 100644 fuentes/src/main/resources/templates/index.html
create mode 100644 fuentes/src/main/resources/templates/layout/default.html
create mode 100644 fuentes/src/test/java/io/carlosaltamirano/facturacion/pruebasIntegracion/core/service/EmpresaServiceTest.java
create mode 100644 fuentes/src/test/java/io/carlosaltamirano/facturacion/pruebasIntegracion/core/service/UsuarioServiceTest.java
create mode 100644 fuentes/src/test/java/io/carlosaltamirano/facturacion/pruebasUnitarias/core/model/EmpresaTest.java
create mode 100644 fuentes/src/test/java/io/carlosaltamirano/facturacion/pruebasUnitarias/core/model/UsuarioTest.java
create mode 100644 fuentes/src/test/java/io/carlosaltamirano/facturacion/test/categoria/PruebaIntegracion.java
create mode 100644 fuentes/src/test/java/io/carlosaltamirano/facturacion/test/categoria/PruebaUnitaria.java
```

- **git push -u origin master**

```
Nathalie@DESKTOP-GCKSATM MINGW64 /c/proyectos-git/facturacion-web-sesion3 (master)
$ git push -u origin master
Enumerating objects: 74, done.
Counting objects: 100% (74/74), done.
Delta compression using up to 4 threads
Compressing objects: 100% (58/58), done.
Writing objects: 100% (74/74), 20.69 KiB | 605.00 KiB/s, done.
Total 74 (delta 9), reused 0 (delta 0)
To http://192.168.0.11/caltamiranob/facturacion-web-sesion3.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

En caso les solicite credenciales, el usuario es "caltamiranob" y clave "admin123"

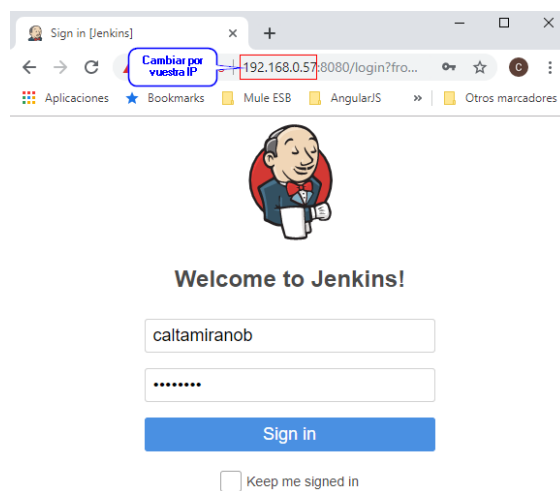
6.4.4. Ir al portal de GitLab y verificar que el proyecto ha sido sincronizado satisfactoriamente.



6.5. Creación de Proyecto en Jenkins

6.5.1. Iniciar sesión en el portal de Jenkins

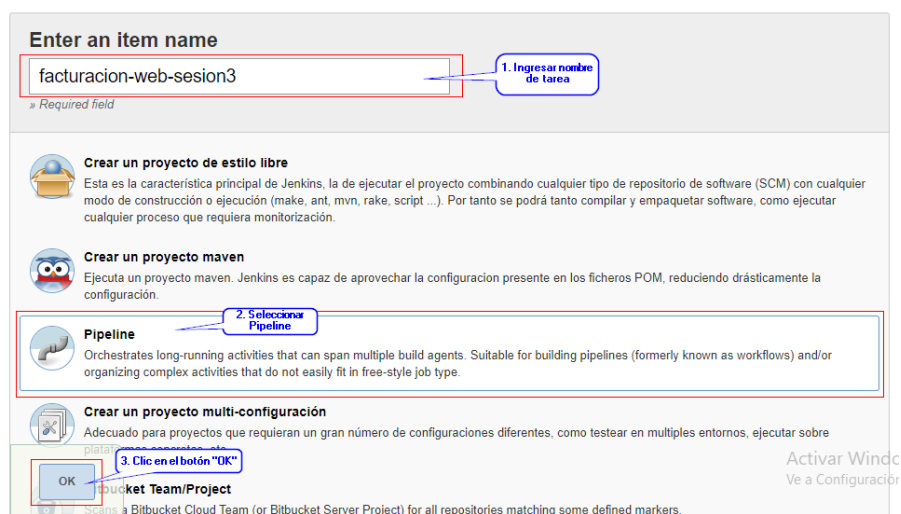
- Url: http://IP_MAQUINA_VIRTUAL:8080/
- Usuario: caltamiranob
- Clave: admin123



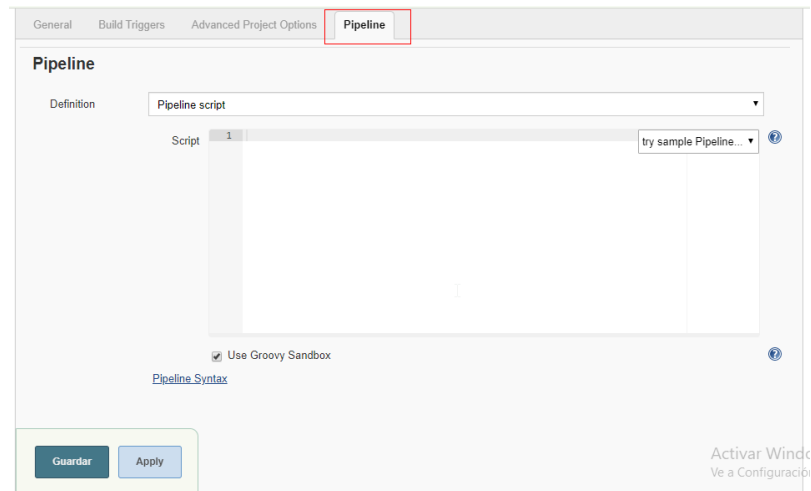
6.5.2. Clic en la opción “Nueva Tarea” del menú izquierdo.



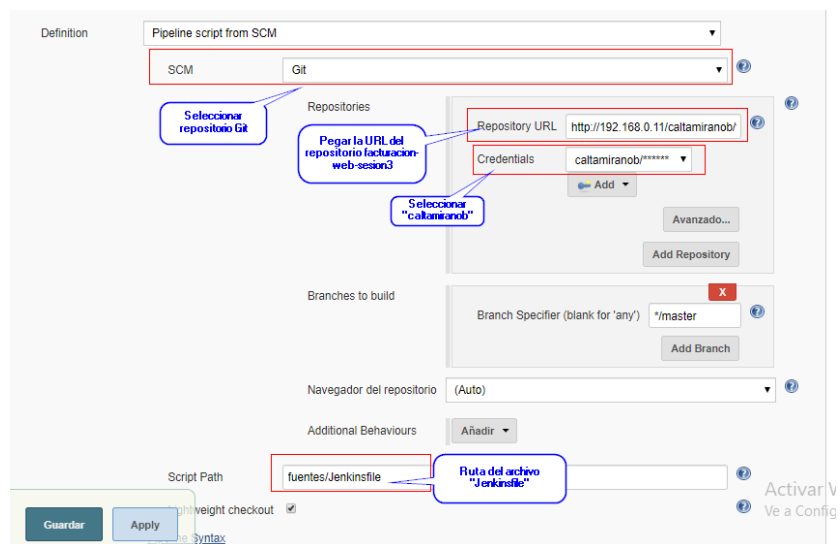
6.5.3. Ingresar “facturacion-web-sesion3” en nombre de tarea, seleccionar “Pipeline” y hacer clic en el botón “OK”



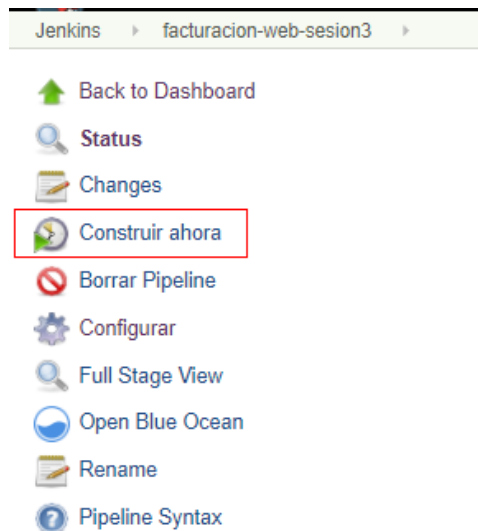
6.5.4. Seleccionar la pestaña “Pipeline”



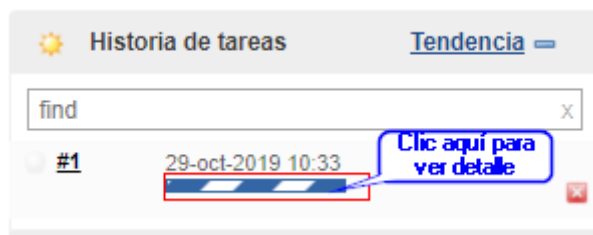
6.5.5. Seleccionar la opción “Pipeline script from SCM”, completar los datos según la imagen y hacer clic en el botón “Guardar”



6.5.6. Para ejecutar nuestro pipeline, hacer clic en la opción “Construir ahora”



6.5.7. En el panel “Historia de tareas” empezará a ejecutar nuestro pipeline, para poder ver los logs de ejecución hacer clic en la barra de progreso.



6.5.8. Luego, se mostrará la siguiente ventana donde nos mostrará los logs de la ejecución de nuestro pipeline.



6.5.9. Al final del log deberá aparecer de “End of Pipeline” que indica que el pipeline ha finalizado.

```

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 32.901 s
[INFO] Finished at: 2019-10-29T10:40:35Z
[INFO] -----
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline

```

6.5.10. Para regresar a la ventana principal del pipeline, hacer clic en “Back to Project” ó “facturación-web-sesion3”

Jenkins > **facturacion-web-sesion3** > #2

[Back to Project](#)

Status

Changes

Console Output

View as plain text

Edit Build Information

Git Build Data

No Tags

Salida de consola

Descartando 75 KB. [Registro completo](#)

```

/usr/lib/jvm/java-1.8-openjdk/jre/lib/jce.jar:/usr/lib/jvm/java-1
openjdk/jre/lib/jfr.jar:/usr/lib/jvm/java-1.8-openjdk/jre/classes
java.vendor.url.bug=https://icedtea.classpath.org/bugzilla, sun.i
SystemEnvironmentPropertySource@383882703 {name='systemEnvironmen
{PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bi
COPY_REFERENCE_FILE_LOG=/var/jenkins_home/copy_reference_file.log
HUDSON_HOME=/var/jenkins_home, RUN_CHANGES_DISPLAY_URL=http://jen
JOB_URL=http://jenkins:8080/job/facturacion-web-sesion3/, HUDSON_
ea6e26cc4977, LD_LIBRARY_PATH=/usr/lib/jvm/java-1.8-openjdk/jre/l
openjdk/jre/..lib/amd64, JENKINS_SERVER_COOKIE= durable-c96e094f5

```

6.5.11. Para verificar que el pipeline se ha ejecutado correctamente, todas las etapas deben estar de color “verde” como aparece en la imagen.

Stage View

	Declarative: Checkout SCM	Descargar fuentes	Compilar	Pruebas Unitarias	Pruebas Integración	Análisis Estático con SonarQube	Build
Average stage times: (Average full run time: ~4min 42s)	2s	2s	11s	16s	41s	43s	22s
#2 Oct 29 08:35 1 commit	2s	738ms	22s	32s	1min 22s	1min 27s	44s

6.5.12. Para revisar los reportes de las pruebas unitarias, integración y análisis estático realizar lo siguiente

- Ver reporte de pruebas unitarias e integración
 - Clic en la opción “Últimos resultados de tests”



[Últimos resultados de tests](#) (Sin fallas)

- Se mostrará el detalle de todas las pruebas realizadas

Resultados de los tests

0 fallidos

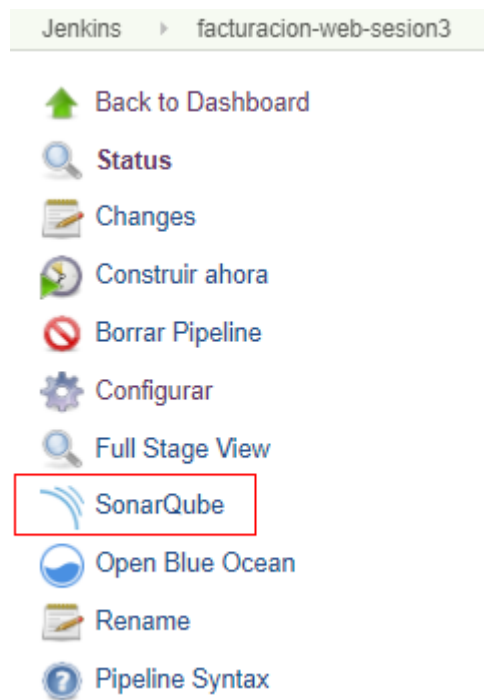
12 tests
Tardó 3.6 Seg.
[añadir descripción](#)

Todos los tests

Paquete	Duración	Fallidos (diferencias)	Omitir (diferencias)	Pass (diferencias)	Total (diferencias)
io.carlosaltamirano.facturacion.pruebasIntegracion.core.service	2.7 Seg	0	0	3 +3	3 +3
io.carlosaltamirano.facturacion.pruebasUnitarias.core.model	0.67 Seg	0	0	9 +9	9 +9

b. Ver reporte de análisis estático con SonarQube

- Clic en la opción “SonarQube” del panel izquierdo



- Se abrirá el dashboard de SonarQube donde nos muestra el proyecto “facturacion-web” analizado.

