

Elaboro	Documento	Versión	Descripción	Fecha
Luis Sánchez Martínez	Práctica Configuración de Jacoco y Pruebas Unitarias	1.0	Creación de documento	27/03/2024

## Práctica Configuración de Jacoco y Pruebas Unitarias.

Tiempo estimado: 2 horas.

Uso de:

- IntelliJ
- jacoco
- sonarQube
- Junit
- Mockito

### Agregar dependencia jacoco y configurar en archivo pom.xml

1. La dependencia se deberá agregar en la sección de build:

```

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <configuration>
        <excludes>
          <exclude>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
          </exclude>
        </excludes>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.jacoco</groupId>
      <artifactId>jacoco-maven-plugin</artifactId>
      <version>0.8.8</version>
      <configuration>
        <excludes>
          <exclude>**/permiso/*Application.class</exclude>
          <exclude>**/configuration/*.class</exclude>
          <exclude>**/model/**/**.class</exclude>
          <exclude>**/test/**/**.class</exclude>
        </excludes>
      </configuration>
    </plugin>
  </plugins>
</build>

```

2. Indicar los paquetes y las clases que se van a excluir de la cobertura de las pruebas unitarias:

```

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <configuration>
        <excludes>
          <exclude>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
          </exclude>
        </excludes>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.jacoco</groupId>
      <artifactId>jacoco-maven-plugin</artifactId>
      <version>0.8.8</version>
      <configuration>
        <excludes>
          <exclude>**/permiso/*Application.class</exclude>
          <exclude>**/configuration/*.class</exclude>
          <exclude>**/model/**/*.*.class</exclude>
          <exclude>**/test/**/*.*.class</exclude>
        </excludes>
      </configuration>
    </plugin>
  </plugins>
</build>

```

- Indicar en que momento se realizara la validación de java code coverage y generación del reporte:

```

86      <exclude>**/permiso/*Application.class</exclude>
87      <exclude>**/configuration/*.class</exclude>
88      <exclude>**/model/**/*.*.class</exclude>
89      <exclude>**/test/**/*.*.class</exclude>
90    </excludes>
91  </configuration>
92  <executions>
93    <execution>
94      <goals>
95        <goal>prepare-agent</goal>
96      </goals>
97    </execution>
98    <execution>
99      <id>report</id>
100      <phase>test</phase>
101      <goals>
102        <goal>report</goal>
103      </goals>
104    </execution>
105  </executions>
106 </plugin>
107 </plugins>
108 </build>
109
110 </project>
111

```

- Integrar la dependencia de Mockito:

```

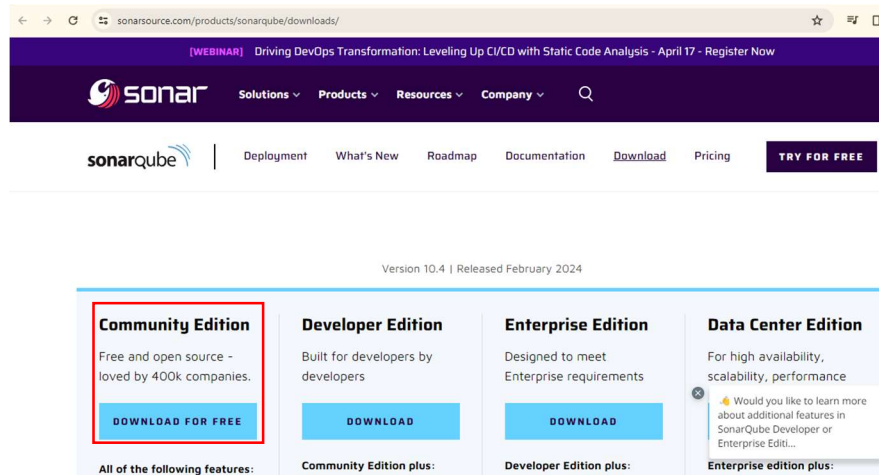
47      <groupId>org.modelmapper</groupId>
48      <artifactId>modelmapper</artifactId>
49      <version>3.2.0</version>
50    </dependency>
51
52    <dependency>
53      <groupId>org.mockito</groupId>
54      <artifactId>mockito-core</artifactId>
55      <version>4.8.0</version>
56      <scope>test</scope>
57    </dependency>
58
59    <dependency>
60      <groupId>org.springframework.boot</groupId>
61      <artifactId>spring-boot-starter-test</artifactId>
62      <scope>test</scope>
63    </dependency>
64  </dependencies>
65

```

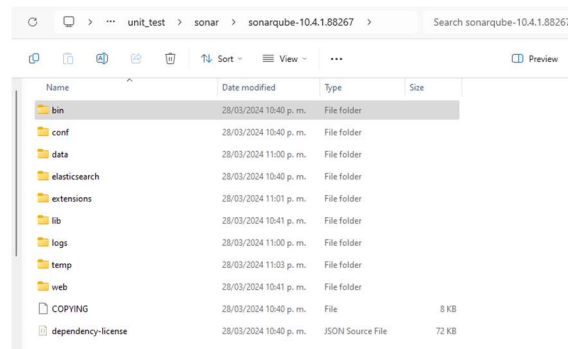
## Descargar SonarQube y desplegar servidor

1. Descargar sonarQube community de la página:

<https://www.sonarsource.com/products/sonarqube/downloads/>



2. Descomprimir la carpeta:



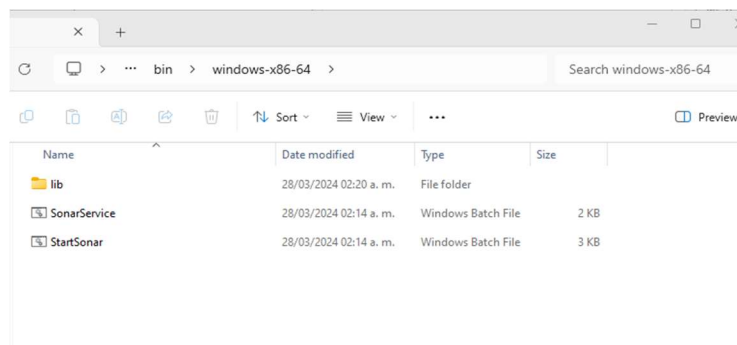
3. Actualizar el archivo settings.xml localizado en <MAVEN\_HOME>/conf colocando las global settings:

```

<settings>
  <pluginGroups>
    <pluginGroup>org.sonarsource.scanner.maven</pluginGroup>
  </pluginGroups>
  <profiles>
    <profile>
      <id>sonar</id>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>
      <properties>
        <!-- Optional URL to server. Default value is http://localhost:9000 -->
        <sonar.host.url>
          http://localhost:9000
        </sonar.host.url>
      </properties>
    </profile>
  </profiles>
</settings>

```

4. Ejecutar en caso de desplegar en Windows el archivo bat StartSonar que se encuentra en la ruta:  
path\_sonarqube\bin\windows-x86-64

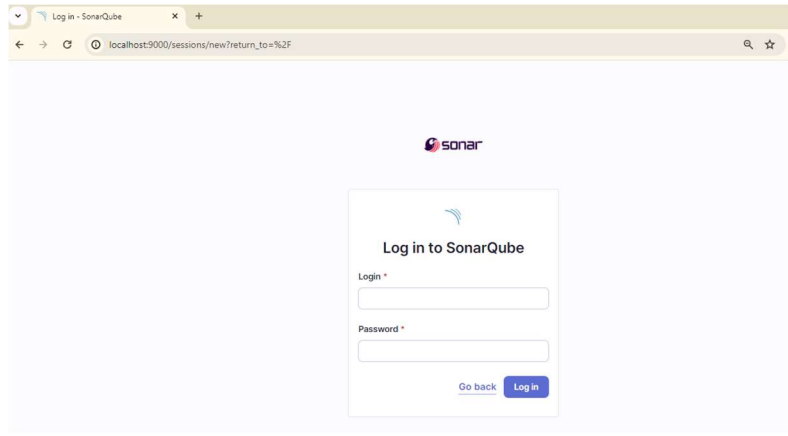


```

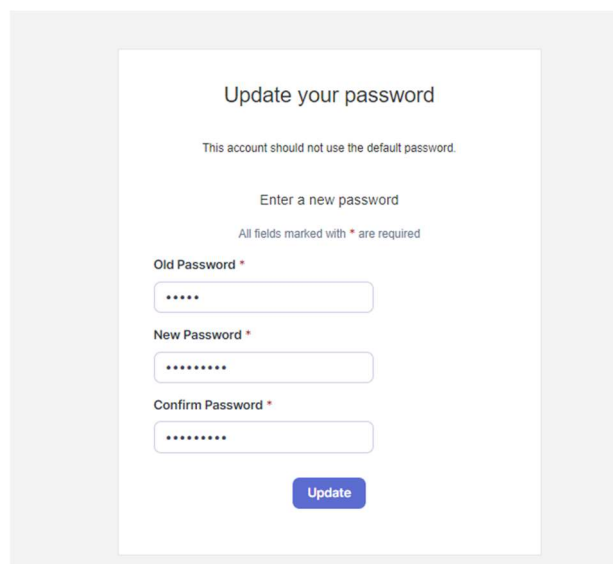
2024.03.28 22:59:36 INFO app[][o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
2024.03.28 23:00:28 INFO app[][o.s.a.SchedulerImpl] Process[es] is up

```

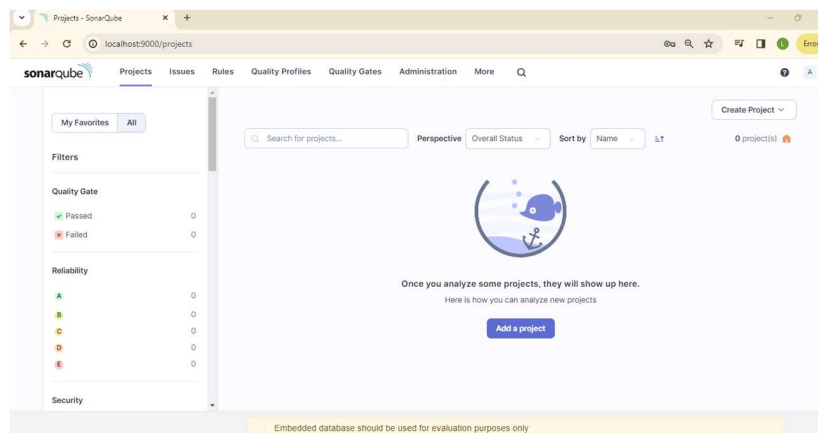
5. Consultar el sitio de Sonarqube



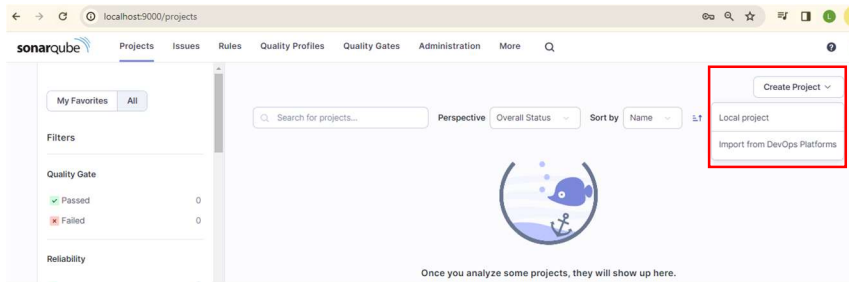
6. Autenticarse usando el usuario admin y el password admin y te solicitará cambiar el password:



7. La pantalla principal muestra que no se encuentra ningún proyecto actualmente:



8. Crear un nuevo proyecto dando clic en el botón de Create Project y seleccionando la opción Local Project:



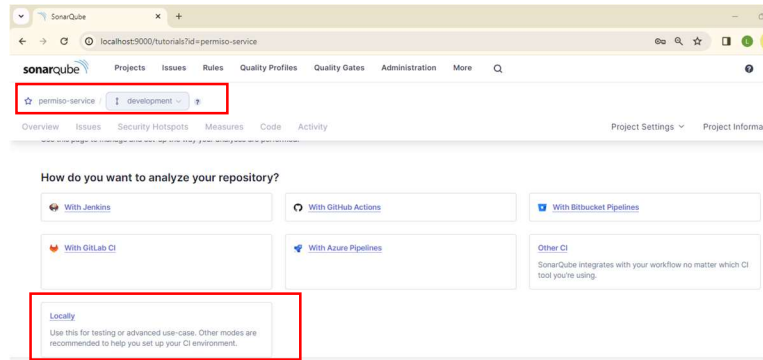
9. Configurar el nombre que se desplegará, el identificador (key) y la rama del proyecto, el cual dejaremos como development y dar clic en el botón Next:

The screenshot shows the 'Create a local project' form in SonarQube. The form has three main sections: 'Project display name', 'Project key', and 'Main branch name'. Each section has a text input field and a green checkmark icon indicating the input is valid. The 'Project display name' and 'Project key' fields both contain 'permiso-service'. The 'Main branch name' field contains 'development'. Below the 'Main branch name' field, there is a link 'Learn More'. At the bottom of the form, there are two buttons: 'Cancel' and 'Next'. The 'Next' button is highlighted with a blue background.

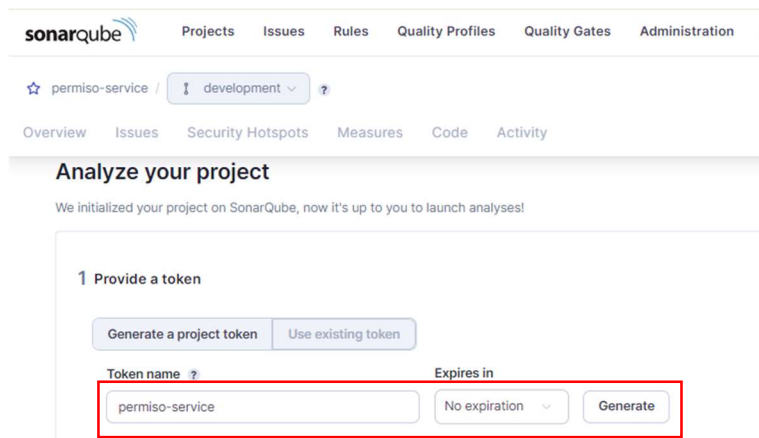
10. Seleccionar la opción de usar las configuraciones globales y dar clic en el botón Create project:

The screenshot shows the 'Set up project for Clean as You Code' form in SonarQube. The form has two main sections: 'Choose the baseline for new code for this project' and 'Define a specific setting for this project'. In the first section, the 'Use the global setting' radio button is selected. In the second section, the 'Previous version' radio button is selected. The 'Previous version' option is highlighted with a blue background. The 'Number of days' option is also visible but not selected.

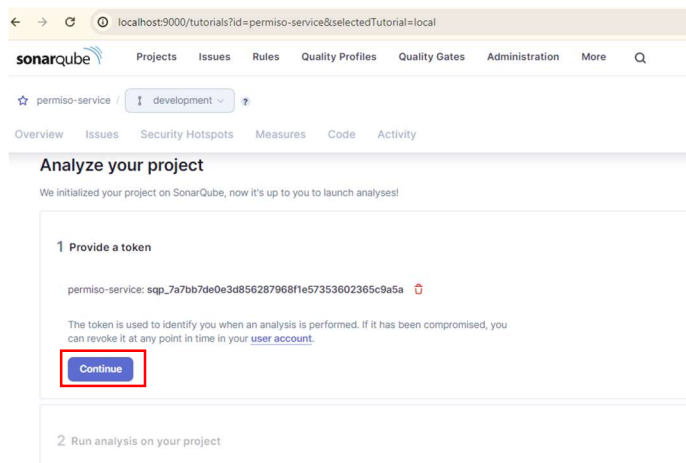
11. Seleccionar que se analizará el proyecto localmente:



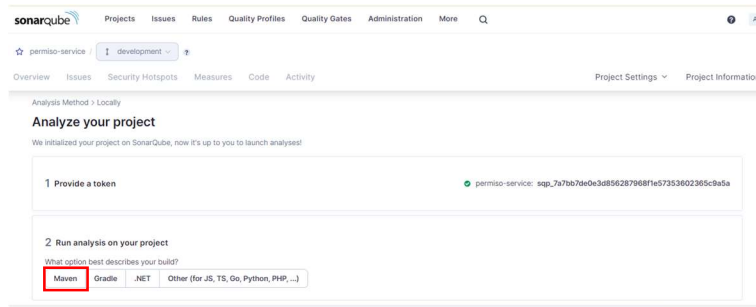
12. Proporcionar la descripción de el token para generar al comando maven y dar clic en el botón Generate:



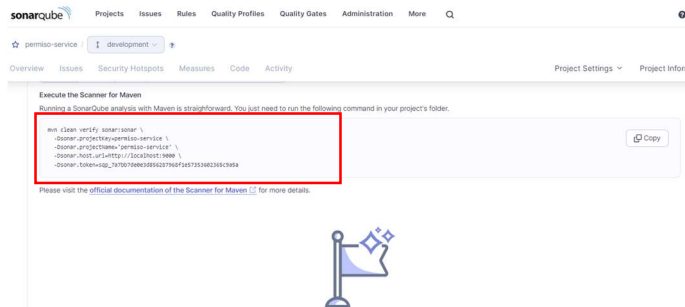
13. Después de que se generó el token dar clic en el botón Continue:



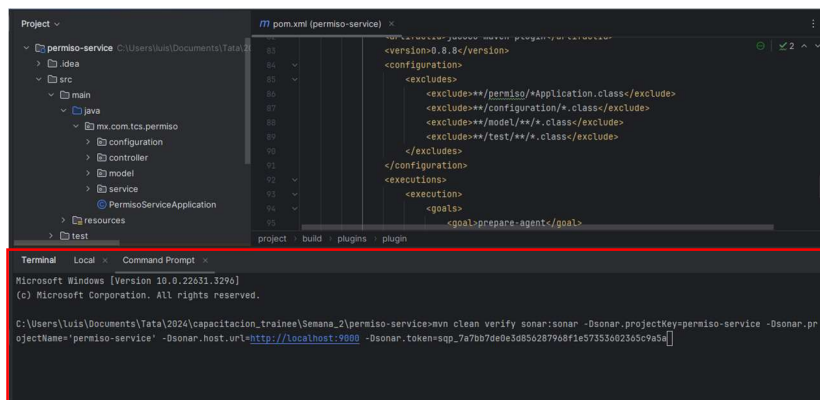
14. Dar clic en el tipo de construcción que se utiliza en el proyecto que se analizará, en este caso maven:



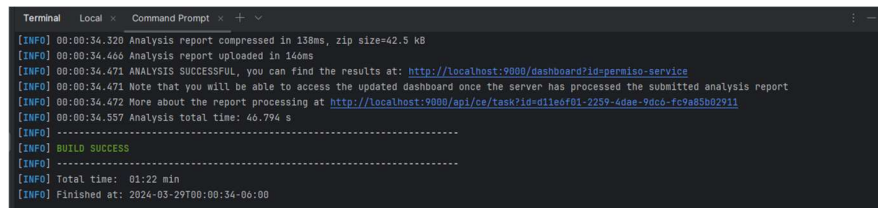
15. Revisar la instrucción Maven generada:



16. Desde la terminal cmd ejecutar la instrucción generada previamente para ligar el proyecto al análisis en sonarQube:

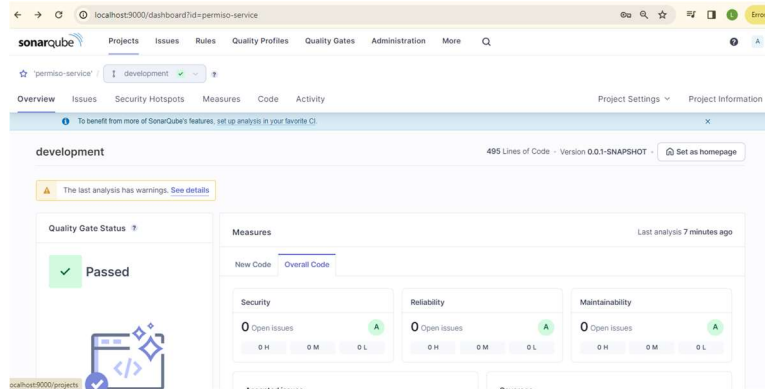


17. Validar que se ejecuto correctamente el comando y revisar el proyecto en el sitio de sonarQube:

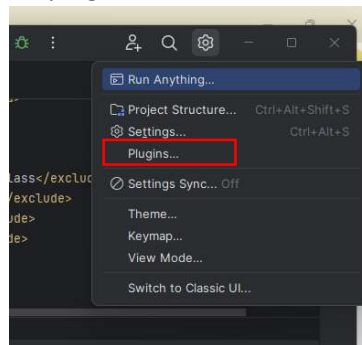


18. Ver la información actualizada:

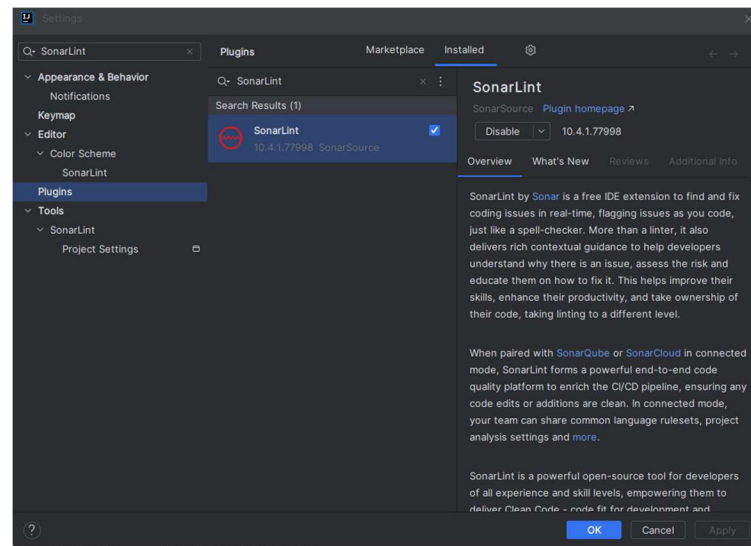




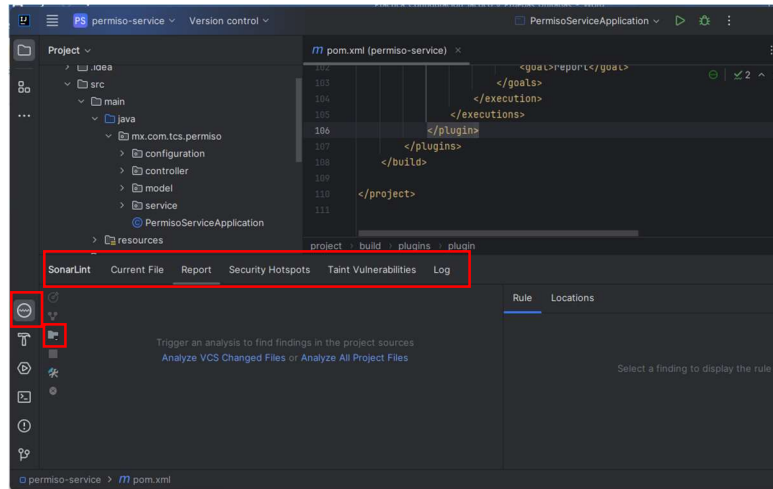
19. En el IDE IntelliJ abrir la opción de plugins:



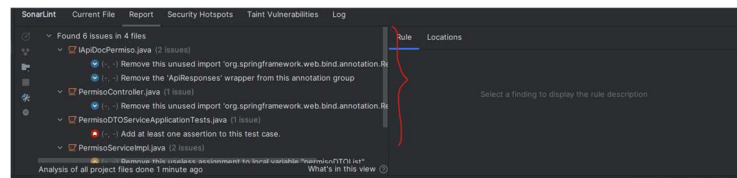
20. Buscar el plugin de SonarLint e instalarlo, en caso de no encontrarlo buscarlo y descargarlo desde el sitio: <https://plugins.jetbrains.com/plugin/7973-sonarlint> e instalarlo manualmente.



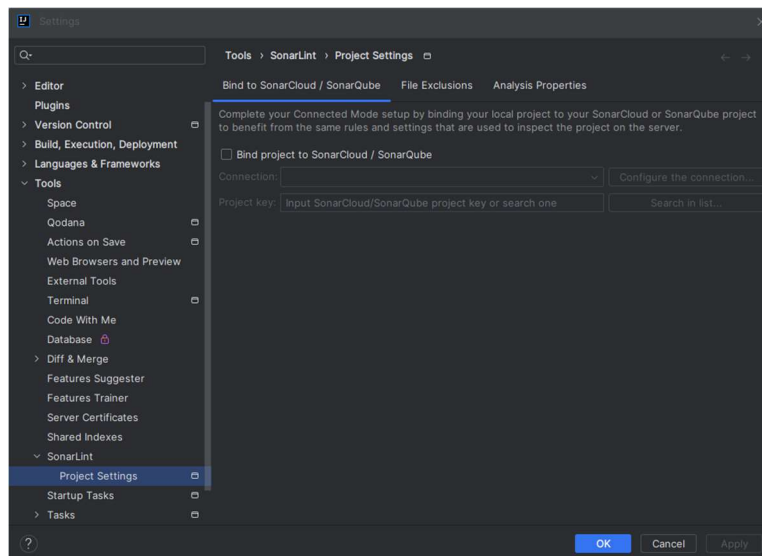
21. Abrir la opción de SonarLint y generar reporte al dar clic en el icono de carpeta:



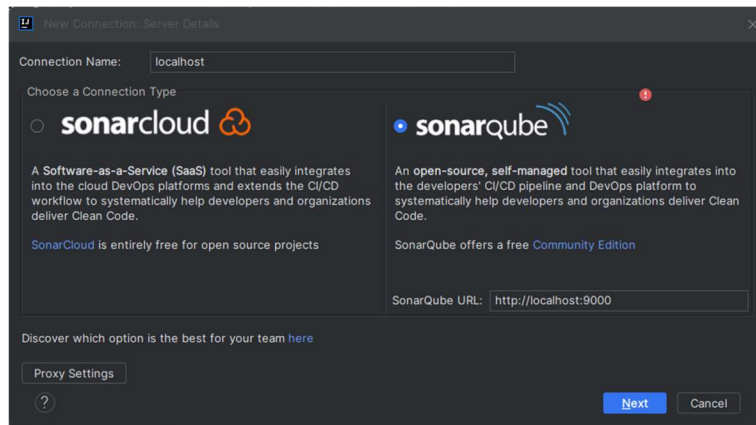
22. Revisar los errores encontrados y corregirlos.



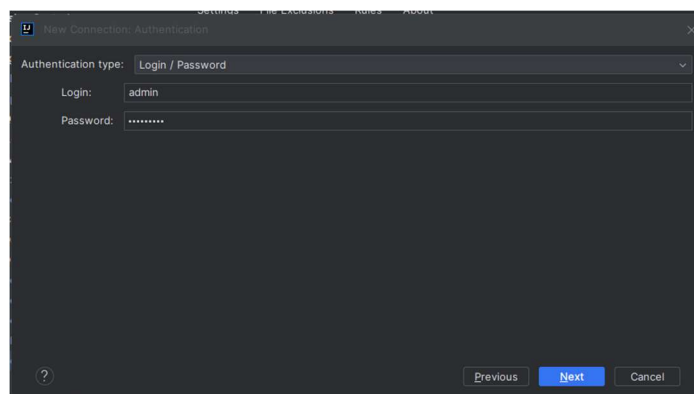
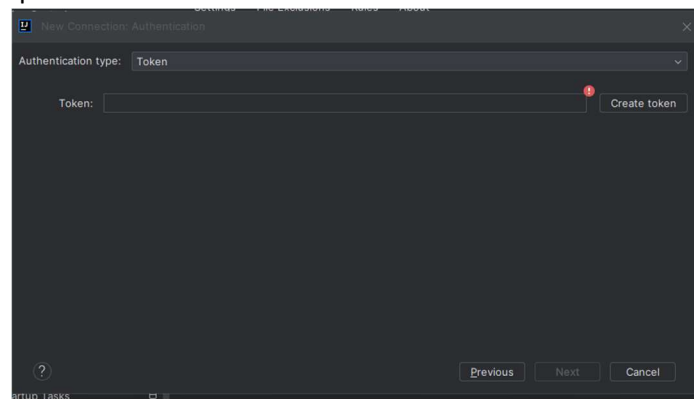
23. Configurar el servidor de Sonar en IntelliJ, seleccionar la opción de settings y buscar SonarLint, Project Settings:



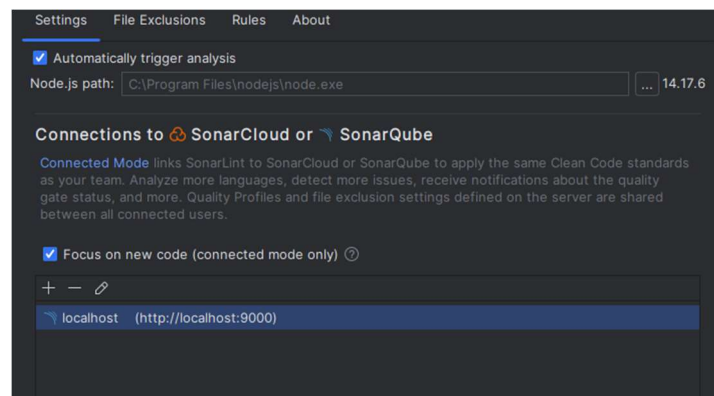
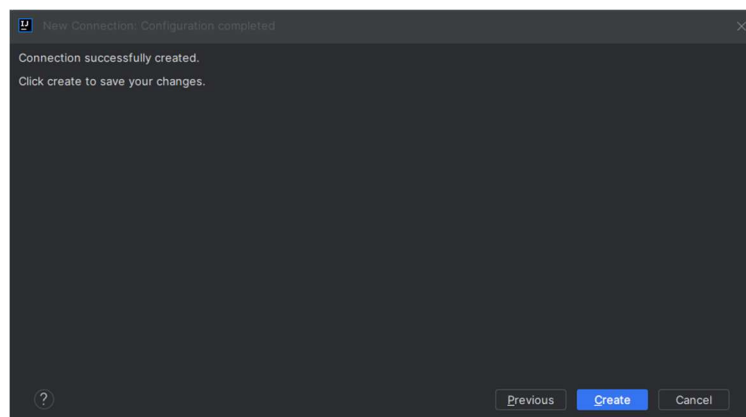
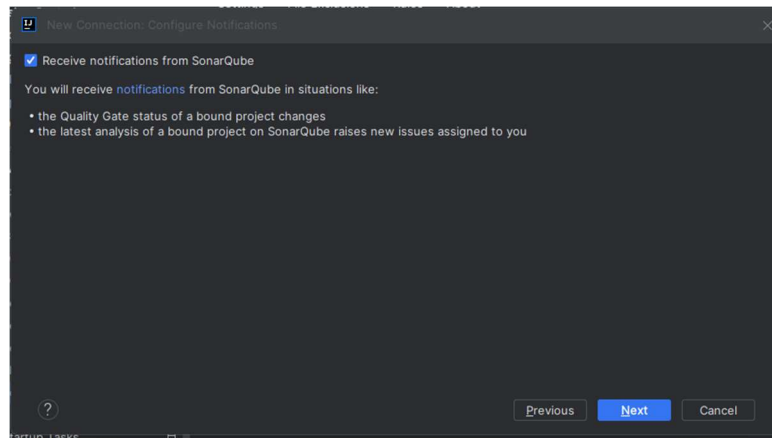
24. Generar una conexión



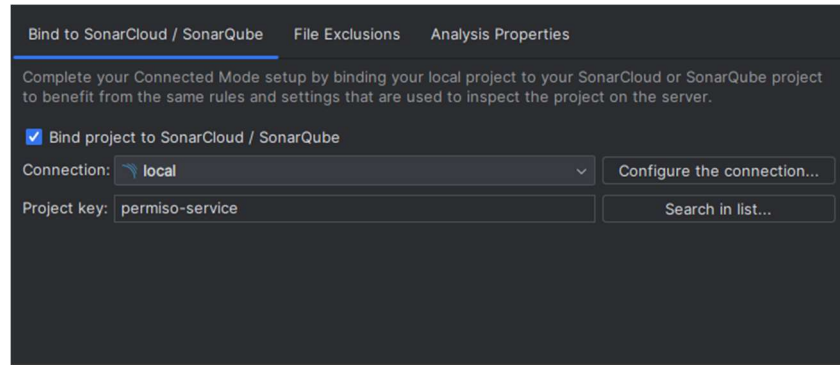
25. Seleccionar el tipo de autenticación:



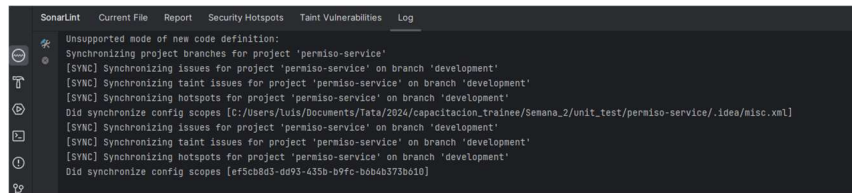
26. Seleccionar que reciban notificaciones:



27. Agregar los valores para la configuración seleccionando la conexión:

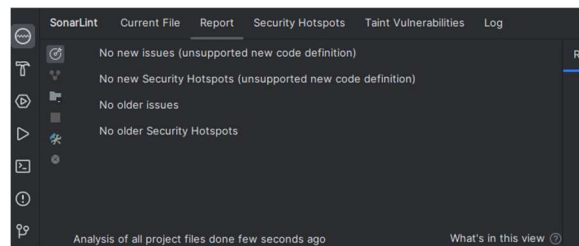


28. Validar el log después de agregar la configuración:

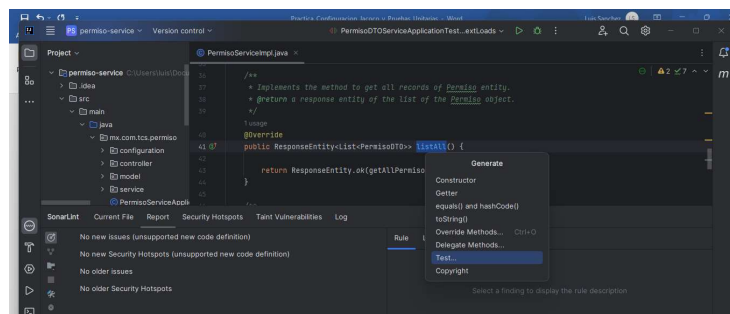


Agregar pruebas unitarias en proyecto permiso-service

1. Corregir errores detectados en la validación de sonar:



2. Agregar prueba unitaria a la clase ServiceImpl, en específico al método: listAll().



3. Agregar el arreglo para la prueba del método:

```

37     private List<Permiso> permisoList;
38     private PermisoDTO permisoDTO;
39
40     @BeforeEach
41     void setUp() {
42         permisoList = new ArrayList<>();
43
44         Permiso permiso = new Permiso();
45         permiso.setId(1);
46         permiso.setNombre("Academico");
47         permiso.setDescripcion("Presentacion de examen profesional");
48         permiso.setActivo(1);
49         permiso.setIdPadre(1);
50         permiso.setIcono("/images/icon_school.gif");
51
52         permisoList.add(permiso);
53
54         permisoDTO = new PermisoDTO();
55         permisoDTO.setId(1);
56         permisoDTO.setNombre("Academico");
57         permisoDTO.setDescripcion("Presentacion de examen profesional");
58         permisoDTO.setActivo(1);
59         permisoDTO.setIdPadre(1);
60         permisoDTO.setIcono("/images/icon_school.gif");

```

4. Agregar la sección de ejecución:

```

66     @DisplayName("Test listAll when result is OK")
67     @Test
68     void listAll() {
69         // Given
70         Mockito.when(repository.findAll()).thenReturn(permisoList);
71         Mockito.doNothing().when(mapper).map(Mockito.any(Permiso.class), Mockito.any(PermisoDTO.class));
72
73         // When
74         ResponseEntity<List<PermisoDTO>> permisoDtoList = service.listAll();
75     }
76 }

```

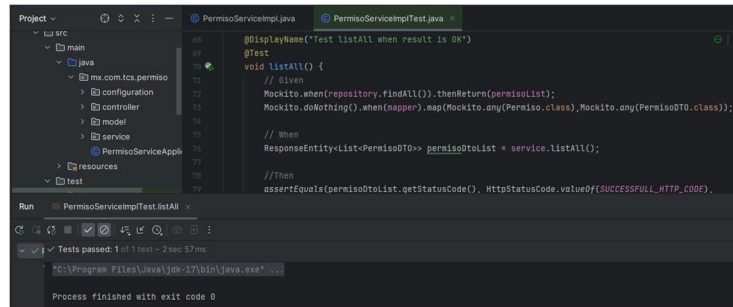
5. Agregar la sección de la evaluación:

```

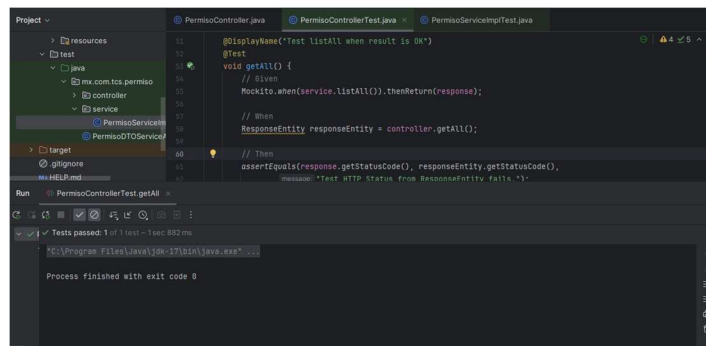
68     @DisplayName("Test listAll when result is OK")
69     @Test
70     void listAll() {
71         // Given
72         Mockito.when(repository.findAll()).thenReturn(permisoList);
73         Mockito.doNothing().when(mapper).map(Mockito.any(Permiso.class), Mockito.any(PermisoDTO.class));
74
75         // When
76         ResponseEntity<List<PermisoDTO>> permisoDtoList = service.listAll();
77
78         //Then
79         assertEquals(permisoDtoList.getStatusCode(), HttpStatusCode.valueOf(SUCCESSFUL_HTTP_CODE),
80             "message: 'Test StatusCode from ResponseEntity fails.'");
81         assertEquals(Objects.requireNonNull(permisoDtoList.getBody()).get(0).getId(), permisoDTO.getId(),
82             "message: 'Test id of first element from ResponseEntity fails.'");
83     }
84 }

```

6. Ejecutar la prueba unitaria para validar que es correcta su ejecución:



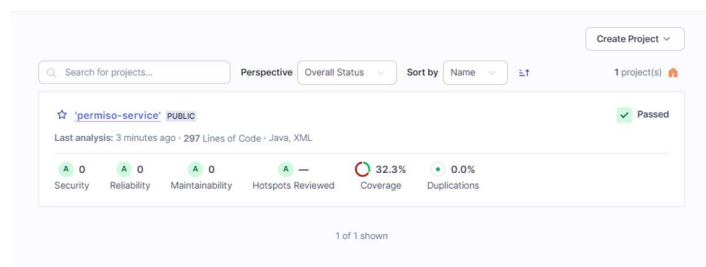
## 7. Crear prueba unitaria para el método del Controller:



## 8. Ejecutar el comando maven para actualizar a cobertura de pruebas unitarias en el código del proyecto:



## 9. Validar el resultado de la ejecución del comando maven:



## Validar la cobertura de proyecto permiso-service

1. Validar el resultado de la cobertura en el detalle del proyecto desplegado en sonar:
- 2.

3.

src/main/java/mx/com/tcs/permiso/service/PermisoServiceImpl.java 100% 0 100

4.

```
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60
```

```
public PermisoServiceImpl(PermisoRepository repository, IMapper mapper) {  
    this.repository = repository;  
    this.mapper = mapper;  
}  
  
/**  
 * Implements the method to get all records of Permiso entity.  
 * @return a response entity of the list of the Permiso object.  
 */  
@Override  
public ResponseEntity<List<PermisoDTO>> listAll() {  
    return ResponseEntity.ok(getAllPermiso());  
}  
  
/**  
 * Method to query from Permiso entity.  
 * @return a list of Permiso object.  
 */  
private List<PermisoDTO> getAllPermiso() {  
    List<Permiso> permisoList = (List<Permiso>) repository.findAll();  
  
    return permisoList.  
        stream().  
        map(permiso -> mapper.map(permiso, PermisoDTO.class)).  
        toList();  
}
```

## Referencias

- <https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner-for-maven/>
- <https://docs.sonarsource.com/sonarqube/latest/user-guide/concepts/>
- <https://plugins.jetbrains.com/plugin/7973-sonarlint>