



Herramientas de Calidad del Producto Software y Documentación.

SonarQube, Maven y Doxygen: Laboratorio de Desarrollo y Herramientas

Paula Regalado de León - alu0101330174@ull.edu.es



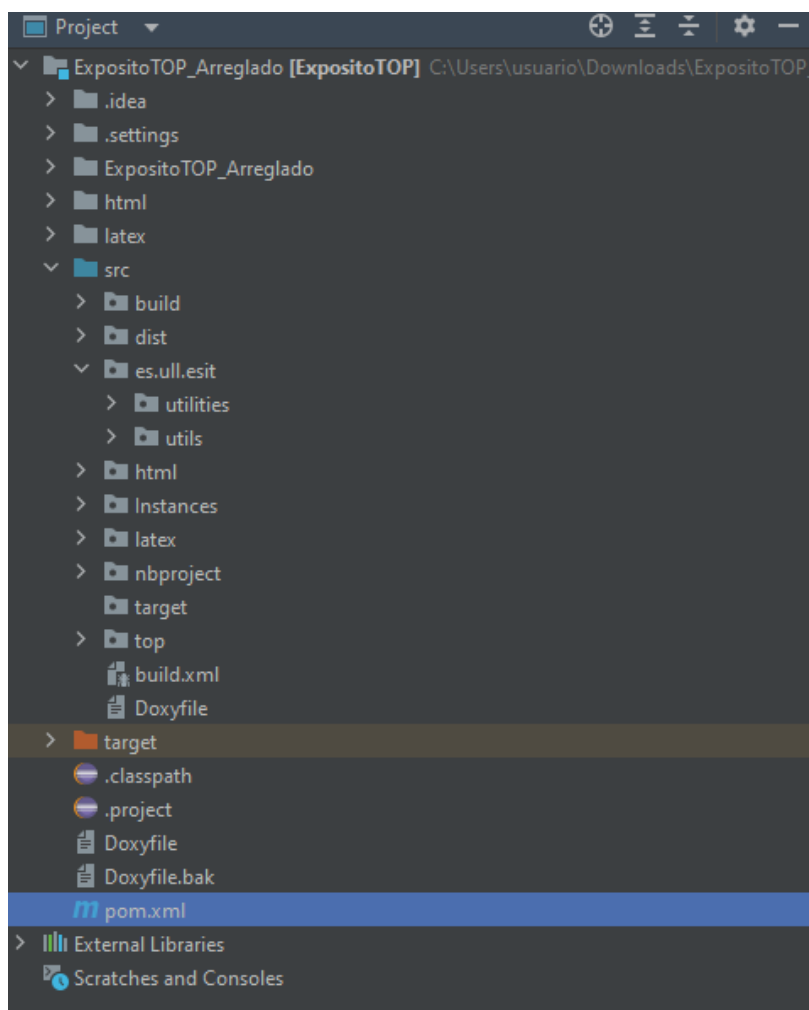
1. Instalación Maven.	2
2. Principales órdenes de Maven.	4
3. SonarQube	5
4. Doxygen.	9



1. Instalación Maven.

Una vez instalado el Maven, abres tu proyecto y añades un fichero llamado pom.xml donde añadirás plugins a elegir. El archivo pom.xml (Project Object Model) en Maven es fundamental para la construcción y gestión de proyectos Java (y otros tipos de proyectos) con Maven. Contiene información esencial sobre el proyecto y sus dependencias, y permite a Maven administrar el ciclo de vida del proyecto. Puedes configurar plugins de Maven en el pom.xml para personalizar y extender las funcionalidades de Maven durante el proceso de construcción.

Esto permite a Maven automatizar y simplificar la construcción y gestión de proyectos de desarrollo de software.





```
C:\Windows\System32\cmd.exe

[INFO] Finished NVD CVE Analyzer (0 seconds)
[INFO] Finished Sonatype OSS Index Analyzer (1 seconds)
[INFO] Finished Vulnerability Suppression Analyzer (0 seconds)
[INFO] Finished Dependency Bundling Analyzer (0 seconds)
[INFO] Analysis Complete (4 seconds)
[WARNING]

One or more dependencies were identified with known vulnerabilities in ExpositoTOP:

log4j-1.2.14.jar (pkg:maven/log4j/log4j@1.2.14, cpe:2.3:a:apache:log4j:1.2.14:*:*:*:*:*:*): CVE-2019-17571, CVE-2020-
9493, CVE-2021-4104, CVE-2022-23302, CVE-2022-23305, CVE-2022-23307, CVE-2023-26464

See the dependency-check report for more details.

[INFO] Generating "Dependencies" report --- maven-project-info-reports-plugin:3.1.1:dependencies
[INFO] Generating "Dependency Information" report --- maven-project-info-reports-plugin:3.1.1:dependency-info
[INFO] Generating "About" report --- maven-project-info-reports-plugin:3.1.1:index
[INFO] Generating "Plugin Management" report --- maven-project-info-reports-plugin:3.1.1:plugin-management
[INFO] Generating "Plugins" report --- maven-project-info-reports-plugin:3.1.1:plugins
[INFO] Generating "Summary" report --- maven-project-info-reports-plugin:3.1.1:summary
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 07:07 min
[INFO] Finished at: 2023-10-29T16:47:01Z
[INFO] -----
```

Imagen ejecución desde terminal

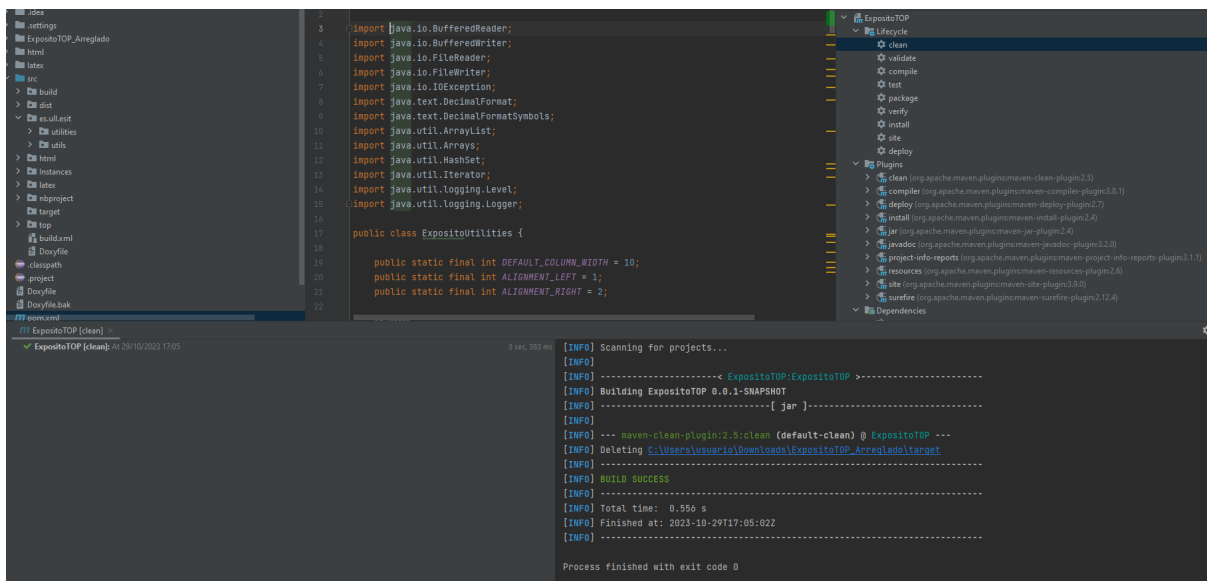
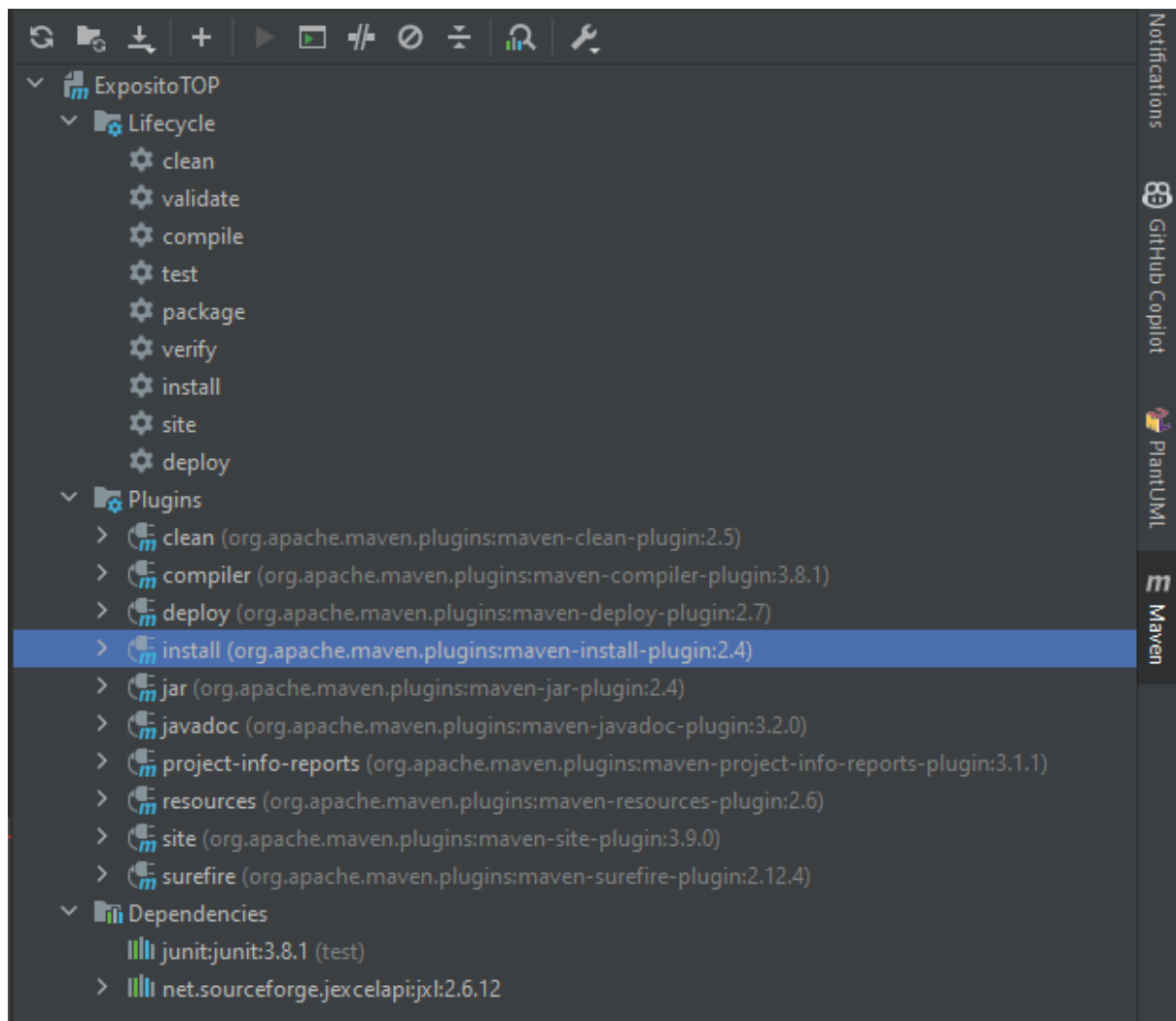


Imagen ejecución desde el IDE (IntelliJ)



2. Principales órdenes de Maven.

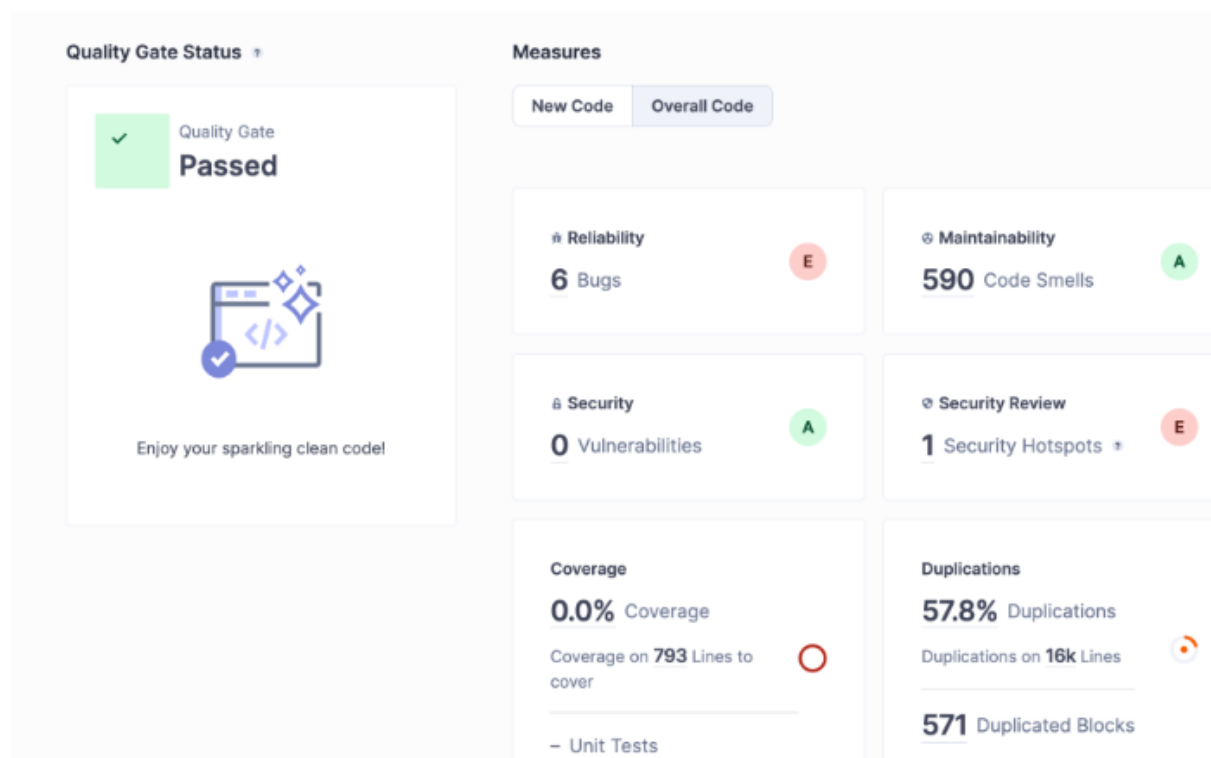
1. **mvn clean:** Limpia el directorio target del proyecto, eliminando archivos compilados y generados anteriormente.
2. **mvn compile:** Compila el código fuente del proyecto.
3. **mvn test:** Ejecuta las pruebas unitarias del proyecto.
4. **mvn package:** Empaqueta el proyecto en un formato específico (por ejemplo, JAR, WAR).
5. **mvn install:** Instala el artefacto en el repositorio local para su uso en otros proyectos.
6. **mvn dependency:resolve:** Resuelve y descarga las dependencias del proyecto.
7. **mvn site:** Genera informes y documentación del proyecto.





3. SonarQube y errores.

SonarQube es una plataforma de código abierto que se utiliza principalmente para evaluar y mejorar la calidad del código fuente en proyectos de desarrollo de software. Su funcionalidad principal es proporcionar una serie de herramientas y características destinadas a identificar y corregir problemas de calidad de código.





Error 1:

Intentionality issue

☐ [A "NullPointerException" could be thrown; "reader" is nullable here.](#) cert cwe ... +

☐ Open

☐ Not assigned

☒ Reliability

☐ Bug

☐ Major

El mensaje "Se puede lanzar un 'NullPointerException' ya que 'reader' es nulo" indica que en tu código se está intentando acceder a un objeto llamado "reader," pero dicho objeto no ha sido inicializado y, por lo tanto, es nulo. Cuando intentas realizar operaciones en un objeto nulo, se arroja una excepción de tipo `NullPointerException`, lo que puede provocar que tu programa falle.

Para solucionar este problema, debes garantizar que el objeto "reader" se inicialice correctamente antes de intentar usarlo.

Error 2:

Intentionality issue

☐ [Use try-with-resources or close this "BufferedWriter" in a "finally" clause.](#) cert cwe ... +

☐ Open

☐ Not assigned

☒ Reliability

☐ Bug

☐ Blocker

El mensaje de advertencia "use try-with-resources or close this `BufferedReader` in a finally clause" te sugiere que debes asegurarte de cerrar el objeto `BufferedReader` de manera adecuada después de usarlo. Esto se debe a que los objetos que implementan la interfaz `Closeable`, como `BufferedReader`, deben cerrarse para liberar recursos y evitar problemas de memoria.

Error 3:

Consistency issue

☐ [Add a "NoSuchElementException" for iteration beyond the end of the collection.](#) error-handling unpredictable +

☐ Open

☐ Not assigned

☒ Reliability

☐ Bug

☐ Minor

5min effort · 29 minutes ago



El mensaje "add a NoSuchElementException for iteration beyond the end of the collection" es una sugerencia para mejorar la gestión de excepciones cuando se itera más allá del final de una colección. Indica que deberías considerar lanzar una excepción NoSuchElementException cuando se intenta acceder a un elemento que no existe en la colección durante un proceso de iteración.

Error 4:

Intentionality issue

☐ [Refactor this code so that the Iterator supports multiple traversal](#)

pitfall +

☐ Open ▾

☐ Not assigned ▾

☒ Reliability ⚠

☐ Bug ⚠

☐ Major ⚠

El mensaje "refactor this code so that the iterator supports multiple traversal" significa que el código actual solo permite recorrer la colección una vez utilizando el iterador. Para hacer que el iterador sea reutilizable y admita múltiples recorridos, debes realizar una refactorización del código.

La solución implica modificar la implementación del iterador para que pueda volver al principio de la colección o reiniciarse para permitir múltiples recorridos. Esto se logra al mantener un estado interno que controla la posición actual del iterador y proporcionar un método para reiniciar o restablecer ese estado.

Error 5:

Intentionality issue

☐ [Save and re-use this "Random".](#)

No tags +

☐ Open ▾

☐ Not assigned ▾

☒ Reliability ⚠

☐ Bug ⚠

☐ Critical ⚠

Puede ser ineficiente y llevar a resultados no verdaderamente aleatorios el crear y usar múltiples instancias de "Random".



Error 6:

Intentionality issue

☐ [Either override `Object.equals\(Object\)`, or rename the method to prevent any confusion.](#)

suspicious +

☐ Open ▾ ☐ Not assigned ▾ ☒ Reliability ⚠ ☐ Bug ⚠ ☐ Major ⚠

El mensaje "sobrescribir `Object.equals(Object)` o renombrar el método para evitar confusiones" te está indicando que en tu código hay un método que podría estar causando confusiones en relación a la convención de nombres o la implementación de la comparación de igualdad.

Para solucionar este problema, puedes tomar una de las siguientes dos acciones:

Sobrescribir el método `equals(Object)`: Esto significa que debes proporcionar tu propia implementación del método `equals` en una clase que esté destinada a ser comparada con otra instancia de la misma clase.

Renombrar el método: Si no deseas proporcionar tu propia implementación de `equals`, puedes considerar cambiar el nombre del método para evitar confusiones.

4. Doxygen.

Doxygen es una herramienta versátil que simplifica la creación de documentación de código fuente, lo que facilita la comprensión y el mantenimiento del software. Para generar esta documentación se deberán usar estos comandos.

```
doxygen -g Doxyfile
doxygen Doxyfile
```

La documentación se encontrará en el fichero `Doxyfile`.



```
Doxyfile
36
37 DOXYFILE_ENCODING      = UTF-8
38
39 # The PROJECT_NAME tag is a single word (or a sequence of words surrounded by
40 # double-quotes, unless you are using Doxywizard) that should identify the
41 # project for which the documentation is generated. This name is used in the
42 # title of most generated pages and in a few other places.
43 # The default value is: My Project.
44
45 PROJECT_NAME           = "My Project"
46
47 # The PROJECT_NUMBER tag can be used to enter a project or revision number. This
48 # could be handy for archiving the generated documentation or if some version
49 # control system is used.
50
51 PROJECT_NUMBER         =
52
53 # Using the PROJECT_BRIEF tag one can provide an optional one line description
54 # for a project that appears at the top of each page and should give viewer a
55 # quick idea about the purpose of the project. Keep the description short.
56
57 PROJECT_BRIEF          =|
58
59 # With the PROJECT_LOGO tag one can specify a logo or an icon that is included
60 # in the documentation. The maximum height of the logo should not exceed 55
61 # pixels and the maximum width should not exceed 200 pixels. Doxygen will copy
62 # the logo to the output directory.
63
64 PROJECT_LOGO           =
65
66 # The OUTPUT_DIRECTORY tag is used to specify the (relative or absolute) path
67 # into which the generated documentation will be written. If a relative path is
68 # entered, it will be relative to the location where doxygen was started. If
69 # left blank the current directory will be used.
70
71 OUTPUT_DIRECTORY       =
```