## Realización de comandos básicos en la consola

Mediante la línea de comandos se han ejecutado varias funcionalidades de maven relacionadas con el proyecto como son Package, Validate, Compile, Test e Install. A continuación se añaden unas capturas del correcto funcionamiento de los mismos.

## Package

### Validate

```
| alicia@LAPTOP-TSVBF43T:/mnt/c/Users/alici/Downloads/LHD_P3/ExpositoTOP$ mvn validate
| [INFO] | Scanning for projects...
| [INFO] | Building ExpositoTOP 1.0-SNAPSHOT
| [INFO] | Suilding ExpositoTOP 1.0-SNAPSH
```

### Compile

#### Test

```
licia@LAPTOP-TSVBF43T:/mnt/c/Users/alici/Downloads/LHD_P3/ExpositoTOP$    mvn test
[INFO] Scanning for projects...
[INFO] ------ groupId:ExpositoTOP >-----
[INFO] Building ExpositoTOP 1.0-SNAPSHOT
[INFO] ------[ jar ]------
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ ExpositoTOP ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ ExpositoTOP ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ ExpositoTOP ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /mnt/c/Users/alici/Downloads/LHD_P3/ExpositoTOP/src/test/resources
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ ExpositoTOP ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ ExpositoTOP ---
[INFO] No tests to run.
[INFO] BUILD SUCCESS
```

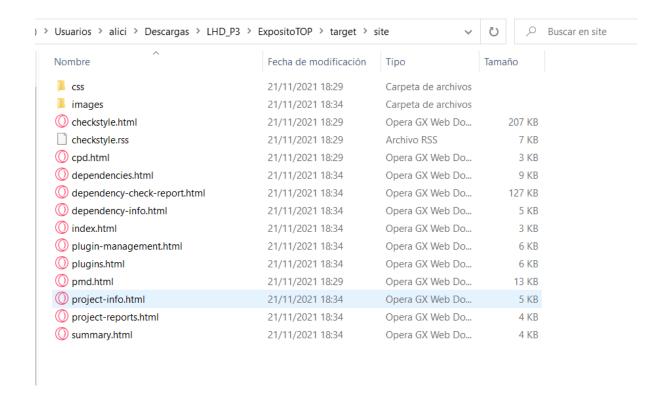
### Install

# Plugins de Maven

Maven funciona a partir de un fichero pom el cual podemos modificar para añadirle plugins. En este caso hemos añadido el plugin de CheckStyle, el cual se ejecuta con el comando:

mvn site

Este plugin produce un directorio "site" que contiene los archivos HTML que describen el proyecto



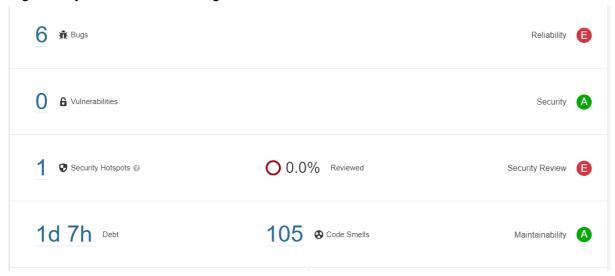
Por otro lado está el plugin de OWASP, el cual es un verificador de dependencias entre los archivos del proyecto. Este monta el .jar del proyecto una vez verificado el mismo

La principal ventaja de este plugin es que permite identificar cuales son los componentes más vulnerables.

## Análisis y correcciones

Una vez nuestro proyecto está soportado por Maven y contiene unos plugins podemos utilizar SonnarScanner para verificar el estado del código.

Con el primer análisis obtenemos este resultado donde contamos con 6 bugs, 1 fallo de seguridad y 105 errores de código.



Partiendo de este punto se han arreglado varios code smells, el error de seguridad y alguno de los bugs.

Se incluye una imagen de un ejemplo de code smell solucionado

## Documentación con Doxygen

El último paso consiste en la documentación del proyecto, para ello empleamos Doxygen. Más concretamente el "Doxygen GUI frontend +", esta herramienta nos facilita el procedimiento.

Para poder emplear doxygen debemos emplear comentarios en el código fuente, así y especificando ciertos detalles en el asistente obtendremos unos archivos html que nos proporciona la información del proyecto.

Se adjunta a continuación una imagen de dos métodos del proyecto obtenida de la documentación del doxygen.

Documentación de las funciones miembro

