LINT REPORT

Grupo E7.02

Álvaro Úbeda Ruiz (alvuberui@alum.us.es)

Mario Pérez Coronel (marpercor8@alum.us.es)

Carlos Garrido Rodríguez (cargarrod12@alum.us.es)

Ramón Rodríguez Bejarano (ramrodbej@alum.us.es)

Mario Rodríguez García (marpercor8@alum.us.es)

Juan Carlos Gómez Rodríguez (juagomram4@alum.us.es)

Repositorio: https://github.com/alvuberui/Acme-Toolkits

Tablero: https://github.com/users/alvuberui/projects/4

25 DE MARZO DE 2022

Contenido

1. Resumen ejecutivo

En este documento se hará uso de la herramienta SonarLint, que previamente se instaló con el workspace proporcionado. Con esta herramienta analizaremos el proyecto de *Acme-Toolkits* y comprobaremos los bad smells que existen en el proyecto. La idea del documento es explicar el cambio que necesitaremos para que estos errores no aparezcan.

Una vez refactorizado el proyecto analizar de nuevo el proyecto para estar conforme de que no queda ningún bad smells anterior o, uno nuevo creado al refactorizar.

2. Tabla de revisión

Núm. Revisión	Fecha	Descripción
1	23/05/2022	Creación del documento

3. Introducción

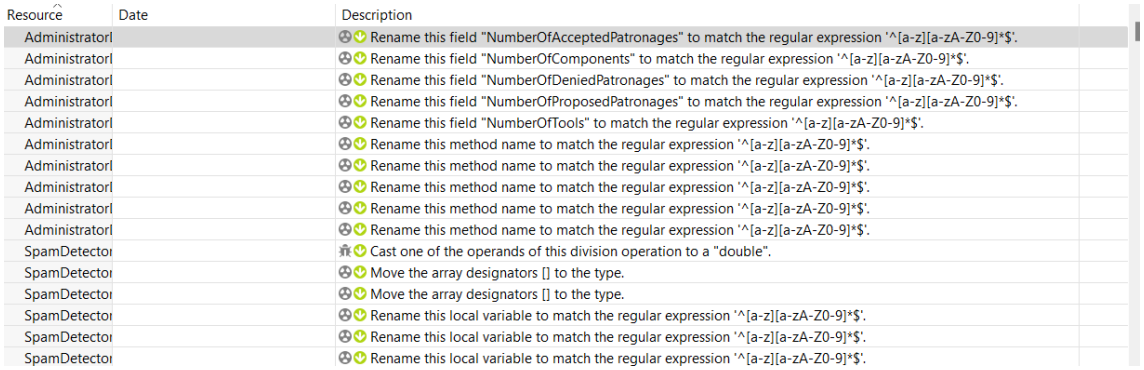
Esta herramienta, SonarLint, analiza nuestro proyecto en busca de *bad smells*. También es utilizada a lo largo del uso de Eclipse ya que mientras escribes el código te proporciona información sobre errores que cometes o bugs que estás creando.
Además, esta herramienta te da las sugerencias oportunas para corregir estos fallos y explica el porqué de cada uno.

4. Contenido

4.1 SonarLint

En este apartado se mostrará la primera vez que SonarLint ha examinado el proyecto justo después de finalizar las tareas oportunas para este Sprint.

A continuación, los diferentes errores que nos muestra la herramienta:



Resource	Date	Description
Administratorl		Rename this field "NumberOfAcceptedPatronages" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
Administratorl		Rename this field "NumberOfComponents" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
Administratorl		Rename this field "NumberOfDeniedPatronages" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
Administratorl		Rename this field "NumberOfProposedPatronages" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
Administratorl		Rename this field "NumberOfTools" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
Administratorl		Rename this method name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
Administratorl		Rename this method name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
Administratorl		Rename this method name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
Administratorl		Rename this method name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
Administratorl		Rename this method name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
SpamDetector		Cast one of the operands of this division operation to a "double".
SpamDetector		Move the array designators [] to the type.
SpamDetector		Move the array designators [] to the type.
SpamDetector		Rename this local variable to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
SpamDetector		Rename this local variable to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
SpamDetector		Rename this local variable to match the regular expression '^[a-z][a-zA-Z0-9]*$'.

Tenemos un total de 16 bad smells.

Para comenzar, tenemos 5 variables inicializadas con letras en mayúsculas cuando debería comenzar en minúsculas.

Estas 3 son un ejemplo de cómo NO deben estar.

```
int NumberOfProposedPatronages;

int NumberOfAcceptedPatronages;

int NumberOfDeniedPatronages;
```

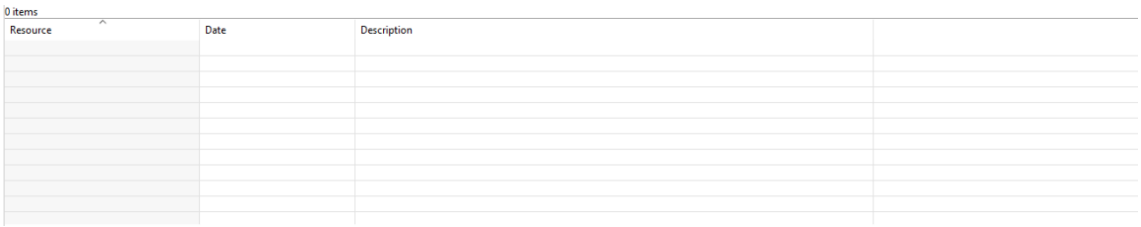
Y así deben de quedar

```
int numberOfProposedPatronages;

int numberOfAcceptedPatronages;

int numberOfDeniedPatronages;
```

También hemos creado una variable que tiene el mismo nombre que una de las entradas del método, por ello lo hemos cambiado. También hemos movido los corchetes que asignan el array junto al tipo de la variable y hemos puesto (doublé) en la variable porcentageWeakTerms.

```
private static boolean spamTerms(final String words, final String terms, Double Threshold) {
    final String[] wordsSplit = terms.split(",");
    final String[] inputWordsSplits = words.split("[ ,.:;/()]");
    int k = findWordsRecursive(0, 0, wordsSplit, inputWordsSplits);
    final double porcentageWeakTerms = (double) (k*100)/wordsSplit.length;
```

4.2 SonarLint Corrección



En este apartado se muestra el *Report* generado por SonarLint después de corregir los errores que pueden ser corregidos.

Con lo que con las soluciones que hemos dicho anteriormente quedan solucionados los bad smells del proyecto.

5. Conclusión

La herramienta SonarLint nos proporciona mucha información durante el proceso de programación de cada feature. Debido a esto los errores del proyecto eran mínimos, pues se van corrigiendo a medida que avanzamos.

6. Bibliografía

Intencionadamente en blanco.