



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 (marrodgar62@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/3>

25 DE MARZO DE 2022

Contenido

- 1. Resumen ejecutivo2
- 2. Tabla de revisión.....2
- 3. Introducción.....3
- 4. Contenido3
 - 4.1 SonarLint3
 - 4.2 SonarLint Corrección4
- 5. Conclusión4
- 6. Bibliografía5

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	24/04/2022	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:

Items		
Resource	Date	Description
form.jsp	few seconds ago	Add a description to this table.
form.jsp	few seconds ago	Add a description to this table.
form.jsp	few seconds ago	Add a description to this table.
form.jsp	few seconds ago	Add a description to this table.
form.jsp	few seconds ago	Add a description to this table.
InventorToolkitsRepository.java		Rename this method name to match the regular expression <code>^[a-z][a-zA-Z0-9]*\$</code> .
AuthenticatedToolKitShowService.java	few seconds ago	A <code>NullPointerException</code> could be thrown: <code>systemConfiguration</code> is nullable here. [+2 locations]

En este caso tenemos sólo dos bad smells.

Por un lado, un valor que puede llegar a ser *null* y provocar un *Null Exception*, para solucionarlos hemos creado la condición que faltaba para comprobar si era *null*.

```
if(!artefacts.isEmpty() && systemConfiguration != null) {
    final Double price = artefacts.stream().map(x -> this.computePrice(x)).findFirst().orElse(0.0);
    final Money money = new Money();
    money.setAmount(price);
    money.setCurrency(systemConfiguration.getCurrency());
    model.setAttribute("price",money);
}else {
    final Money money = new Money();
    money.setAmount(0.0);
    money.setCurrency(systemConfiguration.getCurrency());
    model.setAttribute("price",money);
}
```

```

model.addAttribute("price", money);
}
}

if(systemConfiguration != null && systemConfiguration.getCurrency() != null) {
    if(!artefacts.isEmpty()) {
        final Double price = artefacts.stream().map(x -> this.computeMoneyExchange(x)).reduce(0.0, Double::sum);
        final Money money = new Money();
        money.setAmount(price);
        money.setCurrency(systemConfiguration.getCurrency());
        model.addAttribute("price", money);
    }
    else {
        final Money money = new Money();
        money.setAmount(0.0);
        money.setCurrency(systemConfiguration.getCurrency());
        model.addAttribute("price", money);
    }
}
}

```

Y por otro lado tenemos un método que no cumple con camelCase ya que empezaba en mayúsculas. Simplemente se ha cambiado el nombre.

```

@Query("select distinct a from Toolkit a join Quan b on a.id = b.toolkitId")
Collection<Artefact> ArtefactByToolkitId(int id);

Collection<Artefact> artefactByToolkitId(int id);

```

Y por último tenemos distintas tablas dentro de un *form* que necesitan una descripción. Esto se ha solucionado añadiendo descripciones con el `aria-describedby=""` con su propia descripción.

```

<table class="table table-sm">
    <tr>
        <td><input type="text" value="" /></td>
        <td><input type="text" value="" /></td>
        <td><input type="text" value="" /></td>
    </tr>
</table>

s="table table-sm" aria-describedby="numberPatronages">

```

4.2 SonarLint Corrección

0 items		
Resource	Date	Description

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.