

# Universidad de Sevilla

Ingeniería Informática. Ingeniería del Software

## DO3 – Displaying data

### R21 – Lint Report

Fecha	Versión
02/09/2022	0.1

Enlace al repositorio: <https://github.com/raumonmar1/Acme-Recipes.git>

Grupo de prácticas	Septiembre #3
<i><b>Autores</b></i>	<i><b>Roles</b></i>
<i>Raúl Montalbán Martín raumonmar1@alum.us.es</i>	<i>Manager, Developer and Tester</i>
<i>Daniel Suárez Perea dansuaper@alum.us.es</i>	<i>Developer and Tester</i>

## Tabla de contenido

1. Resumen ejecutivo
2. Tabla de versiones
3. Introducción
4. Contenido
5. Conclusión
6. Bibliografía

### 1. Resumen ejecutivo.

En este documento describiremos los resultados del informe Lint sobre la calidad del código, así como las mejoras introducidas en este para garantizar el correcto funcionamiento del mismo en cuanto a bugs de implementación se refiere.

### 2. Tabla de versiones

Versión	Fecha	Descripción
0.1	02/09/2022	Creación del documento
0.2	03/09/2022	Realización del informe

### 3. Introducción

Comenzaremos exponiendo los resultados del informe realizado para nuestro proyecto actual, para continuar con las mejoras necesarias que se han tenido que realizar para modificar las sugerencias indicadas por SonarLint. Por último, se añaden las conclusiones del informe.

### 4. Contenido

El informe inicial de SonarLint nos refleja 6 malos olores, que se detallan a continuación:

Resource	Date	Description
AdministratorD		⚠️📌 Convert this Map to an EnumMap.
AdministratorD		⚠️📌 Remove this unused "e" local variable.
EpicureDashbo		⚠️📌 Convert this Map to an EnumMap.
EpicureDashbo		⚠️📌 Remove this unused "e" local variable.
form.jsp		🔗📌 Add a description to this table.
form.jsp		🔗📌 Add a description to this table.

- Convertir el Map en un EnumMap.

```
50    final Map<FineDishStatus, Integer> totalNumberOfFineDishesByStatus = new HashMap<>();
```

Para arreglar este bug, se inicializa la variable como un EnumMap.

```
51     final EnumMap<FineDishStatus, Integer> totalNumberOfFineDishesByStatus = new EnumMap<>(FineDishStatus.class);
```

- Remover la variable e sin utilizar.

```
for(final Tuple e: this.repository.totalNumberOfFineDishesByStatus()) {
    this.repository.totalNumberOfFineDishesByStatus().stream()
        .forEach(x -> totalNumberOfFineDishesByStatus.put((FineDishStatus) x.get(0), Integer.parseInt(x.get(1).toString())));
}
```

Para arreglar este bug, se procede a cambiar la estructura del bucle for.

```
103     for(int i = 0; i < this.repository.totalNumberOfFineDishesByStatus().size(); i++) {
104         this.repository.totalNumberOfFineDishesByStatus().stream()
105             .forEach(x -> totalNumberOfFineDishesByStatus.put((FineDishStatus) x.get(0), Integer.parseInt(x.get(1).toString())));
106     }
```

- Añadir descripción a la tabla.

```
<table class="table table-sm">
```

Para arreglar este bug, se añade el campo `aria-describedby` en la etiqueta de creación de la tabla.

```
<table class="table table-sm" aria-describedby="Dashboard">
```

Estas modificaciones se han realizado tanto en los archivos pertenecientes al AdministratorDashboard como en los archivos pertenecientes al EpicureDashboard.

El resultado final del análisis SonarLint nos muestra un código sin presencia de malos olores.

## **5. Conclusión**

Una vez realizado el análisis con SonarLint, podemos asegurar que el código de nuestro proyecto se encuentra libre de malos olores, lo que significa que hay un mayor nivel de confianza en cuanto a la ausencia de fallos en el mismo. No obstante, este informe es sólo una herramienta más a la hora de intentar garantizar el correcto funcionamiento de la aplicación, y se debe combinar con una buena estrategia de testing de nuestro código para garantizar un resultado exitoso.

## **6. Bibliografía**

Intencionadamente en blanco.