

DP2 2021-2022

“Lint REPORT”



Grupo E1.02:

- Juan García Oliva (juagaroli@alum.us.es)
- María José Ruíz Vázquez (marruivaz1@alum.us.es)
- Álvaro Paradas Borrego (alvparbor1@alum.us.es)
- José Antonio Grande Romero (josgrarom@alum.us.es)
- Pablo Rodríguez Gómez (pabrodgom5@alum.us.es)

Repositorio: <https://github.com/juagaroli/Acme-Toolkits.git>

Proyecto: <https://github.com/users/juagaroli/projects/1>

Fecha: 25/04/2022

1. Índice

1. Índice	2
2. Resumen ejecutivo	2
3. Tabla de revisión	2
4. Introducción	3
5. Contenidos	3
5.1. D03	3
5.2. D04	6
6. Conclusiones	7
6.1. D03	7
6.2. D04	7
7. Bibliografía	7

2. Resumen ejecutivo

Vamos a realizar un análisis en profundidad del código en busca de bugs y malos olores, con el fin de obtener un código óptimo y limpio.

3. Tabla de revisión

Tabla de versionado, donde se detallan las distintas versiones del documento, explicando las nuevas incorporaciones o cambios realizados.

Versión	Fecha	Descripción de cambios
1.0	25/04/2022	Versión inicial del documento
1.1	15/05/22	Insertados los distintos entregables
2.0	22/05/22	Versión final D04

4. Introducción

A continuación, detallaremos un informe donde veremos los distintos tipos de malos olores y bugs en nuestro código durante el desarrollo del sprint para obtener un análisis de cómo ha ido progresando el proyecto con respecto a la calidad del código.

Dichos análisis los obtenemos a partir de la extensión de sonarLint que nos produce un resumen del estado de nuestro código.



























5. Contenidos

5.1. D03

El 11/04/2022 realizamos un análisis de sonarLint en el que nos dieron los siguientes errores:

- 7 malos olores
- 6 bugs

Este análisis se realizó cuando solo estaban el 30% de las tareas completas, por lo que es normal que tuviésemos tan pocos errores.

Resource	Date	Description
Administrato		  Remove this unused "averageRetailPriceOfComponentsByTechnologyAndCurrency" local variable.
Authenticate		  Remove this unused import 'org.springframework.web.bind.annotation.RequestMapping'.
Authenticate		  Remove this unused import 'org.springframework.web.bind.annotation.RequestMapping'.
anyChipListS		  Immediately return this expression instead of assigning it to the temporary variable "sol".
anyChipListS		  Rename this class name to match the regular expression '^([A-Z][a-zA-Z0-9])*\$'.
anyChirpCor		  Rename this class name to match the regular expression '^([A-Z][a-zA-Z0-9])*\$'.
anyChirpRep		  Rename this interface name to match the regular expression '^([A-Z][a-zA-Z0-9])*\$'.
form.jsp		  Add a description to this table.
form.jsp		  Add a description to this table.
form.jsp		  Add a description to this table.
form.jsp		  Add a description to this table.
form.jsp		  Add a description to this table.
form.jsp		  Add a description to this table.

El 18/04/2022 realizamos otro análisis de sonarLint en el que nos dieron los siguientes errores:

- 9 malos olores
- 7 bugs

Este análisis se realizó cuando solo estaban el 50% de las tareas completas. Aquí tuvimos mas cuidado a la hora de realizar el código, pero aun así seguimos teniendo malos olores, además de algunos que no se solucionaron del anterior análisis.

Resource	Date	Description
Administrato		Remove this unused "averageRetailPriceOfComponentsByTechnologyAndCurrency" local variable.
Administrato		Remove this unused "minimumRetailPriceOfToolsByCurrencyEUR" local variable.
Administrato		Remove this useless assignment to local variable "minimumRetailPriceOfToolsByCurrencyEUR".
Authenticate		Remove this unused import 'org.springframework.web.bind.annotation.RequestMapping'.
Authenticate		Remove this unused import 'org.springframework.web.bind.annotation.RequestMapping'.
ExchangeMo		A "NullPointerException" could be thrown; "currencyExchange" is nullable here. [+2 locations]
anyChipListS		Immediately return this expression instead of assigning it to the temporary variable "sol".
anyChipListS		Rename this class name to match the regular expression '^[A-Z][a-zA-Z0-9]*\$'.
anyChirpCor		Rename this class name to match the regular expression '^[A-Z][a-zA-Z0-9]*\$'.
anyChirpRep		Rename this interface name to match the regular expression '^[A-Z][a-zA-Z0-9]*\$'.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.

El 25/04/2022 realizamos el último análisis de sonarLint en el que nos dieron los siguientes errores:

- 12 malos olores
- 18 bugs

Este análisis se realizó con todas las tareas completas. El número de errores se disparó enormemente debido a que tuvimos poco cuidado a la hora de mantener una buena calidad del código en las tareas restantes ya que tuvimos que darnos un poco mas de prisa para acabar las features.

Authenticate		Remove this unused import 'org.springframework.web.bind.annotation.RequestMapping'.
Authenticate		Remove this unused import 'org.springframework.web.bind.annotation.RequestMapping'.
Authenticate		Immediately return this expression instead of assigning it to the temporary variable "result".
Authenticate		Immediately return this expression instead of assigning it to the temporary variable "result".
Authenticate		Immediately return this expression instead of assigning it to the temporary variable "result".
InventorItem	29 minutes ago	Remove this useless assignment to local variable "result".
InventorPatr	29 minutes ago	Immediately return this expression instead of assigning it to the temporary variable "result".
PatronPatron	29 minutes ago	Immediately return this expression instead of assigning it to the temporary variable "result".
anyChiplistS		Immediately return this expression instead of assigning it to the temporary variable "sol".
anyChiplistS		Rename this class name to match the regular expression '^([A-Z][a-zA-Z0-9])*\$'.
anyChirpCor		Rename this class name to match the regular expression '^([A-Z][a-zA-Z0-9])*\$'.
anyChirpRep		Rename this interface name to match the regular expression '^([A-Z][a-zA-Z0-9])*\$'.
form.jsp	29 minutes ago	Add a description to this table.
form.jsp	29 minutes ago	Add a description to this table.
form.jsp	29 minutes ago	Add a description to this table.
form.jsp	29 minutes ago	Add a description to this table.
form.jsp	29 minutes ago	Add a description to this table.
form.jsp	29 minutes ago	Add a description to this table.
form.jsp	29 minutes ago	Add a description to this table.
form.jsp	29 minutes ago	Add a description to this table.
form.jsp	29 minutes ago	Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp	1 day ago	Add a description to this table.
form.jsp	1 day ago	Add a description to this table.
form.jsp	1 day ago	Add a description to this table.

Como podemos ver la mayoría son errores menores como variables que devolvíamos directamente en lugar de declararla primero, añadir descripciones a las tablas de los formularios e imports que no se usaban.



Para solucionar estos errores, simplemente creamos una rama a partir de máster y fuimos corrigiéndolos uno a uno, dando como resultado un análisis en blanco, lo que quiere decir que nuestro código está libre de bugs y malos olores:

5.2. D04

El 15/05/2022 realizamos un análisis de sonarLint en el que nos dieron los siguientes errores:

- 1 mal olor

Este análisis se realizó con el 70% de las tareas completas y se puede apreciar el cuidado que hemos tenido respecto al anterior entregable a la hora de mantener un código limpio y optimo.

Resource	Date	Description
AnyToolkitList	13 minutes ago	  Use a StringBuilder instead.

El 22/05/2022 realizamos un análisis de sonarLint y no nos dio ningún tipo de error a pesar de tener ya el 100% de las tareas completadas

6. Conclusiones

6.1. D03

Como conclusión, después de los distintos análisis podemos decir que al principio nos esforzamos por mantener un código limpio y óptimo. Sin embargo, a medida que avanzaba el sprint y se acercaba la fecha de entrega, empezamos a ser más descuidados. Esto nos llevó al último análisis en el que tuvimos un gran número de errores.

Como lección aprendida nos llevamos que antes de dar por terminada una tarea, debemos de realizar un análisis de sonarLint previo y solucionar los errores para ahorrarnos tiempo en el futuro

6.2. D04

En este entregable como conclusión podemos sacar que como equipo hemos mejorado en cuanto a la optimización y limpieza de código se refiere. Ya que en los distintos análisis solo nos hemos encontrado con un bad smell.

Esto ha sido posible ya que aprendimos la lección de hacer mas uso de la herramienta de sonarLint antes de realizar una entrega.

7. Bibliografía

SonarLint - <https://www.sonarlint.org/eclipse/>