# **Portada**

Número de grupo: E3.08

Url al repositorio: <https://github.com/antsermen/Acme-Toolkits-E3.08.git>

Miembros (+ emails):

- José Ramón Arias Expósito ([jrae1112jrae@gmail.com](mailto:rae1112jrae@gmail.co)),

- Manuel Carnero Vergel ([mancarver1@alum.us.es](mailto:mancarver1@alum.us.es)),

- Juan Carlos Moreno Pérez ([morenoperezjuancarlos@gmail.com](mailto:morenoperezjuancarlos@gmail.com)),

- Pablo Santos Pérez ([pablo.santos.232001@gmail.com](mailto:pablo.santos.232001@gmail.com)),

- David Sabugueiro Troya ([dsabugueiro14@gmail.com](mailto:dsabugueiro14@gmail.com)),

- Antonio Roberto Serrano Mena ([robertoserranomena@gmail.com](mailto:robertoserranomena@gmail.com))

# **Tabla de contenidos**

Contenido

[**1.** **Portada** 1](#_Toc104237621)

[**2.** **Tabla de contenidos** 1](#_Toc104237622)

[**3.** **Resumen ejecutivo** 2](#_Toc104237623)

[**4.** **Tabla de revisión** 2](#_Toc104237624)

[**5.** **Introducción** 3](#_Toc104237625)

[**6.** **Contenidos** 3](#_Toc104237626)

[**7.** **Conclusiones** 5](#_Toc104237627)

[**8.** **Bibliografía** 5](#_Toc104237628)

# **Resumen ejecutivo**

En este documento podemos encontrarnos con 8 apartados: una portada, que contiene el número de grupo, la url al repositorio en github, los miembros (junto a sus emails) y la fecha de creación del documento; una tabla de contenidos, que sería un índice típicamente conocido para un documento; un resumen ejecutivo, en el que se resumen los contenidos de cada apartado del documento; una tabla de revisión, formada por los datos de las revisiones realizadas sobre el documento (número de revisión, fecha y descripción); una introducción, que indica lo que se encuentra en el cuerpo del documento; los contenidos del documento, como su propio nombre indica, recoge el análisis que hemos hecho usando la herramienta Sonarlint, así como capturas y justificaciones de este; unas conclusiones, extraídas a partir de la finalización del documento y una bibliografía, que contiene las fuentes que nos sirvieron de apoyo para el documento (en caso de no encontrarse ninguna, aparecerá “Intencionalmente en blanco.”).

# **Tabla de revisión**

|  |  |  |
| --- | --- | --- |
| **N.º** | **Fecha** | **Descripción** |
| 1 | 20/05/2022 | Creación del documento y relleno parcial |
| 2 | 23/05/2022 | Puesta a punto del documento |

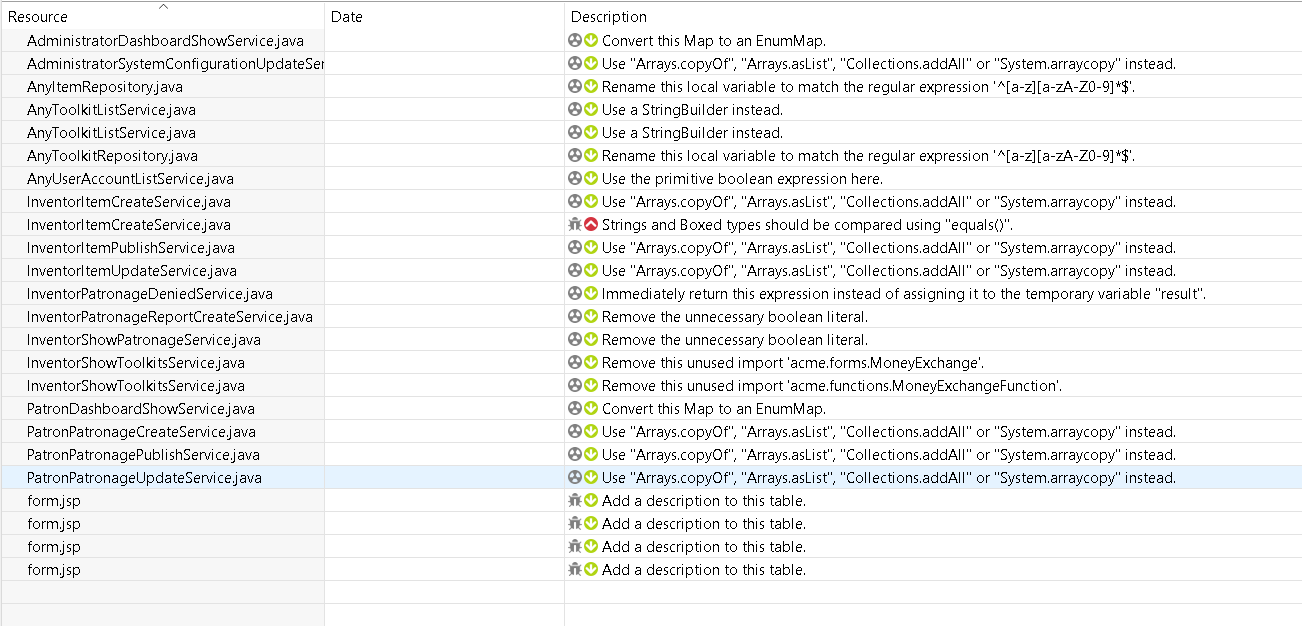
# **Introducción**

A lo largo del contenido de este documento encontraremos diferentes capturas de análisis realizados con la herramienta SonarLint y sus respectivos resultados.

La estructura del documento está dividida en 8 apartados, de los cuales los que más peso tienen son Contenidos y Conclusión, ya que en estos 2 apartados podemos encontrar esencialmente lo que nos pide el reporte.

# **Contenidos**

Sonarlint es una herramienta muy útil para detectar problemas en el código que a simple vista parecen inócuos, pero que generan deuda técnica a largo plazo si no se corrigen lo más rápido posible. Los ejemplos más claros de este efecto son los malos olores. Casi imperceptibles a simple vista, son errores que no afectan a la funcionalidad del código, pero acumulados en el tiempo suponen un aumento de la carga de trabajo para el futuro. Debido a esto es importante detectarlos a tiempo y ponerles solución. Huelga mencionar que, si bien se deben solucionar lo antes posible, a veces son necesarios para poder entregar la funcionalidad completa del código en la fecha acordada.

Para el análisis de nuestro proyecto, Sonarlint detectó los siguientes malos olores:

Existen dos bugs, uno en los form.jsp (se tratan de bugs menores debido a que no añadimos una descripción a las tablas que aparecen en los “show”. Hemos creído conveniente no añadirlos ya que añadiría valores al código que no son relevantes) y otro en la clase InventorItemCreateService (un bug grave, ya que usábamos “==” para comparar Strings y lo hemos solucionado sustituyéndolo por el método “equals”). Todo lo demás son malos olores.

Se han resuelto todos los malos olores siguiendo las indicaciones proporcionadas por SonarLint, como podemos ver en la siguiente imagen:

# **Conclusiones**

Como podemos ver, las notificaciones que Sonarlint nos devuelve al hacer el segundo análisis solo incluyen el bug en los form.jsp, que ya hemos explicado por qué consideramos que es inócuo. Dicho esto podemos asegurar que nuestro proyecto ha quedado libre de malos olores, y minimizando la deuda técnica, el código será mucho más entendible y accesible para los futuros encargados de gestionar dicho proyecto.

# **Bibliografía**

Intencionadamente en blanco.