

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

Lint Report



ACME SOFTWARE FACTORY

OUR FIRST PROJECT IN D&T

Grado en Ingeniería Informática – Ingeniería del Software


Diseño y Pruebas 2

Curso 2023 – 2024

Grupo de prácticas: C1-009


Autores por orden alfabético:

García Galocha, Rafael David

	Diseño y Pruebas II Acme-Software-Factory
	Lint Report


Índice de contenido

1. Resumen ejecutivo.....	3
2. Tabla de revisiones	4
3. Introducción	5
4. Contenido	6
5. Conclusiones	7
6. Bibliografía.....	8

	Diseño y Pruebas II Acme-Software-Factory
	Lint Report


1. Resumen ejecutivo

En este informe detallaré el análisis del código realizado gracias a la herramienta SonarLint.

	Diseño y Pruebas II Acme-Software-Factory
	Lint Report


2. Tabla de revisiones

Fecha	Versión	Descripción
24/04/2024	1.0	Primera versión del documento.

	Diseño y Pruebas II Acme-Software-Factory
	Lint Report

3. Introducción

A continuación, tras realizar el análisis sobre todo el proyecto, comentaré algunos detalles encontrados por la herramienta. También, en ciertos casos, incluiré una imagen para facilitar la comprensión.

	Diseño y Pruebas II Acme-Software-Factory
	Lint Report

4. Contenido

Algunas de las cosas más repetidas que nos destacan son:

- Rename this package name to match the regular expression `^[a-z_]+(\.[a-z_][a-z0-9_]*)*$`:
 Nos la indica porque cuando nuestros paquetes contienen dos palabras usamos la nomenclatura “**package** `acme.features.developer.trainingModule`;

- Replace this assert with a proper check:


Se debe a que en muchos sitios podemos encontrar

```
@Override
public void unbind(final TrainingModule object) {
    assert object != null;
```

- Define a constant instead of duplicating this literal “string” X times:


Ocurre porque en los servicios se indican propiedades en el bind y en el unbind, y normalmente suelen ser casi las mismas en ambos métodos

Lo que tienen en común todas estas incidencias es que son heredadas del framework y de los proyectos starters. Es decir, aparecen al seguir la nomenclatura y estilo de código de estos. Podemos comprobar fácilmente esto si el análisis con SonarLint se lo hacemos a Acme Framework. Por este motivo se ha decidido no modificar lo que nos indican y mantener el código de esta forma.

	Diseño y Pruebas II Acme-Software-Factory
	Lint Report

5. Conclusiones

Como hemos podido comprobar, aunque las herramientas de análisis de código son muy útiles para encontrar problemas en el código y bad smells, en nuestro caso, solo nos ha señalado lo anteriormente comentado, por lo que no hemos tenido que hacer cambios sustanciales gracias a la herramienta.

	Diseño y Pruebas II Acme-Software-Factory
	Lint Report

6. Bibliografía

Intencionadamente en blanco