

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

## Lint Report



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas 2.

Curso 2023 – 2024

Group:C2.001

Repository: <https://github.com/Albertoescobarsanchez/Acme-ANS-D04>

Student #3


ID Number: \*\*86\*0\*\*A

UVUS: juagonluc1

Name: González Lucena, Juan Antonio

Roles: development, analyst, tester, operator, manager

Date: May, 26, 2025

	Diseño y Pruebas 2 Lint Report
	Control de versiones

### Control de Versiones

Fecha	Versión	Descripción
26/05/2025	v1.0.0	Desarrollo de la primera versión.
30/06/2025	V1.2.0	Revisión del documento para convocatoria de Julio

### Índice de contenido

1.	Introducción	2
2.	Contenido	3
3.	Conclusiones	11



## 1. Introducción

Este documento recoge una lista de “bad smells” (malas prácticas o patrones problemáticos) detectados por SonarLint en nuestro proyecto. Concretamente, se realizara sobre las funcionalidades del student 3.

SonarLint es una herramienta de análisis estático que señala posibles errores, fallos de calidad o prácticas poco recomendables en el código. No obstante, no todos los hallazgos afectan de forma significativa al funcionamiento o rendimiento del software.

A continuación, se detallan los “bad smells” encontrados, con una descripción y su ubicación en el código. También se incluye una justificación para ellos, explicando por qué consideramos que no tienen impacto relevante y no requieren una corrección inmediata.

El objetivo del documento es mantener la transparencia en el mantenimiento del código, mostrar que se comprenden los avisos de SonarLint y explicar por qué se opta por corregir o ignorar cada caso. Así, se busca equilibrar la calidad del código con la eficiencia del desarrollo, evitando cambios innecesarios que no aporten valor real al proyecto.



## Diseño y Pruebas 2 Lint Report

## 2. Contenido

### ActivityLog

Resource	Date	Description
FlightCrewMemberActivityLogController.java	14 da...	Rename this package name to match the regular expression '^[a-z_](\.[a-z_][a-...
FlightCrewMemberActivityLogController.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberActivityLogController.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberActivityLogController.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberActivityLogController.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberActivityLogController.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberActivityLogController.java	14 da...	Remove this field injection and use constructor injection instead.

En la imagen se muestran 6 advertencias generadas por SonarLint en el archivo FlightCrewMemberActivityLogController.java

**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular. Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.

**Cinco advertencias sobre el uso de inyección de campos (@Autowired o similar).** Se sugiere reemplazar la inyección de campos por inyección por constructor. Aunque la inyección por constructor es una práctica recomendada por motivos de testabilidad y claridad, en la práctica el uso de inyección de campos funciona correctamente y no genera errores de ejecución. Si el proyecto no requiere una alta cobertura de tests unitarios, no hay una necesidad urgente de modificar este patrón.

Resource	Date	Description
FlightCrewMemberActivityLogRepository.java	14 da...	Rename this package name to match the regular expression '^[a-z_](\.[a-z_][a-...

En la imagen se muestra 1 advertencia generada por SonarLint en el archivo FlightCrewMemberActivityLogRepository.java

**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular. Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.

Resource	Date	Description
FlightCrewMemberActivityLogCreateService.java	3 days...	Replace this assert with a proper check.
FlightCrewMemberActivityLogCreateService.java	14 da...	Rename this package name to match the regular expression '^[a-z_](\.[a-z_][a-...
FlightCrewMemberActivityLogCreateService.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberActivityLogCreateService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberActivityLogCreateService.java	14 da...	This block of commented-out lines of code should be removed. [+3 locations]
FlightCrewMemberActivityLogCreateService.java	14 da...	Define a constant instead of duplicating this literal "masterId" 3 times. [+3 locati...

En la imagen se muestran 6 advertencias generadas por SonarLint en el archivo FlightCrewMemberActivityLogCreateService.java



## Diseño y Pruebas 2 Lint Report

**Dos assert sin verificación adecuada:** Se recomienda sustituir assert por una verificación con control de errores explícito. Aunque mejora la robustez, los assert no causan errores si se usan correctamente. No es urgente cambiarlos.

**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular. Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.

**Advertencia sobre el uso de inyección de campos (@Autowired o similar).** Se sugiere reemplazar la inyección de campos por inyección por constructor. Aunque la inyección por constructor es una práctica recomendada por motivos de testabilidad y claridad, en la práctica el uso de inyección de campos funciona correctamente y no genera errores de ejecución. Si el proyecto no requiere una alta cobertura de tests unitarios, no hay una necesidad urgente de modificar este patrón.

**Bloque de código comentado.** SonarLint sugiere eliminar líneas comentadas. Esto es una mejora estética, pero no afecta al funcionamiento.

**Literal duplicado ("masterId").** Se recomienda definirlo como constante para evitar repeticiones. Sin embargo, si solo aparece pocas veces, el beneficio es mínimo y no justifica cambios inmediatos.

Resource	Date	Description
FlightCrewMemberActivityLogDeleteService.java	14 da...	Rename this package name to match the regular expression '^[a-z_]+(\.[a-z_])?[a-z_]'
FlightCrewMemberActivityLogDeleteService.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberActivityLogDeleteService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberActivityLogDeleteService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberActivityLogDeleteService.java	14 da...	Replace this assert with a proper check.

En la imagen se muestran 5 advertencias generadas por SonarLint en el archivo FlightCrewMemberActivityLogDeleteService.java

**Tres assert sin verificación adecuada:** Se recomienda sustituir assert por una verificación con control de errores explícito. Aunque mejora la robustez, los assert no causan errores si se usan correctamente. No es urgente cambiarlos.

**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular. Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.

**Advertencia sobre el uso de inyección de campos (@Autowired o similar).** Se sugiere reemplazar la inyección de campos por inyección por constructor. Aunque la inyección por constructor es una práctica recomendada por motivos de testabilidad y claridad, en la práctica el uso de inyección de campos funciona correctamente y no genera errores de ejecución. Si el proyecto no requiere una alta cobertura de tests unitarios, no hay una necesidad urgente de modificar este patrón.



## Diseño y Pruebas 2

### Lint Report

Resource	Date	Description
FlightCrewMemberActivityLogListService.java	14 da...	Rename this package name to match the regular expression '^([a-z_]+)([a-z_]+)[a-z_]*\$'.
FlightCrewMemberActivityLogListService.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberActivityLogListService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberActivityLogListService.java	14 da...	Define a constant instead of duplicating this literal "draftMode" 3 times. [+3 locati...
FlightCrewMemberActivityLogListService.java	14 da...	Define a constant instead of duplicating this literal "masterId" 3 times. [+3 locati...

En la imagen se muestran 5 advertencias generadas por SonarLint en el archivo FlightCrewMemberActivityLogListService.java

**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular.

Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.

**Literal duplicado ("masterId", "draftMode").** Se recomienda definirlo como constante para evitar repeticiones. Sin embargo, si solo aparece pocas veces, el beneficio es mínimo y no justifica cambios inmediatos.

**Advertencia sobre el uso de inyección de campos (@Autowired o similar).** Se sugiere reemplazar la inyección de campos por inyección por constructor. Aunque la inyección por constructor es una práctica recomendada por motivos de testabilidad y claridad, en la práctica el uso de inyección de campos funciona correctamente y no genera errores de ejecución. Si el proyecto no requiere una alta cobertura de tests unitarios, no hay una necesidad urgente de modificar este patrón.

**Assert sin verificación adecuada:** Se recomienda sustituir assert por una verificación con control de errores explícito. Aunque mejora la robustez, los assert no causan errores si se usan correctamente. No es urgente cambiarlos.

Resource	Date	Description
FlightCrewMemberActivityLogPublishService.java	14 da...	Rename this package name to match the regular expression '^([a-z_]+)([a-z_]+)[a-z_]*\$'.
FlightCrewMemberActivityLogPublishService.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberActivityLogPublishService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberActivityLogPublishService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberActivityLogPublishService.java	14 da...	Replace this assert with a proper check.

En la imagen se muestran 5 advertencias generadas por SonarLint en el archivo FlightCrewMemberActivityLogPublishService.java

**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular. Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.

**Advertencia sobre el uso de inyección de campos (@Autowired o similar).** Se sugiere reemplazar la inyección de campos por inyección por constructor. Aunque la inyección por constructor es una práctica recomendada por motivos de testabilidad y claridad, en la práctica el uso de inyección de campos funciona correctamente y no genera errores de ejecución. Si el proyecto no requiere una alta cobertura de tests unitarios, no hay una necesidad urgente de modificar este patrón.

**Tres assert sin verificación adecuada:** Se recomienda sustituir assert por una verificación con control de errores explícito. Aunque mejora la robustez, los assert no causan errores si se usan correctamente. No es urgente cambiarlos.



## Diseño y Pruebas 2 Lint Report

Resource	Date	Description
FlightCrewMemberActivityLogShowService.java	14 da...	Rename this package name to match the regular expression <code>^[a-z_]+\.[a-z_][a-z_...</code>
FlightCrewMemberActivityLogShowService.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberActivityLogShowService.java	14 da...	Replace this assert with a proper check.

En la imagen se muestran 3 advertencias generadas por SonarLint en el archivo `FlightCrewMemberActivityLogShowService.java`

**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular. Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.

**Assert sin verificación adecuada:** Se recomienda sustituir `assert` por una verificación con control de errores explícito. Aunque mejora la robustez, los `assert` no causan errores si se usan correctamente. No es urgente cambiarlos.

**Advertencia sobre el uso de inyección de campos (@Autowired o similar).** Se sugiere reemplazar la inyección de campos por inyección por constructor. Aunque la inyección por constructor es una práctica recomendada por motivos de testabilidad y claridad, en la práctica el uso de inyección de campos funciona correctamente y no genera errores de ejecución. Si el proyecto no requiere una alta cobertura de tests unitarios, no hay una necesidad urgente de modificar este patrón.

Resource	Date	Description
FlightCrewMemberActivityLogUpdateService.java	14 da...	Rename this package name to match the regular expression <code>^[a-z_]+\.[a-z_][a-z_...</code>
FlightCrewMemberActivityLogUpdateService.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberActivityLogUpdateService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberActivityLogUpdateService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberActivityLogUpdateService.java	14 da...	This block of commented-out lines of code should be removed. [+3 locations]

En la imagen se muestran 5 advertencias generadas por SonarLint en el archivo `FlightCrewMemberActivityLogUpdateService.java`

**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular. Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.

**Dos assert sin verificación adecuada:** Se recomienda sustituir `assert` por una verificación con control de errores explícito. Aunque mejora la robustez, los `assert` no causan errores si se usan correctamente. No es urgente cambiarlos.

**Advertencia sobre el uso de inyección de campos (@Autowired o similar).** Se sugiere reemplazar la inyección de campos por inyección por constructor. Aunque la inyección por constructor es una práctica recomendada por motivos de testabilidad y claridad, en la práctica el uso de inyección de campos funciona correctamente y no genera errores de ejecución. Si el proyecto no requiere una alta cobertura de tests unitarios, no hay una necesidad urgente de modificar este patrón.

**Bloque de código comentado.** SonarLint sugiere eliminar líneas comentadas. Esto es una mejora estética, pero no afecta al funcionamiento.



## Diseño y Pruebas 2 Lint Report

### FlightAssignment

Resource	Date	Description
CrewMemberFlightAssignmentController.java	14 da...	Rename this package name to match the regular expression '^[a-z_]+\.[a-z_][a-...
CrewMemberFlightAssignmentController.java	14 da...	Remove this field injection and use constructor injection instead.
CrewMemberFlightAssignmentController.java	14 da...	Remove this field injection and use constructor injection instead.
CrewMemberFlightAssignmentController.java	14 da...	Remove this field injection and use constructor injection instead.
CrewMemberFlightAssignmentController.java	14 da...	Remove this field injection and use constructor injection instead.
CrewMemberFlightAssignmentController.java	14 da...	Remove this field injection and use constructor injection instead.
CrewMemberFlightAssignmentController.java	14 da...	Remove this field injection and use constructor injection instead.

En la imagen se muestran 7 advertencias generadas por SonarLint en el archivo CrewMemberFlightAssignmentController.java

**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular. Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.

**Seis advertencias sobre el uso de inyección de campos (@Autowired o similar).** Se sugiere reemplazar la inyección de campos por inyección por constructor. Aunque la inyección por constructor es una práctica recomendada por motivos de testabilidad y claridad, en la práctica el uso de inyección de campos funciona correctamente y no genera errores de ejecución. Si el proyecto no requiere una alta cobertura de tests unitarios, no hay una necesidad urgente de modificar este patrón.

Resource	Date	Description
FlightCrewMemberFlightAssignmentRepository.java	14 da...	Rename this package name to match the regular expression '^[a-z_]+\.[a-z_][a-...
FlightCrewMemberFlightAssignmentRepository.java	14 da...	This block of commented-out lines of code should be removed.

En la imagen se muestran 2 advertencias generadas por SonarLint en el archivo FlightCrewMemberFlightAssignmentRepository.java

**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular. Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.

**Bloque de código comentado.** SonarLint sugiere eliminar líneas comentadas. Esto es una mejora estética, pero no afecta al funcionamiento.

Resource	Date	Description
FlightCrewMemberAssignmentFlightCreateService.java	14 da...	Rename this package name to match the regular expression '^[a-z_]+\.[a-z_][a-...
FlightCrewMemberAssignmentFlightCreateService.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberAssignmentFlightCreateService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberAssignmentFlightCreateService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberAssignmentFlightCreateService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberAssignmentFlightCreateService.java	14 da...	This block of commented-out lines of code should be removed. [+5 locations]



	Diseño y Pruebas 2 Lint Report
---	-----------------------------------

En la imagen se muestran 6 advertencias generadas por SonarLint en el archivo FlightCrewMemberFlightAssignmentCreateService.java

**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular. Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.

**Advertencia sobre el uso de inyección de campos (@Autowired o similar).** Se sugiere reemplazar la inyección de campos por inyección por constructor. Aunque la inyección por constructor es una práctica recomendada por motivos de testabilidad y claridad, en la práctica el uso de inyección de campos funciona correctamente y no genera errores de ejecución. Si el proyecto no requiere una alta cobertura de tests unitarios, no hay una necesidad urgente de modificar este patrón.

**Bloque de código comentado.** SonarLint sugiere eliminar líneas comentadas. Esto es una mejora estética, pero no afecta al funcionamiento.

**Tres assert sin verificación adecuada:** Se recomienda sustituir assert por una verificación con control de errores explícito. Aunque mejora la robustez, los assert no causan errores si se usan correctamente. No es urgente cambiarlos.

Resource	Date	Description
FlightCrewMemberAssignmentFlightDeleteService.java	14 da...	Remove this empty statement.
FlightCrewMemberAssignmentFlightDeleteService.java	14 da...	Rename this package name to match the regular expression <code>^[a-z_]+\.[a-z_][a-z_...</code>
FlightCrewMemberAssignmentFlightDeleteService.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberAssignmentFlightDeleteService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberAssignmentFlightDeleteService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberAssignmentFlightDeleteService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberAssignmentFlightDeleteService.java	14 da...	Define a constant instead of duplicating this literal "status" 3 times. [+3 locations]

En la imagen se muestran 7 advertencias generadas por SonarLint en el archivo FlightCrewMemberFlightAssignmentDeleteService.java

**Advertencia sobre una instrucción vacía (;).** Es un detalle menor de limpieza. No causa errores. Puede ignorarse sin consecuencias.

**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular. Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.

**Advertencia sobre el uso de inyección de campos (@Autowired o similar).** Se sugiere reemplazar la inyección de campos por inyección por constructor. Aunque la inyección por constructor es una práctica recomendada por motivos de testabilidad y claridad, en la práctica el uso de inyección de campos funciona correctamente y no genera errores de ejecución. Si el proyecto no requiere una alta cobertura de tests unitarios, no hay una necesidad urgente de modificar este patrón.

**Tres assert sin verificación adecuada:** Se recomienda sustituir assert por una verificación con control de errores explícito. Aunque mejora la robustez, los assert no causan errores si se usan correctamente. No es urgente cambiarlos.

	<div>Diseño y Pruebas 2</div> <div>Lint Report</div>
---	--

**Literal duplicado (“status”).** Se recomienda definirlo como constante para evitar repeticiones. Sin embargo, si solo aparece pocas veces, el beneficio es mínimo y no justifica cambios inmediatos.

Resource	Date	Description
FlightCrewMemberAssignmentFlightListService.java	14 da...	Rename this package name to match the regular expression '^([a-z_]+\.[a-z_])'...
FlightCrewMemberAssignmentFlightListService.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberAssignmentFlightListService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberAssignmentFlightListService.java	14 da...	Define a constant instead of duplicating this literal "draftMode" 3 times. [+3 loca...

En la imagen se muestran 4 advertencias generadas por SonarLint en el archivo FlightCrewMemberFlightAssignmentListService.java

**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular. Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.

**Advertencia sobre el uso de inyección de campos (@Autowired o similar).** Se sugiere reemplazar la inyección de campos por inyección por constructor. Aunque la inyección por constructor es una práctica recomendada por motivos de testabilidad y claridad, en la práctica el uso de inyección de campos funciona correctamente y no genera errores de ejecución. Si el proyecto no requiere una alta cobertura de tests unitarios, no hay una necesidad urgente de modificar este patrón.

**Assert sin verificación adecuada:** Se recomienda sustituir assert por una verificación con control de errores explícito. Aunque mejora la robustez, los assert no causan errores si se usan correctamente. No es urgente cambiarlos.

**Literal duplicado (“draftMode”).** Se recomienda definirlo como constante para evitar repeticiones. Sin embargo, si solo aparece pocas veces, el beneficio es mínimo y no justifica cambios inmediatos.

Resource	Date	Description
FlightCrewMemberAssignmentFlightPublishService.java	14 da...	Rename this package name to match the regular expression '^([a-z_]+\.[a-z_])'...
FlightCrewMemberAssignmentFlightPublishService.java	14 da...	Merge this if statement with the enclosing one. [+1 location]
FlightCrewMemberAssignmentFlightPublishService.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberAssignmentFlightPublishService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberAssignmentFlightPublishService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberAssignmentFlightPublishService.java	14 da...	This block of commented-out lines of code should be removed. [+3 locations]
FlightCrewMemberAssignmentFlightPublishService.java	14 da...	This block of commented-out lines of code should be removed. [+9 locations]
FlightCrewMemberAssignmentFlightPublishService.java	14 da...	Define a constant instead of duplicating this literal "status" 3 times. [+3 locations]

En la imagen se muestran 8 advertencias generadas por SonarLint en el archivo FlightCrewMemberFlightAssignmentPublishService.java

**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular. Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.



## Diseño y Pruebas 2 Lint Report

**Advertencia sobre el uso de inyección de campos (@Autowired o similar).** Se sugiere reemplazar la inyección de campos por inyección por constructor. Aunque la inyección por constructor es una práctica recomendada por motivos de testabilidad y claridad, en la práctica el uso de inyección de campos funciona correctamente y no genera errores de ejecución. Si el proyecto no requiere una alta cobertura de tests unitarios, no hay una necesidad urgente de modificar este patrón.

**Dos bloques de código comentado.** SonarLint sugiere eliminar líneas comentadas. Esto es una mejora estética, pero no afecta al funcionamiento.

**Dos assert sin verificación adecuada:** Se recomienda sustituir assert por una verificación con control de errores explícito. Aunque mejora la robustez, los assert no causan errores si se usan correctamente. No es urgente cambiarlos.

**Literal duplicado ("status").** Se recomienda definirlo como constante para evitar repeticiones. Sin embargo, si solo aparece pocas veces, el beneficio es mínimo y no justifica cambios inmediatos.

**Fusionar un if anidado con su contenedor.** Mejora la legibilidad, pero no cambia el comportamiento. No es necesario modificarlo ahora.

Resource	Date	Description
FlightCrewMemberAssignmentFlightShowService.java	14 da...	Rename this package name to match the regular expression '^([a-z_]+)(\\.[a-z_][a-z_...'
FlightCrewMemberAssignmentFlightShowService.java	14 da...	Remove this field injection and use constructor injection instead.

En la imagen se muestran 2 advertencias generadas por SonarLint en el archivo FlightCrewMemberFlightAssignmentShowService.java

**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular. Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.

**Advertencia sobre el uso de inyección de campos (@Autowired o similar).** Se sugiere reemplazar la inyección de campos por inyección por constructor. Aunque la inyección por constructor es una práctica recomendada por motivos de testabilidad y claridad, en la práctica el uso de inyección de campos funciona correctamente y no genera errores de ejecución. Si el proyecto no requiere una alta cobertura de tests unitarios, no hay una necesidad urgente de modificar este patrón.

Resource	Date	Description
FlightCrewMemberAssignmentFlightUpdateService.java	14 da...	Rename this package name to match the regular expression '^([a-z_]+)(\\.[a-z_][a-z_...'
FlightCrewMemberAssignmentFlightUpdateService.java	14 da...	Remove this field injection and use constructor injection instead.
FlightCrewMemberAssignmentFlightUpdateService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberAssignmentFlightUpdateService.java	14 da...	Replace this assert with a proper check.
FlightCrewMemberAssignmentFlightUpdateService.java	14 da...	Define a constant instead of duplicating this literal "status" 3 times. [+3 locations]

En la imagen se muestran 5 advertencias generadas por SonarLint en el archivo FlightCrewMemberFlightAssignmentUpdateService.java



**Advertencia sobre el nombre del paquete:** El nombre del paquete no cumple con el formato recomendado por la expresión regular. Este tipo de advertencia es de estilo y no afecta en absoluto la funcionalidad del código. No es prioritario cambiarlo si no hay conflictos o problemas reales.

**Advertencia sobre el uso de inyección de campos (@Autowired o similar).** Se sugiere reemplazar la inyección de campos por inyección por constructor. Aunque la inyección por constructor es una práctica recomendada por motivos de testabilidad y claridad, en la práctica el uso de inyección de campos funciona correctamente y no genera errores de ejecución. Si el proyecto no requiere una alta cobertura de tests unitarios, no hay una necesidad urgente de modificar este patrón.

**Dos assert sin verificación adecuada:** Se recomienda sustituir assert por una verificación con control de errores explícito. Aunque mejora la robustez, los assert no causan errores si se usan correctamente. No es urgente cambiarlos.

**Literal duplicado ("status").** Se recomienda definirlo como constante para evitar repeticiones. Sin embargo, si solo aparece pocas veces, el beneficio es mínimo y no justifica cambios inmediatos.

### 3. Conclusiones

En conclusión, aunque el código no está completamente libre de "code smells", es importante destacar que muchos de estos no son el resultado de malas prácticas o errores propios. Las líneas de código que generan estas alertas son, en su mayoría, necesarias para el correcto funcionamiento del framework que se está utilizando, y se ha implementado siguiendo las especificaciones y directrices proporcionadas por los docentes en clase.

La decisión de mantener estos "code smells" está respaldada por una evaluación cuidadosa de su impacto en la funcionalidad y rendimiento de la aplicación. En muchos casos, las modificaciones sugeridas por SonarLint podrían comprometer la estabilidad o el comportamiento esperado del software, por lo que se ha optado por priorizar la adherencia a los requisitos del framework y las prácticas recomendadas.

El compromiso con la calidad del código sigue siendo una prioridad, y se continúa revisando y mejorando el trabajo de manera constante. La presencia de estos "code smells" no afecta negativamente la eficiencia o la fiabilidad de la aplicación, y se han tomado las medidas necesarias para asegurar que cualquier posible riesgo esté adecuadamente mitigado. Se agradece su comprensión y estamos abiertos a discutir cualquier preocupación o sugerencia adicional para mejorar nuestro proceso de desarrollo.