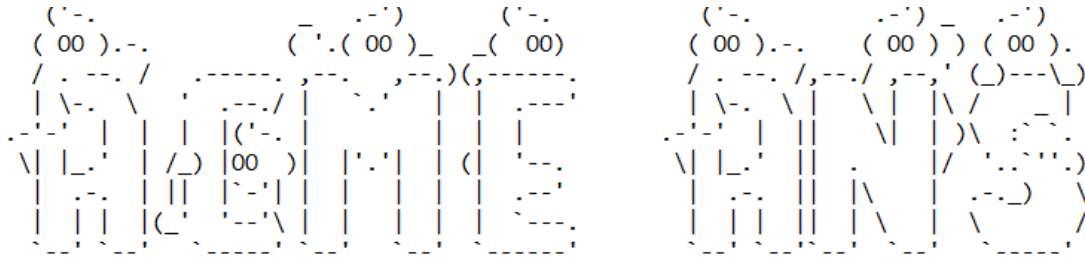


REPORTE DE LINT

Acme-ANS-D04



Repositorio: <https://github.com/FranciscoFernandezN/Acme-ANS>

Creado por el grupo C1.022, del G1

Participantes	
Nombres	Correos
Varo Vera, Juan	juavarver@alum.us.es

13 de Marzo de 2025

Índice

Portada.....	1
Índice.....	2
Resumen ejecutivo.....	3
Tabla de revisiones.....	4
Introducción.....	5
Contenidos.....	6
Conclusiones.....	15
Referencias.....	16

Resumen ejecutivo

Este es el proyecto del grupo C1.022 sobre Acme AirNav Solutions, S.A. (Acme ANS, S.A. abreviado), la cual es una compañía ficticia especializada en ayudar a aeropuertos a organizar y coordinar sus operaciones a partir de soluciones desarrolladas en software. La logística de los vuelos (la programación de los vuelos, la organización de reservas y de tripulación, etc.) se gestionan mediante el desarrollo de un WIS.

Con esto, para el correcto funcionamiento de la aplicación, se deberá escribir un reporte de testing donde se describan los valores de cobertura de código junto con los valores correspondientes al rendimiento de la aplicación, ambos previos y posteriores a realizar mutaciones en el código.

Tabla de revisiones

Número	Fecha	Descripción
V1.0.0	25/05/2025	Versión finalizada del documento para el entregable 4

Introducción

Para el contenido de este documento, se han tenido en cuenta los resultados de SonarLint para los requisitos grupales, tanto de aquellos requisitos obligatorios como opcionales.

Para cada entrada relevante de SonarLint, explicaré por qué pienso que no debe ser cambiado, o en su defecto, por qué lo he cambiado.

Contenidos

Debido a la gran cantidad de errores o recomendaciones que nos salen al hacer el análisis con SonarLint, se ha decidido hacer una selección, dado que realmente la mayoría de recomendaciones hacen referencia al uso de inyecciones y la duplicación de literales, siendo ambas inofensivas y gestionadas por el framework.

Cambios en Aircraft, Airline, Airport, Review y Service :

Aircraft.java	2 months ago	Override the "equals" method in this class.
Airline.java	2 months ago	Override the "equals" method in this class.
Airport.java	2 months ago	Override the "equals" method in this class.

Figura 1: Resultados SonarLint para Aircraft, Airline y Airport.

Review.java	2 months ago	Override the "equals" method in this class.
Service.java	2 months ago	Override the "equals" method in this class.

Figura 2: Resultados SonarLint para Review y Service.

Estos errores vienen provocados por no sobrecribir el método equals por defecto de Java, como sabemos que esto lo gestiona el framework internamente, podemos omitir este error para esta y todas las futuras entidades que aparezcan.

Cambios en AdministratorAirportController, AdministratorAirportCreateService, AdministratorAirportListService, AdministratorShowService y AdministratorAirportUpdateService:

AdministratorAirportController.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorAirportController.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorAirportController.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorAirportController.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorAirportCreateService.java	1 month ago	Define a constant instead of duplicating this literal "confirm" 3 times. [+3 locations]
AdministratorAirportCreateService.java	1 month ago	Define a constant instead of duplicating this literal "iATACode" 3 times. [+3 locations]
AdministratorAirportCreateService.java	1 month ago	Define a constant instead of duplicating this literal "operationalScope" 3 times. [+3 locations]
AdministratorAirportCreateService.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorAirportListService.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorAirportShowService.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorAirportUpdateService.java	1 month ago	Define a constant instead of duplicating this literal "confirm" 4 times. [+4 locations]
AdministratorAirportUpdateService.java	1 month ago	Define a constant instead of duplicating this literal "iATACode" 3 times. [+3 locations]
AdministratorAirportUpdateService.java	1 month ago	Define a constant instead of duplicating this literal "operationalScope" 3 times. [+3 locations]
AdministratorAirportUpdateService.java	1 month ago	Remove this field injection and use constructor injection instead.

Figura 3: Resultados SonarLint para AdministratorAirportController, AdministratorAirportCreateService, AdministratorAirportListService, AdministratorShowService y AdministratorAirportUpdateService.

Las recomendaciones hacen referencia a partes del código globalmente usadas en Acme-Framework, en concreto la notación @Autowired, por lo que las recomendaciones son inofensivas y no se cambiarán.

También se reportan errores que hacen referencia a la no duplicación de literales, la cual se produce a la hora de poner los atributos de un objeto en forma de String para su método unbind, bind, etc. Como es irracional poner una constante para estos literales, esta entrada de SonarLint también será omitida.

Cambios en AdministratorAirlineController, AdministratorAirlineCreateService, AdministratorAirlineListService, AdministratorAirlineShowService y AdministratorAirlineUpdateService:

AdministratorAirlineController.java	2 months ago	Remove this field injection and use constructor injection instead.
AdministratorAirlineController.java	2 months ago	Remove this field injection and use constructor injection instead.
AdministratorAirlineController.java	2 months ago	Remove this field injection and use constructor injection instead.
AdministratorAirlineController.java	2 months ago	Remove this field injection and use constructor injection instead.
AdministratorAirlineCreateService.java	1 month ago	Define a constant instead of duplicating this literal "iATACode" 3 times. [+3 locatio
AdministratorAirlineCreateService.java	2 months ago	Remove this field injection and use constructor injection instead.
AdministratorAirlineListService.java	2 months ago	Remove this field injection and use constructor injection instead.
AdministratorAirlineShowService.java	2 months ago	Remove this field injection and use constructor injection instead.
AdministratorAirlineUpdateService.java	1 month ago	Define a constant instead of duplicating this literal "iATACode" 3 times. [+3 locatio
AdministratorAirlineUpdateService.java	2 months ago	Remove this field injection and use constructor injection instead.

Figura 4: Resultados SonarLint AdministratorAirlineController, AdministratorAirlineCreateService, AdministratorAirlineListService, AdministratorAirlineShowService y AdministratorAirlineUpdateService

Las recomendaciones hacen referencia a partes del código globalmente usadas en Acme-Framework, en concreto la notación @Autowired, por lo que las recomendaciones son inofensivas y no se cambiarán.

También se reportan errores que hacen referencia a la no duplicación de literales, la cual se produce a la hora de poner los atributos de un objeto en forma de String para su método unbind, bind, etc. Como es irracional poner una constante para estos literales, esta entrada de SonarLint también será omitida.

Cambios en AdministratorAircraftController, AdministratorAircraftCreateService, AdministratorAircraftListService, AdministratorAircraftShowService y AdministratorAircraftUpdateService:


















AdministratorAircraftController.java	1 month ago		Remove this field injection and use constructor injection instead.
AdministratorAircraftController.java	2 months ago		Remove this field injection and use constructor injection instead.
AdministratorAircraftController.java	2 months ago		Remove this field injection and use constructor injection instead.
AdministratorAircraftController.java	2 months ago		Remove this field injection and use constructor injection instead.
AdministratorAircraftController.java	2 months ago		Remove this field injection and use constructor injection instead.
AdministratorAircraftCreateService.java	11 hours ago		Use a primitive boolean expression here.
AdministratorAircraftCreateService.java	11 hours ago		Define a constant instead of duplicating this literal "airline" 3 times. [+3 locations]
AdministratorAircraftCreateService.java	1 month ago		Define a constant instead of duplicating this literal "registrationNumber" 3 times. [
AdministratorAircraftCreateService.java	2 months ago		Remove this field injection and use constructor injection instead.
AdministratorAircraftDisableService.java	11 hours ago		Define a constant instead of duplicating this literal "airline" 4 times. [+4 locations]
AdministratorAircraftDisableService.java	1 month ago		Define a constant instead of duplicating this literal "registrationNumber" 3 times. [
AdministratorAircraftDisableService.java	1 month ago		Remove this field injection and use constructor injection instead.
AdministratorAircraftListService.java	2 months ago		Remove this field injection and use constructor injection instead.
AdministratorAircraftShowService.java	2 months ago		Remove this field injection and use constructor injection instead.
AdministratorAircraftUpdateService.java	11 hours ago		Define a constant instead of duplicating this literal "airline" 4 times. [+4 locations]
AdministratorAircraftUpdateService.java	1 month ago		Define a constant instead of duplicating this literal "registrationNumber" 3 times. [
AdministratorAircraftUpdateService.java	2 months ago		Remove this field injection and use constructor injection instead.

Figura 5: Resultados SonarLint para AdministratorAircraftController, AdministratorAircraftCreateService, AdministratorAircraftListService, AdministratorAircraftShowService y AdministratorAircraftUpdateService.

Las recomendaciones hacen referencia a partes del código globalmente usadas en Acme-Framework, en concreto la notación `@Autowired`, por lo que las recomendaciones son inofensivas y no se cambiarán.

También se reportan errores que hacen referencia a la no duplicación de literales, la cual se produce a la hora de poner los atributos de un objeto en forma de String para su método `unbind`, `bind`, etc. Como es irracional poner una constante para estos literales, esta entrada de SonarLint también será omitida.

Podemos apreciar que el error que sale en azul hace referencia a que una condición de tipo `<condición>?<result1>:<result2>`, la condición no debe ser el propio booleano, sino que debe ser: `booleano == condición_booleano`. Esto, no es relevante para el proyecto y además, empeora la legibilidad del código, por lo que no supone un problema para el proyecto.

Cambios en AnyServiceController, AnyServiceListService y AnyServiceShowService:

AnyServiceController.java	1 month ago	Remove this field injection and use constructor injection instead.
AnyServiceController.java	1 month ago	Remove this field injection and use constructor injection instead.
AnyServiceListService.java	1 month ago	Remove this field injection and use constructor injection instead.
AnyServiceShowService.java	1 month ago	Remove this field injection and use constructor injection instead.

Figura 6: Resultados SonarLint para AnyServiceController, AnyServiceListService y AnyServiceShowService

Las recomendaciones hacen referencia a partes del código globalmente usadas en Acme-Framework, en concreto la notación `@Autowired`, por lo que las recomendaciones son inofensivas y no se cambiarán.

Cambios en SupportedCurrency y SupportedCurrencyValidator:

SupportedCurrency.java	9 days ago	A "NullPointerException" could be thrown; "getBody()" can return null. [+2 location]
SupportedCurrency.java	9 days ago	A "NullPointerException" could be thrown; "getBody()" can return null. [+2 location]
SupportedCurrency.java	9 days ago	Catch Exception instead of Throwable.
SupportedCurrency.java	9 days ago	Catch Exception instead of Throwable.
SupportedCurrency.java	9 days ago	Replace this use of System.out by a logger.
SupportedCurrency.java	9 days ago	Replace this use of System.out by a logger.
SupportedCurrency.java	2 months ago	Override the "equals" method in this class.
SupportedCurrencyValidator.java	2 months ago	Remove this field injection and use constructor injection instead.
SupportedCurrencyValidator.java	2 months ago	Replace this assert with a proper check.

Figura 7: Resultados SonarLint para SupportedCurrency y SupportedCurrencyValidator.

Los dos primeros errores reportados por SonarLint hacen referencia a posibles `NullPointerException` al acceder a `lastData` y su contenido. Sin embargo, estas advertencias se consideran inofensivas, ya que el acceso se encuentra protegido por una comprobación explícita de nulidad, y el resto del flujo está gestionado adecuadamente por el framework. Además, este patrón es común en la interacción con servicios externos, y su modificación añadiría complejidad innecesaria sin beneficios reales, por lo que se ha decidido no realizar cambios.

Los dos errores señalados por SonarLint hacen referencia a la captura de una excepción de tipo `Throwable` en lugar de una más específica como `Exception`. Sin embargo, en este caso concreto, capturar `Throwable` permite cubrir no solo excepciones comprobadas, sino también errores inesperados que podrían surgir en tiempo de ejecución, como problemas de red o fallos del servicio externo. Dado que se trata de un método de utilidad que no compromete la integridad del sistema y cuyo uso está limitado, esta estrategia ofrece una mayor robustez y seguridad. Además, el framework se encarga de gestionar adecuadamente el contexto de ejecución, por lo que no se

considera necesario restringir artificialmente el tipo de excepción capturada. Por tanto, se ha decidido mantener esta implementación.

Las dos siguientes recomendaciones señaladas por SonarLint hacen referencia al uso de `System.out.println()` en lugar de un sistema de logging formal. No obstante, en este contexto su uso no representa un problema significativo, ya que se emplea exclusivamente en bloques `catch` para capturar excepciones que no afectan al flujo crítico de la aplicación. Además, dado que este método no forma parte de un entorno productivo o de misión crítica, la implementación de un logger supondría una complejidad innecesaria para un manejo de errores puntual y no esencial. Por tanto, se ha decidido mantener `System.out.println()` en este punto del código

El error marcado en azul viene provocado por no sobrescribir el método `equals` por defecto de Java, como sabemos que esto lo gestiona el framework internamente, podemos omitir este error para esta y todas las futuras entidades que aparezcan.

La penúltima hace referencia a partes del código globalmente usadas en Acme-Framework, en concreto la notación `@Autowired`, por lo que las recomendaciones son inofensivas y no se cambiarán.

El último error se reporta la necesidad de eliminar un `assert` innecesario, y tras analizar bien el origen de este, se ha llegado a la conclusión de que originalmente eran del proyecto Hello World, por lo que dicha línea ha sido eliminada y por tanto se ha corregido el error.

Cambios en AnyServiceController, AnyServiceListService y AnyServiceShowService:

AnyServiceController.java	1 month ago	Remove this field injection and use constructor injection instead.
AnyServiceController.java	1 month ago	Remove this field injection and use constructor injection instead.
AnyServiceListService.java	1 month ago	Remove this field injection and use constructor injection instead.
AnyServiceShowService.java	1 month ago	Remove this field injection and use constructor injection instead.

Figura 8: Resultados SonarLint en AnyServiceController, AnyServiceListService y AnyServiceShowService.

Las recomendaciones hacen referencia a partes del código globalmente usadas en Acme-Framework, en concreto la notación `@Autowired`, por lo que las recomendaciones son inofensivas y no se cambiarán.

Cambios en AnyReviewController, AnyReviewCreateService y AnyReviewFetchService, AnyReviewListService, AnyReviewRestController:

AnyReviewController.java	2 months ago	Remove this field injection and use constructor injection instead.
AnyReviewController.java	2 months ago	Remove this field injection and use constructor injection instead.
AnyReviewCreateService.java	2 months ago	Remove this field injection and use constructor injection instead.
AnyReviewFetchService.java	1 month ago	Replace the type specification in this constructor call with the diamond operator ("
AnyReviewFetchService.java	1 month ago	Remove this field injection and use constructor injection instead.
AnyReviewListService.java	1 day ago	Use a primitive boolean expression here.
AnyReviewListService.java	2 months ago	Remove this field injection and use constructor injection instead.
AnyReviewRestController.java	1 month ago	Remove this field injection and use constructor injection instead.

Figura 9: Resultados SonarLint para AnyReviewController, AnyReviewCreateService y AnyReviewFetchService, AnyReviewListService, AnyReviewRestController.

Las recomendaciones de color marrón hacen referencia a partes del código globalmente usadas en Acme-Framework, en concreto la notación @Autowired, por lo que las recomendaciones son inofensivas y no se cambiarán.

La cuarta advertencia no señala un error funcional, sino una sugerencia estilística relacionada con la inferencia de tipos genéricos. SonarLint sugiere que podría omitirse el tipo en la creación del objeto (new DatatableOutput<>(...)) para hacer el código más conciso. Sin embargo, este patrón de construcción es perfectamente válido, no introduce ningún problema de compilación ni ejecución y es, de hecho, común en proyectos que utilizan frameworks que gestionan este tipo de clases genéricas.

El segundo error que sale en azul hace referencia a que una condición de tipo <condición>?<result1>:<result2>, la condición no debe ser el propio booleano, sino que debe ser: booleano == condición_booleano. Esto, no es relevante para el proyecto y además, empeora la legibilidad del código, por lo que no supone un problema para el proyecto.

Cambios en AdministratorServiceController, AdministratorServiceCreateService y AdministratorServiceDeleteService, AdministratorServiceListService, AdministratorServiceShowService y AdministratorServiceUpdateService:

AdministratorServiceController.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorServiceController.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorServiceController.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorServiceController.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorServiceController.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorServiceCreateService.java	1 month ago	Define a constant instead of duplicating this literal "airport" 4 times. [+4 locations]
AdministratorServiceCreateService.java	1 month ago	Define a constant instead of duplicating this literal "promotionCode" 3 times. [+3 locations]
AdministratorServiceCreateService.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorServiceDeleteService.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorServiceDeleteService.java	1 month ago	Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException
AdministratorServiceListService.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorServiceShowService.java	1 month ago	Remove this field injection and use constructor injection instead.
AdministratorServiceUpdateService.java	1 month ago	Define a constant instead of duplicating this literal "airport" 4 times. [+4 locations]
AdministratorServiceUpdateService.java	1 month ago	Define a constant instead of duplicating this literal "promotionCode" 3 times. [+3 locations]
AdministratorServiceUpdateService.java	1 month ago	Remove this field injection and use constructor injection instead.

Figura 10: Resultados SonarLint para AdministratorServiceController, AdministratorServiceCreateService y AdministratorServiceDeleteService, AdministratorServiceListService, AdministratorServiceShowService y AdministratorServiceUpdateService.

Las recomendaciones de color marrón hacen referencia a partes del código globalmente usadas en Acme-Framework, en concreto la notación @Autowired, por lo que las recomendaciones son inofensivas y no se cambiarán.

Otro error común que deteta SonarLint es que el método validate está vacío, y se debería de explicar el por qué de ello. Como en Acme Jobs esta praxis es recurrente, este aviso es totalmente inofensivo, ya que se da por hecho que si el método validate está vacío, implica que no se debe validar nada por la propia naturaleza del requisito.

Se reportan también errores que hacen referencia a la no duplicación de literales, la cual se produce a la hora de poner los atributos de un objeto en forma de String para su método unbind, bind, etc. Como es irracional poner una constante para estos literales, esta entrada de SonarLint también será omitida.

Cambios en AdministratorDashboardController y AdministratorDashboardShowService

AdministratorDashboardController.java	5 days ago	Remove this field injection and use constructor injection instead.
AdministratorDashboardShowService.java	5 days ago	Convert this Map to an EnumMap.
AdministratorDashboardShowService.java	5 days ago	Extract this nested ternary operation into an independent statement.
AdministratorDashboardShowService.java	5 days ago	Extract this nested ternary operation into an independent statement.
AdministratorDashboardShowService.java	5 days ago	Extract this nested ternary operation into an independent statement.
AdministratorDashboardShowService.java	5 days ago	Extract this nested ternary operation into an independent statement.
AdministratorDashboardShowService.java	5 days ago	Refactor this method to reduce its Cognitive Complexity from 27 to the 15 allowed
AdministratorDashboardShowService.java	5 days ago	Replace this usage of 'Stream.collect(Collectors.toList())' with 'Stream.toList()' and
AdministratorDashboardShowService.java	5 days ago	Convert this Map to an EnumMap.
AdministratorDashboardShowService.java	5 days ago	Remove this field injection and use constructor injection instead.

Figura 11: Resultados SonarLint para AdministratorDashboardController y AdministratorDashboardShowService.

La primera recomendación hace referencia a partes del código globalmente usadas en Acme-Framework, en concreto la notación `@Autowired`, por lo que las recomendaciones son inofensivas y no se cambiarán.


Se reporta también un error debido a que un atributo está definido como `Map` en lugar de `EnumMap`. Es una buena recomendación, dado que es una forma de optimizar el código, hacerlo más legible y mejorar el rendimiento. Por ello hemos decidido modificarlo.

Se registran también cuatro errores a causa del uso del operador ternario para evaluar distintas casuísticas. Se trata de un error inofensivo debido a que expresa la misma funcionalidad que una sentencia `if/else` pero en un menor número de líneas de código, de forma que es más legible, y no supone coste de rendimiento, por lo tanto hemos decidido no modificarlos.

Se registra también un error a causa de la complejidad cognitiva, sin embargo no es algo relevante que se tenga que modificar, dado que en los `validate` se implementan las restricciones necesarias para cumplir los requisitos asociados, lo cuál se ve reflejado a través de estructuras `if/else` que evalúen las distintas casuísticas.

Se registra también una recomendación que nos comenta que debemos cambiar un `Stream.collect(Collectors.toList)` por un `.toList()`. Aunque no se trata de un error, se ha decidido cambiarlo para facilitar la legibilidad del código y mejorar el rendimiento.

Cambios en Ban:

A screenshot of the SonarLint interface showing a file named 'Ban.java' with a '2 months ago' timestamp. A red error icon is visible next to the text 'Override the "equals" method in this class.'

Ban.java 2 months ago Override the "equals" method in this class.

Figura 12: Resultados SonarLint para Ban.

El error viene provocado por no sobrecribir el método `equals` por defecto de Java, como sabemos que esto lo gestiona el framework internamente, podemos omitir este error para esta y todas las futuras entidades que aparezcan.

Cambios en AdministratorBanController, AdministratorBanCreateService y AdministratorBanListService, AdministratorBanShowService, AdministratorBanUpdateService, AdministratorBannedPassengersLastMonth y AdministratorBannedPassengersListService:

AdministratorBanController.java	6 days ago	Remove this field injection and use constructor injection instead.
AdministratorBanController.java	6 days ago	Remove this field injection and use constructor injection instead.
AdministratorBanController.java	6 days ago	Remove this field injection and use constructor injection instead.
AdministratorBanController.java	6 days ago	Remove this field injection and use constructor injection instead.
AdministratorBanCreateService.java	6 days ago	Remove this field injection and use constructor injection instead.
AdministratorBanCreateService.java	6 days ago	Define a constant instead of duplicating this literal "confirm" 3 times. [+3 locations]
AdministratorBanCreateService.java	6 days ago	Define a constant instead of duplicating this literal "liftDate" 3 times. [+3 locations]
AdministratorBanCreateService.java	6 days ago	Define a constant instead of duplicating this literal "passenger" 6 times. [+6 locations]
AdministratorBanListService.java	6 days ago	Remove this field injection and use constructor injection instead.
AdministratorBanShowService.java	6 days ago	Remove this field injection and use constructor injection instead.
AdministratorBanUpdateService.java	6 days ago	Remove this field injection and use constructor injection instead.
AdministratorBanUpdateService.java	6 days ago	Define a constant instead of duplicating this literal "confirm" 3 times. [+3 locations]
AdministratorBanUpdateService.java	6 days ago	Define a constant instead of duplicating this literal "liftDate" 3 times. [+3 locations]
AdministratorBannedPassengersLastMonthListService.java	6 days ago	Remove this field injection and use constructor injection instead.
AdministratorBannedPassengersListService.java	6 days ago	Remove this field injection and use constructor injection instead.

Figura 13: Resultados SonarLint para AdministratorBanController, AdministratorBanCreateService y AdministratorBanListService, AdministratorBanShowService, AdministratorBanUpdateService, AdministratorBannedPassengersLastMonth y AdministratorBannedPassengersListService.

Las recomendaciones de color marrón hacen referencia a partes del código globalmente usadas en Acme-Framework, en concreto la notación @Autowired, por lo que las recomendaciones son inofensivas y no se cambiarán.

Se reportan también errores que hacen referencia a la no duplicación de literales, la cual se produce a la hora de poner los atributos de un objeto en forma de String para su método unbind, bind, etc. Como es irracional poner una constante para estos literales, esta entrada de SonarLint también será omitida.

Cambios en AnyServiceController, AnyServiceListService y AnyServiceShowService:

AnyAirportController.java	1 month ago	Remove this field injection and use constructor injection instead.
AnyAirportController.java	1 month ago	Remove this field injection and use constructor injection instead.
AnyAirportListService.java	1 month ago	Remove this field injection and use constructor injection instead.
AnyAirportShowService.java	1 month ago	Remove this field injection and use constructor injection instead.

Figura 14: Resultados SonarLint para AnyServiceController, AnyServiceListService y AnyServiceShowService.

Las recomendaciones hacen referencia a partes del código globalmente usadas en Acme-Framework, en concreto la notación @Autowired, por lo que las recomendaciones son inofensivas y no se cambiarán.

Conclusiones

Como conclusión, el reporte Lint proporciona cierta información sobre el proyecto y propone algunas ideas útiles para mejorar la calidad estática del código, aunque, en la mayoría de casos, estos cambios no suponen una mejora clara.

En este proyecto, al usarse mucho la notación @Autowired y usar los métodos de Acme-Framework para bindear los datos, surgen muchos avisos por parte de SonarLint, pero que pueden ser omitidos sin ningún tipo de problema.

Bibliografía

Intencionalmente en blanco