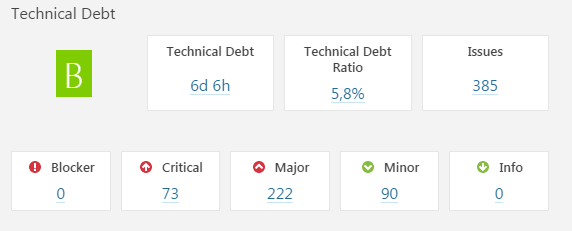
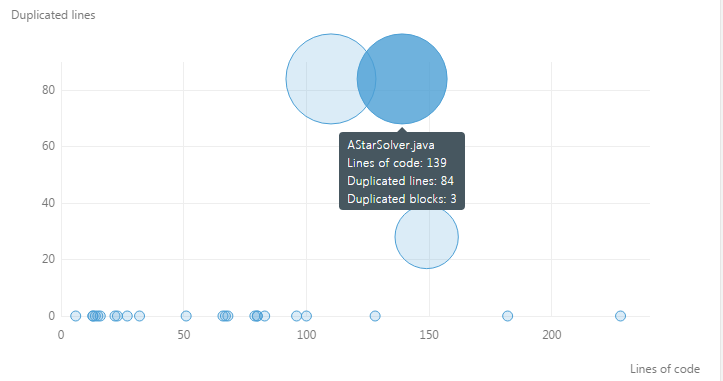
# Análisis del código existente



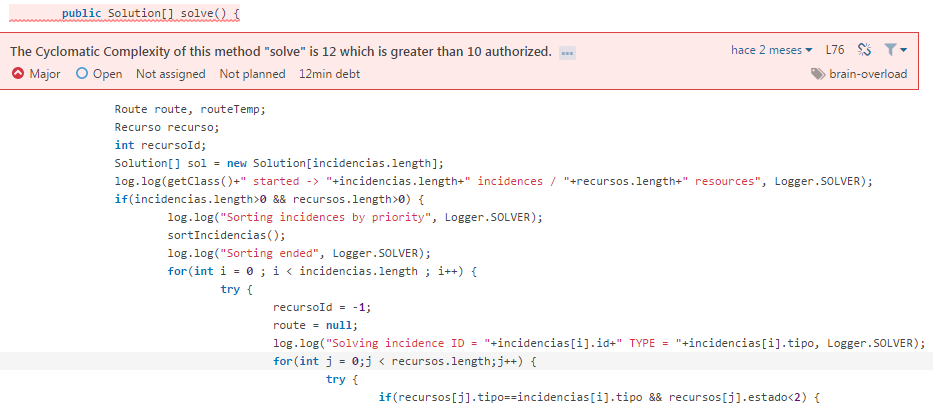
Trás analizar el código del proyecto, se ha decidido empezar por refactorizar las clases Solver, que se encargan de recoger las incidencias y los recursos disponibles y a emparejarlos del modo más óptimo posible.

Otra de las clases que convendría revisar es StationConnectionManager, que es también otra de las partes principales del programa, y tiene casi toda su funcionalidad concentrada en el método **public** **void** manageMsg(Message msg), que habría que dividir o simplificar, pero se ha considerado que los Solver tienen prioridad porque sufren de problemas y malas prácticas más graves.

Se ha optado por revisar esta parte del código porque es una de las partes principales de la aplicación, y de las que más probleas tienen. Las clases relevantes del paquete ia que se tratarán de refactorizar son AStarSolver y RandomSolver.



Por una parte, estas clases cuentan con una grán cantidad de código duplicado, ya que de hecho solo hay un método (**public** Solution[] solve()) que debería estar en estas clases, y el resto podría ser común para todos los Solver.



Por otra parte, las implementaciones de solve(), especialmente la de AstarSolver, tienen demasiados for e if anidados, además de demasiadas líneas, por lo que su complejidad es demasiado alta.

# Pruebas unitarias

Se han creado testeos para las clases AStarSolver y RandomSolver para comprobar que todo sigue funcionando como lo hacía originalmente, además de un test suite para lanzar todos los testeos al mismo tiempo. Estos tests están en el paquete tests del proyecto.

Para testear el funcionamiento de estas clases, lo más directo habría sido hacer un mock de la clase MessageParser, que es la que se encarga de enviar los mensajes a los recursos. Sin embargo, esto es más complicado de lo que parecía en un principio, puesto que el sistema tiene múltiples componentes que dependen unos de otros. Otro problema era la base de datos, ya que hacer un mock del SQLConnector habría sido muy costoso, y como solo se usan procedimientos almacenados, no hay forma de distinguir las consultas de los updates.

Finalmente, se ha optado por usar la base de datos real con unos datos concretos insertados (en DBTests.sql está el backup), y usar el Logger como medio para comprobar el comportamiento de la aplicación. Esto hace los testeos mucho más sencillos (solo hay que hacer un mock del Logger, que solo tiene una o dos funciones relevantes), es incluso más estrcto que comprobar la solución (los logs tienen que ser idénticos).

Otro problema para hacer los testeos es que los Solver funcionan de forma asíncrona: el método que se llama para que empieze a resolver provoca que otro hilo empieze a hacer el trabajo que queremos testear. En caso de intentar llamar directamente a **public** Solution[] solve() habría que arreglar e inicializar múltiples cosas, por lo que se ha decidido llamar a método público que se usa convencionalmente y esperar 4 segundos para que se impriman los logs. Esto debería ser un método aceptable para hacer el testeo, ya que de todas formas si tarda mas de 4 segundos en resolver una incdencia es que algo está mal (así el testeo también comprueba el rendimiento).

# Realización de mejoras

En primér lugar, se ha procedido a cambiar Solver, que pasa de ser una interfaz a ser una clase abstracta. De este modo, todo el código en común entre AStarSolver y RandomSolver pasa a Solver, y ahora las otras clases solo cuentan con la implementacion de **public** Solution[] solve(), que es lo único que tiene que cambiar entre ellas.

Por otra parte, también se han cambiado algunos nombres que podrían resultar confusos o incompletos por unos más correctos (ej: ifac -> incidenciaFacade).

Después, se ha procedido a dividir el método solve(), separándolo en dos funciones, donde la nueva funcion consiste en la resolución de una sola incidencia (anteriormente el loop interior de solve()).

Tras realizar los cambios, se han pasado testeos realizados a priori para comprobar que, efectivamente, no se ha alterado el funcionamiento del programa con estos cambios.

# Repositorio

<https://github.com/jonayerdi/SkynetCore>