

Análisis de calidad

Se pasa el test de sonar y se encuentran una serie de posibles fallos en la implementación del código que deberán ser solucionados, como se detallará a continuación. Se obtiene una 'A' en bugs al no crear ninguno con la implementación del nuevo código, por lo que se considerará este apartado como óptimo y no se deberá aplicar ninguna medida correctiva sobre él. En la parte de vulnerabilidades se obtiene una calificación 'B', esto se produce al encontrar cuatro nuevas vulnerabilidades en la sección de código que ha sido añadido, por lo que serán identificadas y se procederá a aplicar las diferentes medidas correctivas:

- **LineasFragment:** el atributo "callbackParadas" es público y se deberá modificar para que su tipo sea privado. Se añadirán también sus correspondientes get y set para que pueda ser conseguido o modificado desde otra clase.
- **LineasTodasFragment:** sucede lo mismo que en el apartado anterior, el atributo "callbackParadas" es público y se deberá modificar para que su tipo sea privado. Se añadirán también sus correspondientes get y set para que pueda ser obtenido o modificado desde otra clase.
- **ViewHolder:** en este caso también se deberá cambiar el tipo de los atributos textViewNumero y textViewNombre pasando estos de ser públicos a privados. En este caso no será necesario implementar sus "set" ni sus "get".

Por último, se observará que en el apartado de mantenibilidad se obtiene una valoración de 'A', esta es la máxima valoración posible, aunque no significa que no se pueda mejorar. Con la creación del nuevo código hemos generado veintitrés nuevos code smells que aumentan el total de la deuda técnica tres horas. Se deberá comprobar la deuda que generan en la gráfica de hostpost e ir intentado solucionarlos empezando desde los que más generen. Para ello se encuentran las siguientes soluciones en las diferentes clases o paquetes:

- **Paquete Model:** se deberá cambiar la letra inicial del nombre de este paquete pasando a ser minúscula.
- **ParserJson:** la variable local del método readParada "Py" deberá ser declarada e inicializada en una línea independiente a la de otras variables locales.
- **ListLineasPresenter:** creamos un atributo de tipo Context que no utilizamos, el sonar nos pide que se elimine al no tener una función en el punto actual del proyecto, pero se mantendrá al tener función en futuras implementaciones.
- **ListParadaLineasPresenter:** eliminar un parámetro en la creación de paradasLinea del método onPostExecute que no se utilizaba durante la ejecución del código. Eliminar la variable leerLineasParadasInternet al no ser utilizada en la clase.

- **DetallesLineaActivity:** se eliminará el import correspondiente a la clase `ParserJSON` ya que no se intenta acceder a ninguna de las variables que se encuentran en esta clase.
- **LineasFragment:** renombrar `callBack` y todas sus referencias en la clase, se pasará a nombrar como `callBackParadas`. Modificar el tipo del atributo `callback`, pasando este de ser público a privado.
- **LineasTodasFragment:** modificar el tipo del atributo `callback`, pasando este de ser público a privado.
- **ListParadasLineaAdapter:** el constructor crea una variable para retornarla a continuación, por lo que directamente se devolverá el valor sin crear la variable. En el método `onBindViewHolder` se invoca al método `setText` con un parámetro de tipo `int`, lo que deberemos cambiar utilizando `Integer.toString` para que el argumento sea una cadena de caracteres.
- **ViewHolder:** el nombre de la clase deberá ser renombrado, pasándose a llamar `ViewHolderParadasLinea`. Se modificará el tipo de los atributos `textViewNumero` y `textViewNombre` pasando estos de ser públicos a privados.
- **MainActivity:** se eliminará el import que hace referencia a la clase `Log`, ya que no es utilizado durante la ejecución de la clase. Se le añadirá el tipo `transient` a la variable `mBottomBat`. La variable “`_isChangingActivity`” se renombrará para que el primer carácter sea uno que se encuentre en el alfabeto. En el método `onNavigationItemSelected` se encuentran unas series de líneas de código comentadas, que se eliminarán.

Una vez estos cambios hayan sido realizados, la valoración en los tres apartados de sonar será ‘A’ y la deuda técnica junto con los code smells del código introducido se reducirán. Ahora, los datos que se obtendrán en el apartado de mantenibilidad serán, una hora de deuda técnica y ocho nuevos code smells. Los code smell cuya gravedad se marque como “info” no serán tratados.

Aunque la parte nueva del proyecto ya tenga una calidad bastante óptima, hay code smells de la parte antigua del proyecto, que pueden ser solucionados reduciendo así la deuda técnica total. Algunos de estos cambios se realizarán en las siguientes clases o paquetes:

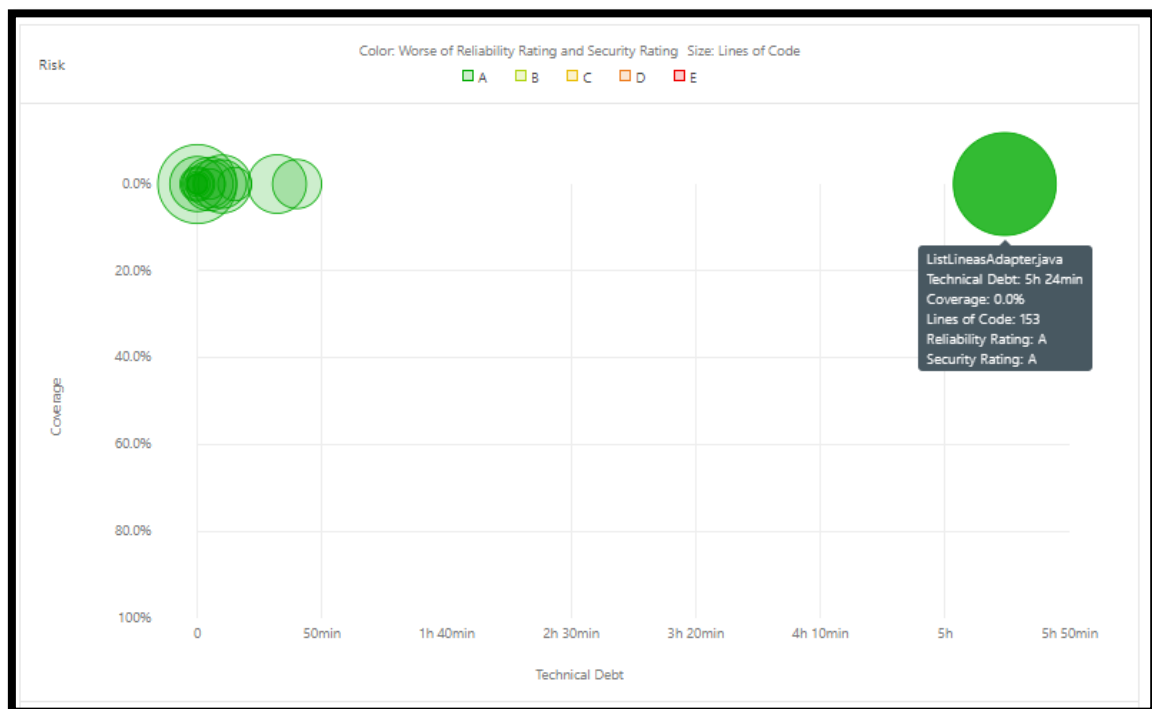
- **ParserJson:** el constructor de esta clase pasará a ser privado, impidiendo así la creación de posibles instancias.
- **Paquete DireccionesFragment:** se deberá cambiar la letra inicial del nombre de este paquete pasando a ser minúscula.
- **DireccionesFragment:** en el método `onCreateView` se crea una variable para retornarla a continuación, por lo que se modificará el código para que devuelva directamente el valor sin crear ninguna variable.
- **Paquete LineasFragments:** se deberá cambiar la letra inicial del nombre de este paquete pasando a ser minúscula.
- **LineasTodasFragments:** en el método `onCreateView` se crea una variable para retornarla a continuación, por lo que se modificará el código para que devuelva directamente el valor sin crear ninguna variable. Se eliminarán líneas de código

comentadas. Se eliminará la variable `DataCommunication` ya que esta no realiza ninguna función dentro de la clase.

- **LineasFavoritasFragment:** en el método `onCreateView` se crea una variable para retornarla a continuación, por lo que se modificará el código para que devuelva directamente el valor sin crear ninguna variable.
- **Paquete ParadasFragments:** se deberá cambiar la letra inicial del nombre de este paquete pasando a ser minúscula.
- **ParadasFavoritasFragment:** en el método `onCreateView` se crea una variable para retornarla a continuación, por lo que se modificará el código para que devuelva directamente el valor sin crear ninguna variable.
- **SplashScreen:** en el método `onCreate` hay un `catch` sin código en su interior, por lo que se deberá añadir un mensaje de error si entra en él.

Unidos estos nuevos cambios al proyecto, la deuda técnica general se reducirá a un día generada por un total de cuarenta code smells, los cuales tienen una gravedad de tipo “info”, o tienen una gravedad mayor o menor, pero consideramos que no deben ser modificados por el momento.

Si observamos ahora la gráfica de hotspots vemos esto:



La gran mayoría de code smells restantes en el proyecto generan una pequeña deuda técnica y no tienen una mala valoración. El único que no cumple estas condiciones se produce en `ListLineasAdapter`. Es un “if infernal” generado a la hora de colorear las diferentes líneas de autobuses, se intentará solucionar en el futuro.

Tanto en el código antiguo como en la parte nueva que se ha implementado, no se encontrará ninguna duplicación.

El código a la hora de realizar este documento no pasa el sonar, esto se debe a un problema con la cobertura de las pruebas, aunque estas no deberían ser miradas. Al obtenerse una calificación de 'A' en los tres apartados, se considera que el código satisface los criterios de calidad establecidos.