Plan de Mantenimiento

Estado inicial del proyecto

Tras haber implementado el plan de pruebas en el proyecto del Sistema Bancario, hemos obtenido una cobertura del 87% en líneas de código, y un 98% en cobertura de decisiones. Hemos detectado los siguientes errores, (al acabar la fase de testing hemos detectado una gran cantidad de errores vamos a identificar los más importantes):

1) Error en el método Cuenta.transferir(): Hemos detectado 2 errores.

- **a.** Error de lógica de negocio: Cuando se realiza una transferencia a una cuenta inexistente, se desquita el dinero de la cuenta original (más la comisión), cuando no debería ocurrir al no poder completarse la transferencia.

- **b.** También se trata de un error de lógica de negocio: Cuando se realiza una transferencia, no se comprueba que la cuenta de origen tenga la cantidad a transferir más la comisión, sólo se comprueba la cantidad a transferir. Esto no debería ocurrir, ya que el programa debería lanzar SaldoInsuficienteException y no se debería de realizar la transferencia.

2) Error en el método Cuenta.insert(): Error de lógica de negocio. Cuando se crea una cuenta con un número ya existente, debe de saltar la siguiente excepción: CuentaYaCreadaException, que indica que ya hay una cuenta creada con ese ID.
3) Errores en los comentarios de las clases que conforman el programa.
4) Errores de Checkstyle.
5) Errores de corchetes en bucles if y for, en algunos bucles no aparecen, al tener sólo una sentencia.
6) Errores en nombres de variables y métodos (algunos nombres de variables coinciden con el nombre de algunos métodos).
7) Algunas variables sólo se usan una vez y no están asignadas como finales.
8) Comentarios requeridos en formato Javadoc
9) Faltan constructores por defecto.
10) Faltan métodos de acceso a los atributos de los objetos.
11) Las clases con constructores privados deben ser finales.
12) Tiempo excesivamente largo de ejecución del pom.
13) Errores encontrados por PMD

Definición del plan de mantenimiento

Estudio de los cambios posibles

Cambio	Error que arregla	ID del cambio
Cambio de caracteres especiales en Javadoc	Impide la correcta generación de informes necesarios para realizar las pruebas y el mantenimiento	0
Cambio en la implementación del método	Cuenta.transferir() (1)	1
Cambio en la implementación del método	Cuenta.transferir() (2)	2
Cambio en la implementación del método	Cuenta.insert()	3
Cambio en la agregación de corchetes	Errores en bucles	4
Cambios en el nombre de algunas variables	Errores en nombres de variables	5
Modificación de algunas clases	Error algunas clases (clases con constructores privados deben de ser finales)	6
Cambio del código	Errores encontrados por PMD	7
Cambio de la codificación del pom.xml	Tiempo de ejecución excesivo del pom	8
Cambio en algunos fragmentos del código	Errores de Checkstyle	9

Análisis de viabilidad

Hemos detectado una gran cantidad de errores: errores de lógica de negocio (errores en los métodos insert y transferir), errores de estilo, errores en comentarios, bucles, algunas clases… que, si ha sido viable arreglarlos, sin embargo, hemos encontrado errores de *Checkstyle*, que hemos conseguido disminuir, pero no ha sido viable arreglarlos todos, ya que al eliminar algunos errores aparecían otros.

Registro de aprobación de cambios

Cambio (ID)	Prioridad	Responsable	Estado
0	1	Francisco Gaspar Pérez Rodríguez	Aprobado
1	1	Antonio Patón Rico	Aprobado
2	1	Álvaro Pardo Benito	Aprobado
3	1	Edilberto Pozo Pozo	Aprobado
4	3	Edilberto Pozo Pozo	Aprobado
5	2	Álvaro Pardo Benito	Aprobado
6	2	Antonio Patón Rico	Aprobado
7	2	Francisco Gaspar Pérez Rodríguez	Aprobado
8	3	Francisco Gaspar Pérez Rodríguez	Aprobado
9	3	Francisco Gaspar Pérez Rodríguez	Aprobado

Correcciones que se deben llevar a cabo para realizar los cambios

1) Se controló que la cantidad que debe de haber como mínimo en la cuenta origen debe de ser la del importe de la transferencia + la comisión.

2) Controlamos que la cuenta a la que hacemos el ingreso exista, en caso contrario, no se debe retirar el saldo de la cuenta origen. Utilizamos el método isPresent().

3) Controlamos que, al crear una cuenta con el mismo número, nos salte una excepción que nos diga que esa cuenta ya está creada.

4) Algunas sentencias "if" o "for" al tener sólo una sentencia, no se incluyen corchetes, lo correcto es que todas estas sentencias tengan corchetes, y se arregla poniendo corchetes en todos los bucles, que indiquen donde empiezan y donde acaban. (PMD)

5) Se soluciona poniendo diferentes nombres a las variables y a los métodos, ya que hay algunas variables y algunos métodos que se llaman igual.

6) Algunas clases tienen constructores privados, y por lo tanto estas clases deben tener el identificador "final", se soluciona agregándolo a cada clase que tenga private los constructores. (PMD)

7) Verificación de las reglas de codificación: Reducido al mínimo.

8) Reducir tiempo de ejecución del código.

9) Reducir espacios en blanco, reducir longitud de las líneas, reducir espacios finales.

Ejecución del Plan de mantenimiento

Ejecución

Para la ejecución del plan hemos creado una rama en el proyecto llamada *Mantenimiento*, con el objetivo de introducir allí todas las modificaciones necesarias.

Una vez que cada cambio ha sido aprobado, se realiza un *merge* al *origin/master* del proyecto. En el repositorio del proyecto se puede observar esta dinámica de trabajo con más detalle.

Por cada cambio realizado y aprobado se ha generado una nueva versión del producto, siguiendo la implantación del Proceso de Gestión de Configuración que ha realizado la empresa **RTSoftware**. Cada versión está marcada y es descargable desde el propio repositorio del proyecto.

Coste

El mantenimiento del proyecto ha durado una semana y dado el equipo de mantenimiento de la empresa **RTSoftware** (un empleado), su sueldo, y el coste de las operaciones de la propia empresa (todo está detallado en la sección de Estructura Empresarial en nuestra Web), el coste de mantenimiento ha sido de:

$$Coste_{TOTAL} = (Coste_{NETO} + BurnRate_{PERIODO}) * 1.1$$

$$Coste_{TOTAL} = (1600€ + 6490€) * 1.1$$

$$Coste_{TOTAL} = 8.899\ €$$

Situación final

Hemos conseguido detectar y reparar tres errores de lógica de negocio, los cuales han sido mencionados y explicados con anterioridad.

Hemos reducido los 634 errores de *checksytle* al mínimo número posible: **26**.

También se han corregido "posibles errores" de programación detectados por PMD. En este caso, el informe generado es tan exhaustivo que marca como posibles errores, casos que después de ser revisados por nuestro equipo, no lo eran. Aun así, el equipo de mantenimiento de **RTSoftware** ha conseguido reducir en gran medida los errores detectados por PMD.

Una vez realizados todos los cambios, se ha escrito una prueba más adicional, siendo el total de pruebas escritas: **49**. Se ha comprobado que todas las pruebas vuelven a ser pasadas con éxito (pruebas de regresión).

Como consecuencia de las modificaciones y del código añadido, ha variado de forma casi imperceptible la cobertura de líneas de código: **87%** y la cobertura de decisiones: **97%.**

El resultado se puede observar tanto en el repositorio como en el informe final que está disponible online en la Wiki y en la Web del proyecto.

Cabe destacar que la *release* del proyecto que incluye todo el mantenimiento y el informe generado es la versión: **0.2.7**, y como se ha mencionado antes, está correctamente etiquetada y descargable desde el repositorio.