



## PRÁCTICA 5: REFACTORIZACIONES

### 1. Objetivos

- Aprender a aplicar refactorizaciones como mecanismo de mantenimiento preventivo de un software heredado.
- En la medida de lo posible, aplicar las refactorizaciones de forma semiautomática utilizando para ello las herramientas que proporciona Eclipse.
- Practicar el cálculo de métricas de calidad de un producto software.

### 2. Actividades

1. Descargar del código fuente del sistema heredado, disponible en Moodle en el archivo Practica5.zip
2. Comprender y analizar la funcionalidad y el código del sistema heredado. Para ello:
  - a. Leer la siguiente descripción de la aplicación:

La aplicación gestiona las cuentas bancarias de clientes de un banco. Mantiene información de los clientes y las cuentas que tienen contratadas. La cuenta permite realizar operaciones de ingreso y retirada de efectivo, que son almacenadas como movimientos. Además, pueden existir tarjetas vinculadas a las cuentas, que pueden ser de crédito o de débito. A través de dichas tarjetas se pueden realizar operaciones de retirada de efectivo y de pago en establecimiento, aunque existen particularidades en cada caso. Las tarjetas de débito tienen un saldo diario máximo disponible. Las de crédito tienen una cantidad a crédito y los movimientos realizados con ella se guardan mensualmente, realizándose a final de mes una liquidación en la que se hacen efectivos en cuenta y pasan a un histórico de movimientos.
  - b. Obtener el diagrama de clases del código heredado. Podrá realizarse manualmente (usando alguna herramienta tipo MagicDraw) o preferiblemente, por tardar menos tiempo, utilizando una herramienta de generación automática (como el plugin ObjectAid de Eclipse).
  - c. Leer el propio código.
3. Obtener las métricas WMC (y WMC normalizado), CBO, DIT, NOC y CCog de todas las clases de la aplicación excepto de las excepciones.

**Nota:** El modo de calcular las métricas de complejidad (WMC y CCog) se debe indicar en el propio código con comentarios (como se ha hecho en el problema 4) para poder evaluar bien el proceso de cálculo de dicha métrica. En el caso del CBO se deben proporcionar tanto el valor numérico como una enumeración de las clases que contribuyen a dicho valor (para el cálculo del CBO sí se han de tener en cuenta las excepciones).
4. Aplicar un proceso de refactorización al código dado. Se pueden aplicar tanto algunas de las refactorizaciones vistas en clase de teoría u otras que se encuentren en la bibliografía.



Además de las refactorizaciones propiamente dichas, se pueden aplicar otras mejoras que se consideren necesarias para mejorar la calidad del código, por ejemplo, mejoras dirigidas a la optimización.

**Nota:** Para llevar a cabo este paso crear otro proyecto Maven independiente, para así realizar después la entrega de ambos proyectos, cada uno con sus métricas.

Cuando se realizan tareas de mantenimiento (como la aplicación de refactorizaciones) es muy importante comprobar que los cambios realizados no introducen nuevos errores en el código. Por ello, después de cada refactorización aplicada, se han de ejecutar el conjunto de pruebas de regresión que se hayan establecido. En este caso, el conjunto de pruebas de regresión son las incluidas en las clases de prueba JUnit dentro de la carpeta test, que se proporcionan ya implementadas. A consecuencia de las refactorizaciones introducidas, puede ser necesario modificar el código de las clases de prueba (no los casos de prueba en sí, sino el modo de implementarlos), o incluso añadir nuevas clases de prueba si aparecen clases nuevas (esto último no es necesario hacerlo en esta práctica).

5. Análisis de la mejora obtenida
  - a. Realizar el diagrama de clases final una vez aplicadas las refactorizaciones.
  - b. Obtener las métricas WMC (y WMC normalizado), CBO, DIT, NOC y CCoG.

### 3. Criterios de Evaluación y Aclaraciones

Se deberá entregar un fichero comprimido que contenga los siguientes elementos:

- Proyectos Maven completos (sin refactorizar y refactorizado) debidamente exportados. Recordad indicar en el código el modo de calcular las métricas de WMC y CCoG.
- Un informe que describa:
  - La lista de refactorizaciones aplicadas, incluyendo el orden y una pequeña explicación de cada una.
  - La situación inicial y final del sistema, incluyendo sus correspondientes diagramas de clases y valores para las métricas.
  - Un análisis razonado sobre si las diferencias en los valores obtenidos para las métricas reflejan las mejoras de calidad introducidas en el código.

*Carlos Blanco  
Patricia López*