*INFORME PRÁCTICA 5A*

* *LISTA DE REFACTORIZACIONES:*

Se aplica la refactorización en las siguientes clases modificando el sistema software ya existente para mejorar la estructura externa conservando el comportamiento interno.

* ***Clase Cliente:***

-*Preplace Magic Number with Symbolic Constant:* Definir la constante CERO\_EUROS con el número de euros igual a 0.0.

-En el método getSaldoTotal() en el for que recorre las Cuentas se cambia el el else if por un nsimple else ya que una vez comparada que la cuenta es o no CuentaAhorro solo quedaría el caso de que fuese CuentaValores.

-*Extract Class*: Vemos que la clase Cliente agrupa propiedades de diferentes dominios, para solventar el problema se desarrolla la clase Dirección, que constará con los atributos calle, zip, localidad. Para relacionar ambas clases la clase Persona tendrá un ArrayList de Dirección llamada direcciones.

* ***Clase Crédito:***

-*Perplace Magic Number with Symbolic Constant*: Definir las constantes COMISIÓN y PRECIO\_INICIAL, con los valores 0.05 y 0.00 respectivamente

-*Reordenación del código*, el bloque del código correspondiente al if con el throw y el else se sube a continuación de la lanzada de la primera excepción. Evitando así ejecutar un else y por tanto el WMC disminuye, pues pasa de ser 4 a ser 3. Contabilizando los dos if y el add a los movimientos mensuales.

-Cambiar nombre de las variables por nombres más descriptivos. Cambio del nombre de los atributos: mCredito a crédito, mMovimientosMnesuales a movimientosMensuales y mhistoricoMoviminetos a historicoMovimientos.

En los métodos, realizamos cambios en algunos de los parámetros como es el caso del método retirar al que se le pasa un valor x; dado que es una constante poco descriptiva, la cambiamos el valor por cantidad, al igual que en el método pagoEnEstablecimiento. En el método liquidar cambiamos la constante r por total por el mismo motivo que en el caso anterior.

* ***Clase Débito:***

Cambio en el nombre del parámetro pasado al método retirar; x por cantidad para hacer el método más explicativo. Al igual que en el método pagoEnEstablecimiento.

* ***Clase Cuenta Ahorro***

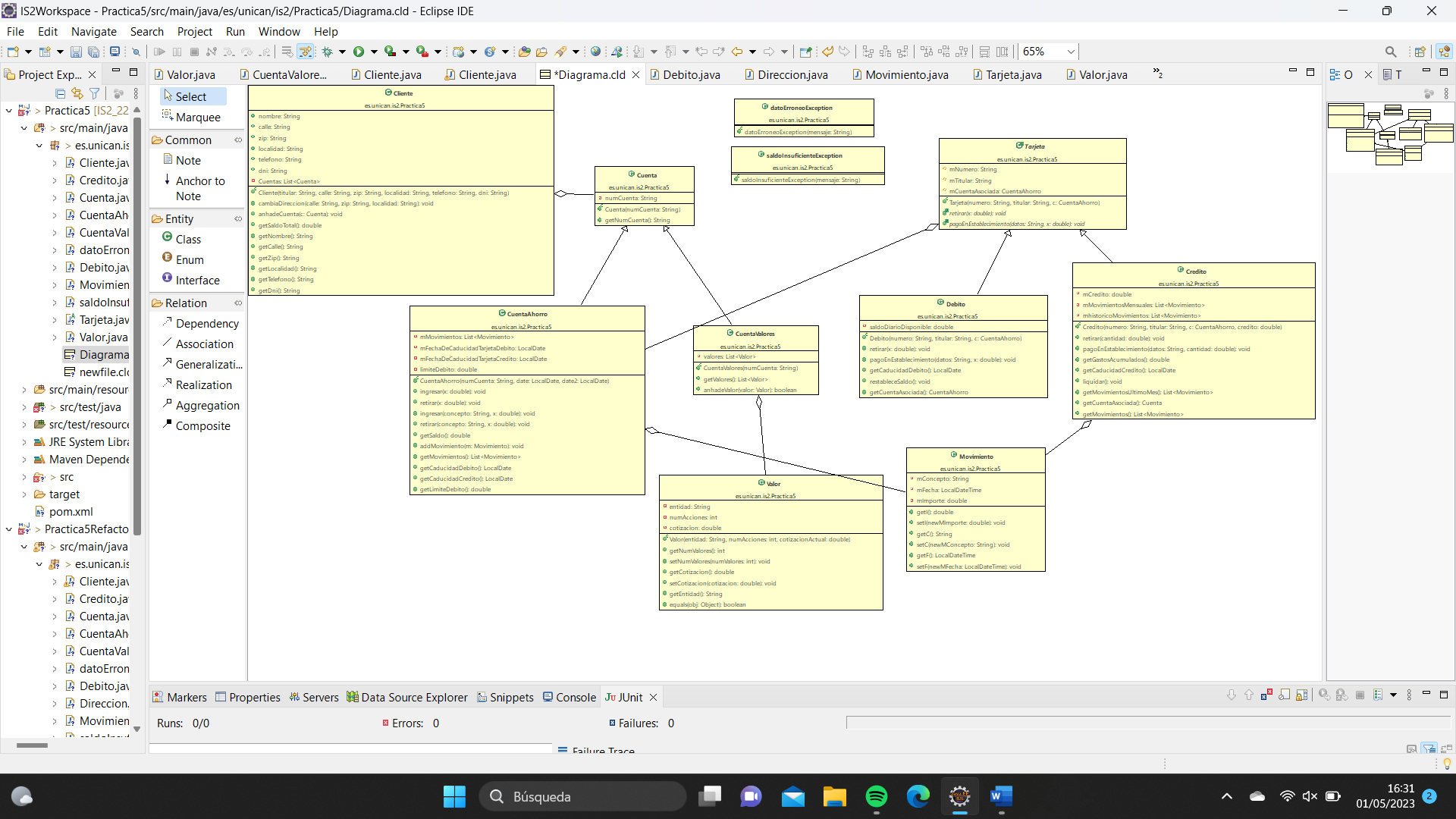
Definir las constantes LIMITE\_MIL Y SALDO\_INICIAL con valores 1000 y 0.0 respectivamente

* *SITUACIÓN INICIAL Y FINAL DEL SISTEMA:*

En la situación inicial el proyecto contiene 11 clases donde 2 de ellas son excepciones lanzadas en las distintas clases. Por el contrario, en la versión refactorizada podemos ver que tenemos 12 clases siendo dos de ellas las clases de las excepciones.

Se han analizado las métricas de cada clase que aparecen comentadas en el código tanto de cada método como de la clase al completo. Para ver la diferencia entre la versión refactorizada y la original, reflejamos los valores totales de las clases con cada una de las métricas en las tablas inferiores.

Mostramos también los diagramas de clases de las dos versiones donde se pueden apreciar algunos cambios como es el caso de los nombres pasados como parámetros o creación de la clase Dirección.

***Versión original:***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Métricas | | | | | | |
|  | WMC | WMCn | CBO | DIT | NOC | CCOG |
| Cliente | 14 | 1.4 | 3 | 0 | 0 | 7 |
| Crédito | 16 | 1.77 | 3 | 0 | 0 | 7 |
| Cuenta | 2 | 1 | 0 | 2 | 0 | 0 |
| CuentaAhorro | 18 | 1.63 | 1 | 0 | 1 | 7 |
| CuentaValores | 6 | 2 | 1 | 0 | 1 | 2 |
| Débito | 6 | 1 | 1 | 1 | 0 | 2 |
| Movimiento | 6 | 1 | 0 | 0 | 0 | 0 |
| Tarjeta | 3 | 1 | 1 | 0 | 2 | 0 |
| Valor | 7 | 1 | 1 | 0 | 0 | 1 |

***Versión refactorizada:***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | WMC | WMCn | CBO | DIT | NOC | CCOG |
| Cliente | 10 | 1.67 | 4 | 0 | 0 | 7 |
| Dirección | 5 | 1 | 0 | 0 | 0 | 0 |
| Crédito | 13 | 1.44 | 3 | 0 | 0 | 5 |
| Cuenta | 2 | 1 | 0 | 2 | 0 | 0 |
| CuentaAhorro | 18 | 1.63 | 1 | 0 | 1 | 7 |
| CuentaValores | 3 | 1 | 1 | 0 | 1 | 0 |
| Débito | 6 | 1 | 0 | 0 | 0 | 0 |
| Movimiento | 6 | 1 | 0 | 0 | 0 | 0 |
| Tarjeta | 3 | 1 | 1 | 0 | 2 | 0 |
| Valor | 7 | 1 | 1 | 0 | 0 | 1 |
|  |  |  |  |  |  |  |

Análisis razonado sobre si las diferencias en los valores obtenidos para las métricas reflejan las mejoras de la calidad introducidas en el código