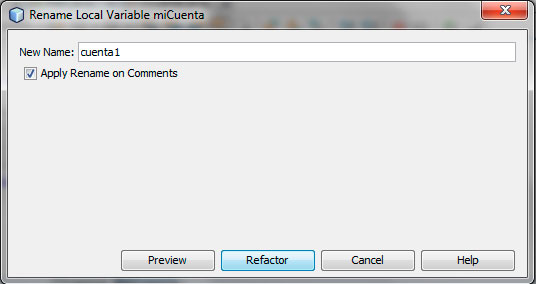
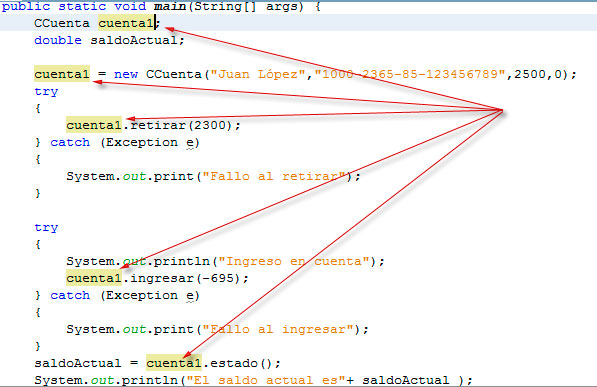


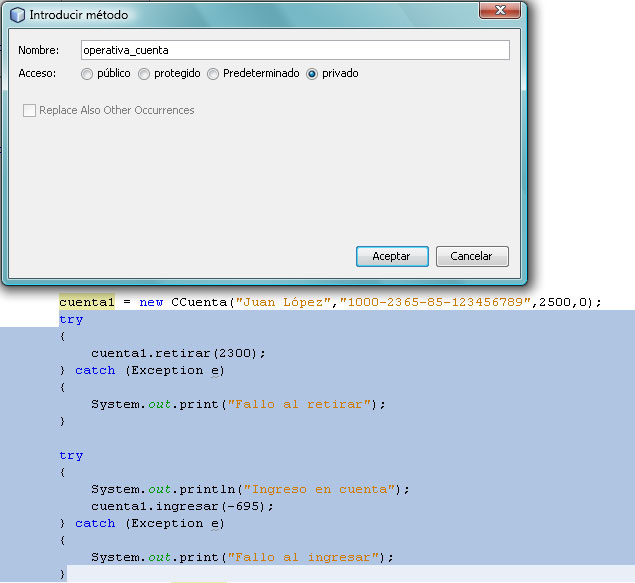
***1*** - Para cambiar el nombre de la variable “miCuenta” por “cuenta1” seleccionamos la variable a cambiar y pulsamos sobre ella con el botón derecho del ratón. Seleccionamos ***Refactor*** y en el nuevo submenú pulsamos en ***Rename***, apareciendo la siguiente pantalla.



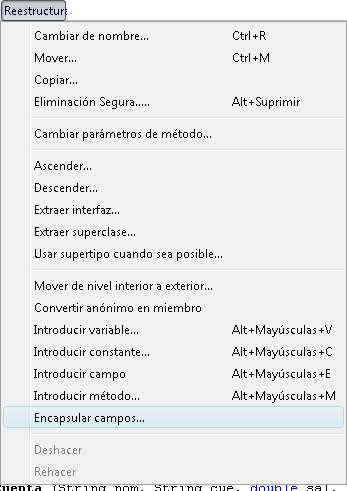
Escribimos “cuenta1” como nuevo nombre y marcamos la casilla que nos permite incluir el renombrado incluso en los comentarios (*por si acaso*). Pulsamos sobre el botón ***Refactor***.



Podemos apreciar como se ha modificado el nombre de la variable en todos los lugares donde antes estaba “miCuenta”.

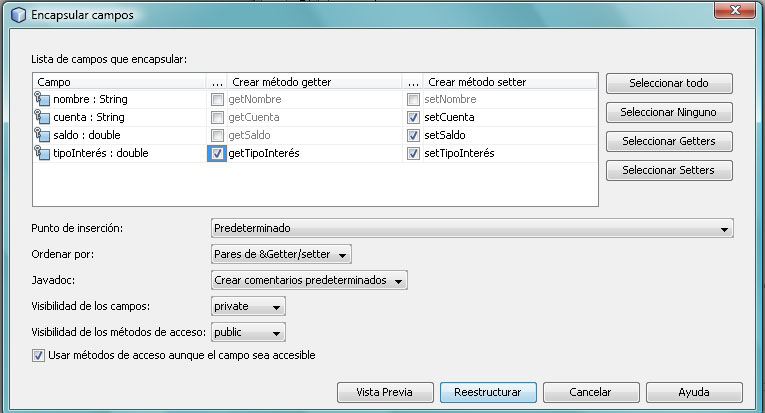


***2*** - Para introducir el método “operativa\_cuenta”, seleccionamos los dos bloques try-catch de main y en el menú ***Refactorizar*** pulsamos sobre ***Introducir método*** donde pondremos el nombre ***operativa\_cuenta*** y pulsamos sobre Aceptar.

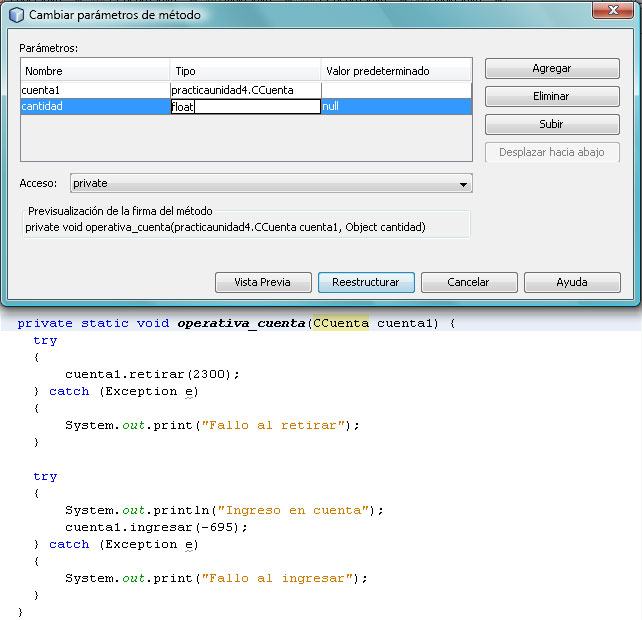


***3*** - Para encapsular los atributos de la clase ***CCuenta*** los pondremos todos de tipo protected con lo que sólo podrán ser accedidos desde la propia clase y desde aquellas que la hereden. También pondremos todos los métodos getter/setter de los atributos de clase, aunque ya existen algunos puestos (asignarNombre, obtenerNombre, etc).

Para hacer esto último procederemos a seleccionar la opción ***Reestructurar*** y escogeremos ***encapsular campos…***



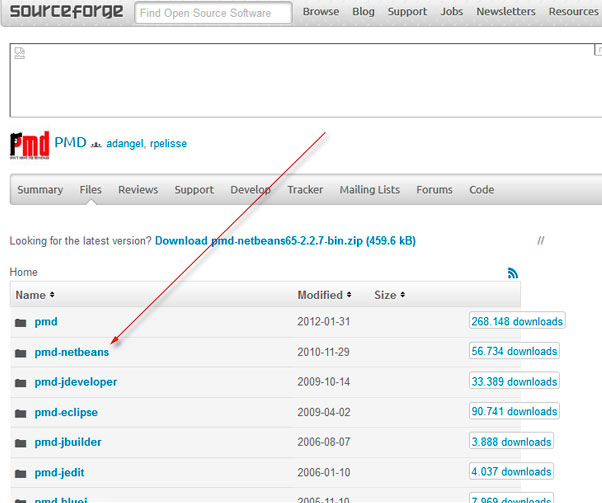
En la nueva pantalla activamos las casillas de los métodos que nos faltan, es decir, el getter de tipoInteres y el setter de cuenta, saldo y tipoInteres. El resto de opciones las dejamos por defecto y pulsamos sobre ***Reestructurar***



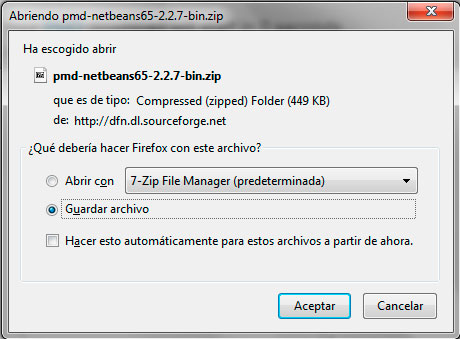
***4*** - Para añadir un nuevo parámetro nos colocamos sobre el método operativa\_cuenta y pulsamos en ***Reesctructurar*** y seleccionamos ***Cambiar parámetro de método***.

Pulsamos el botón ***Agregar*** y tecleamos ***cantidad*** sobre *par1*, ***float*** sobre *Object* y ***0***(cero) sobre *null*. Para cambiar estos valores hacemos doble clic sobre cada uno de los términos.

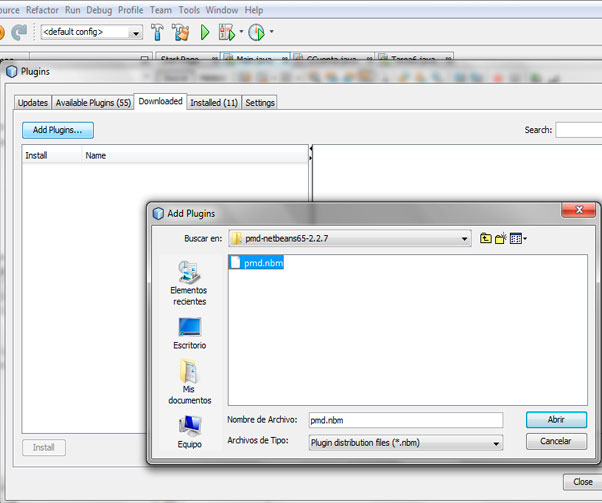
Pulsamos sobre ***Reestructurar*** y ya tendremos el nuevo parámetro añadido al método.



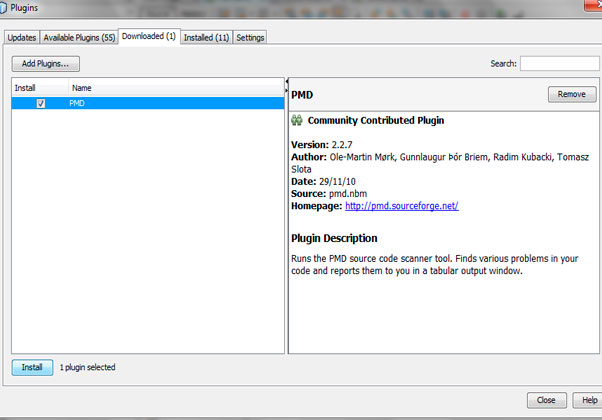
***5*** - Para obtener el plugin PMD vamos a <http://sourceforge.net/projects/pmd> y descargamos la versión para NetBeans



Cuando la hayamos descargado, la descomprimimos (es un fichero zip).

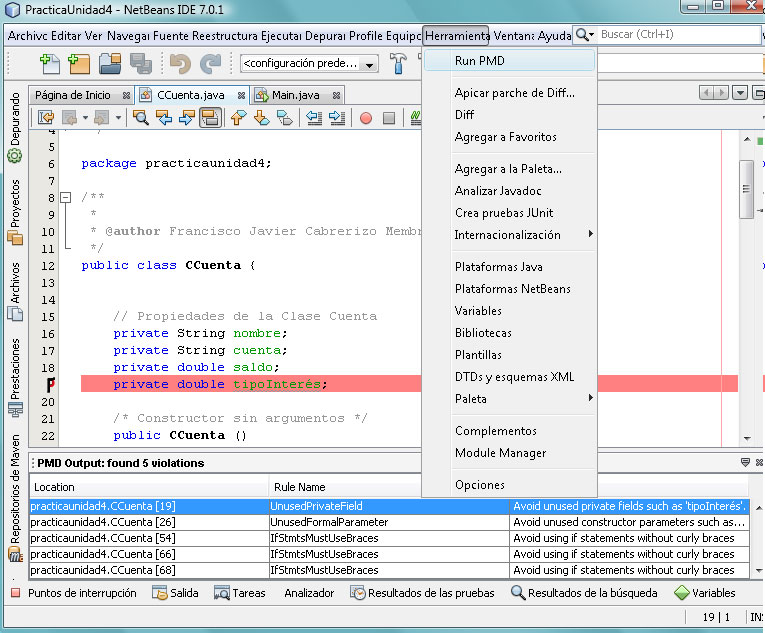


Nos vamos ahora a NetBeans y pulsamos en ***Herramientas / Complementos / Descargado / Agregar Plugins…***

Nos vamos a la carpeta donde hemos descargado el plugin, lo seleccionamos y pulsamos sobre ***Abrir***.

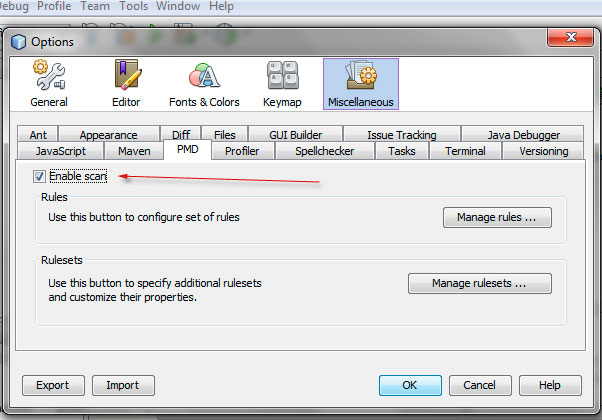
Cuando nos aparece la pantalla de la izquierda, pulsamos en ***Instalar***, aceptamos los términos de utilización y aceptamos instalar un plugin que no está firmado por NetBeans.

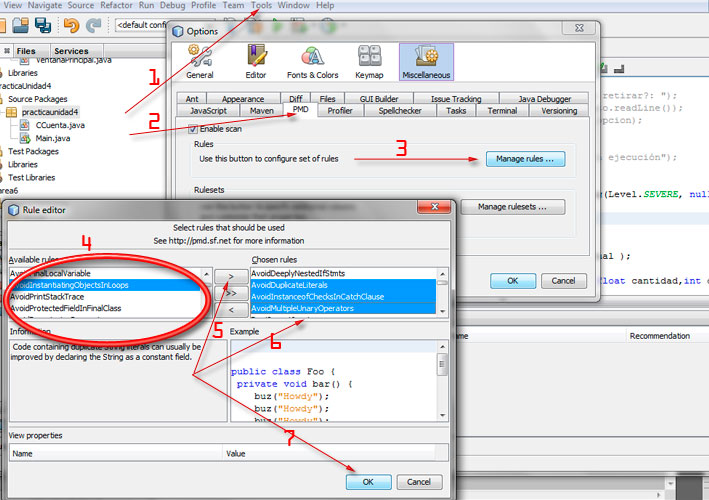
Tras todo este proceso ya tendremos instalado y preparado para usar este analizador de código.



***6*** - Para ejecutar PMD nos situamos sobre el fichero java que deseamos analizar (*en este caso CCuenta.java*) y vamos a ***Herramientas / Run PMD*** .

En este caso nos marca varias “irregularidades” como que tenemos el campo *tipoInteres* sin utilizar en el código, que en el constructor con parámetros hay uno sin utilizar, y tres bloques *if* en los que no utilizamos llaves de apertura y cierre.



***7*** - Si queremos configurar PMD para que escanee de forma automática el proyecto, tendremos que ir a ***Herramientas / Opciones / Varios / PMD*** y marcamos la casilla ***Enable scan***. Al pulsar sobre Aceptar ya lo tendremos preparado.

***8*** - Para añadir tres nuevas reglas tendremos que ir de nuevo a ***Herramientas / Opciones / Varios / PMD***, pero ahora pulsamos sobre ***Manage Rules*** (**3**) y seleccionamos las nuevas reglas que queremos activar (**4**), como por ejemplo en mi caso:

***AddEmptyString***

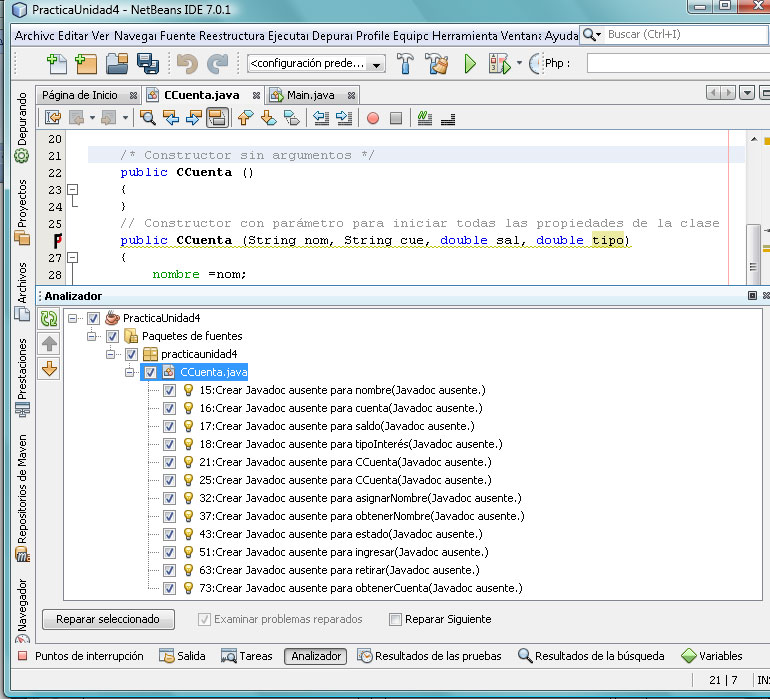
Que nos previene, por ejemplo, que para convertir un entero en cadena lo hagamos de forma correcta (Integer.toString) y no poniendo una cadena vacía a la que añadir el número.

***AtLeastOneConstructor***

Que nos avisará del uso de una clase sin constructores

***AvoidMultipleUnaryOperators***

Nos avisa cuando utilizamos varios operadores unarios seguidos, por ejemplo, - -1 sería igual a 1 por lo que deberíamos poner esto último y no usar la notación con un número doblemente negativo.

Cuando la seleccionamos le damos sobre el botón intermedio que tiene el símbolo de mayor (5) para situarla en la ventana de la derecha (**6**) que es donde nos informa de las reglas que tenemos instaladas. Pulsamos sobre Aceptar (**7**) y de nuevo, en la ventana de Opciones, sobre Aceptar, y ya tendremos en uso las tres nuevas reglas que hemos incluido dentro de todas las que vienen por defecto en el paquete PMD.

***9*** - Para insertar comentarios Javadoc nos iremos a ***Herramientas / Analizar Javadoc*** y en la ventana ***Analizador*** que nos aparece pulsamos sobre la casilla de verificación de la clase CCuenta para que seleccione todos los lugares de la clase donde queremos insertar los comentarios.

Pulsamos sobre ***Reparar Seleccionado*** y nos colocará todos los comentarios javadoc sobre cada método, constructor y campo. Estos comentarios sólo tendrán los parámetros utilizados y las cláusulas return, pero carecerán de información sobre la utilidad de dichos métodos o parámetros, los cuales tendremos que poner nosotros de forma manual.

***10*** - Para generar el informe javadoc de todo el proyecto tendremos que ir a ***Ejecutar / Generar Javadoc*** y tras unos instantes tendremos toda una web llena de enlaces para mostrarnos toda la información relativa a nuestro proyecto (clases que lo componen, métodos de cada una de ellas, constructores, campos, etc). Dentro del proyecto nos habrá creado una nueva carpeta denominada ***dist*** y dentro de ella ***javadoc*** que será la que contenga toda la aplicación web que nos informa de nuestro proyecto y que creamos cuando pulsamos en ***Generar Javadoc***.

***NOTA*** - Para la realización de esta tarea he utilizado dos ordenadores distintos. En uno tenía instalado NetBeans en castellano y en otro NetBeans en inglés, por lo que algunas capturas de pantalla utilizadas en este documento tienen opciones en un idioma y otro, aunque espero resulte lo suficientemente claro.

Incluyo el proyecto completo (Main, CCuenta y javadoc) para que se pueda comprobar.