

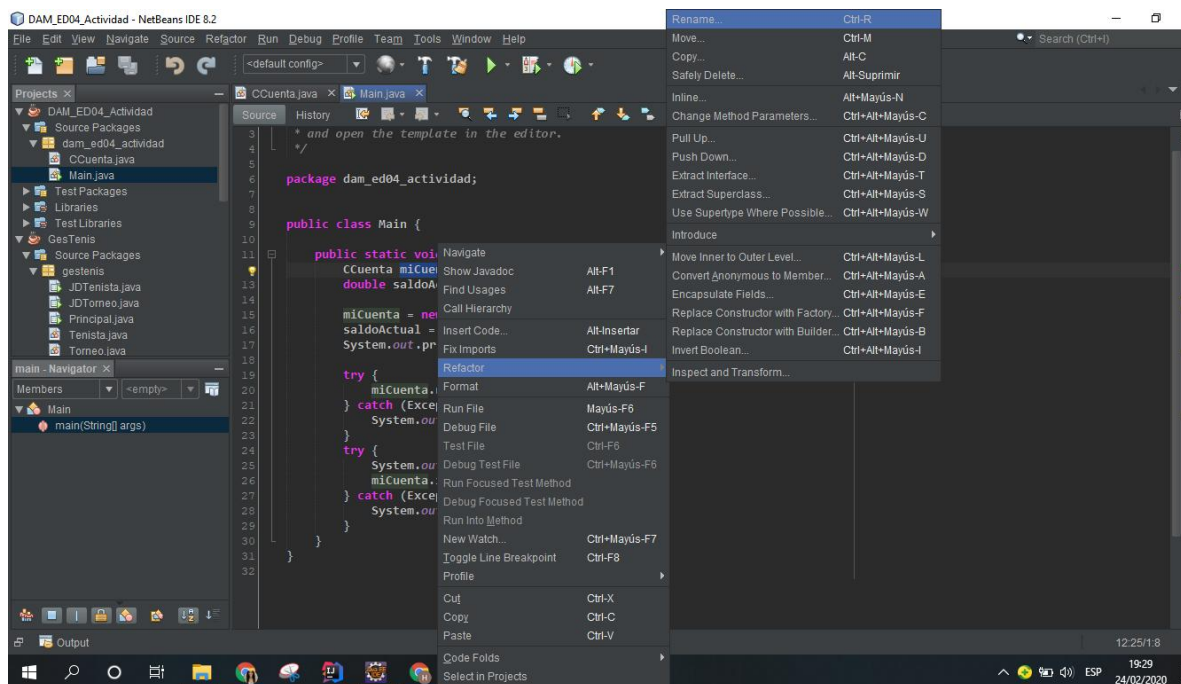
Enunciado.

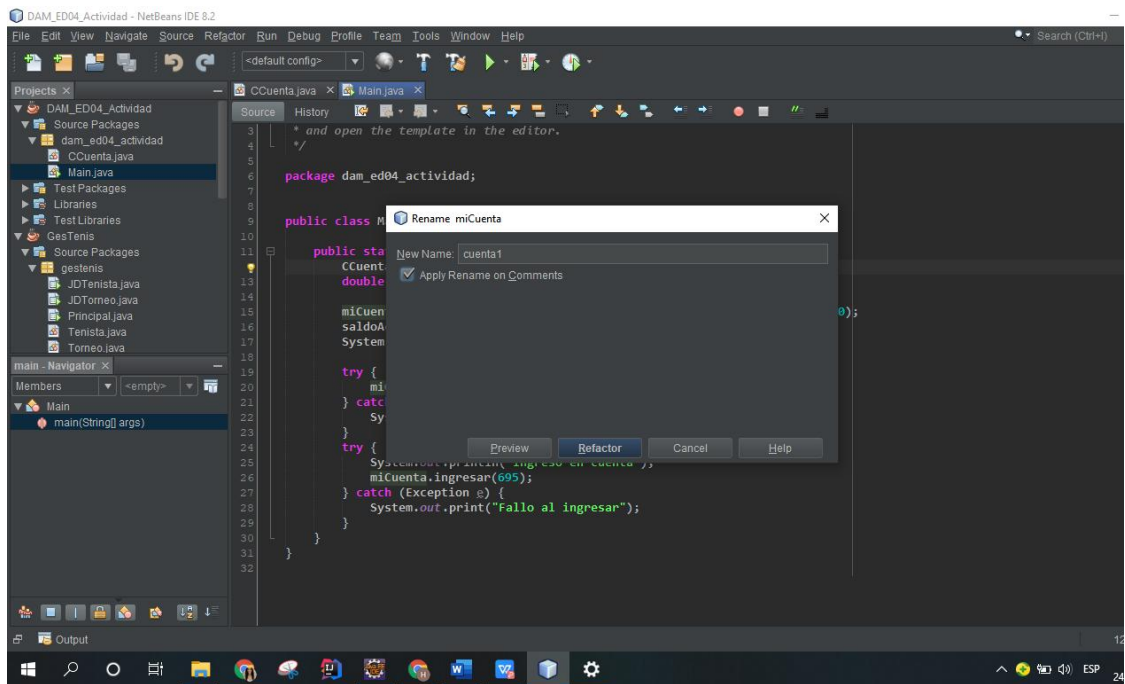
En el proyecto Java que se adjunta, hay definida una Clase llamada **CCuenta**, que tiene una serie de atributos y métodos. El proyecto cuenta asimismo con un Clase **Main**, donde se hace uso de la clase descrita. Basándonos en ese proyecto, vamos a realizar las siguientes actividades.

Refactorización

1. Cambia el nombre de la variable "**miCuenta**" por "**cuenta1**".

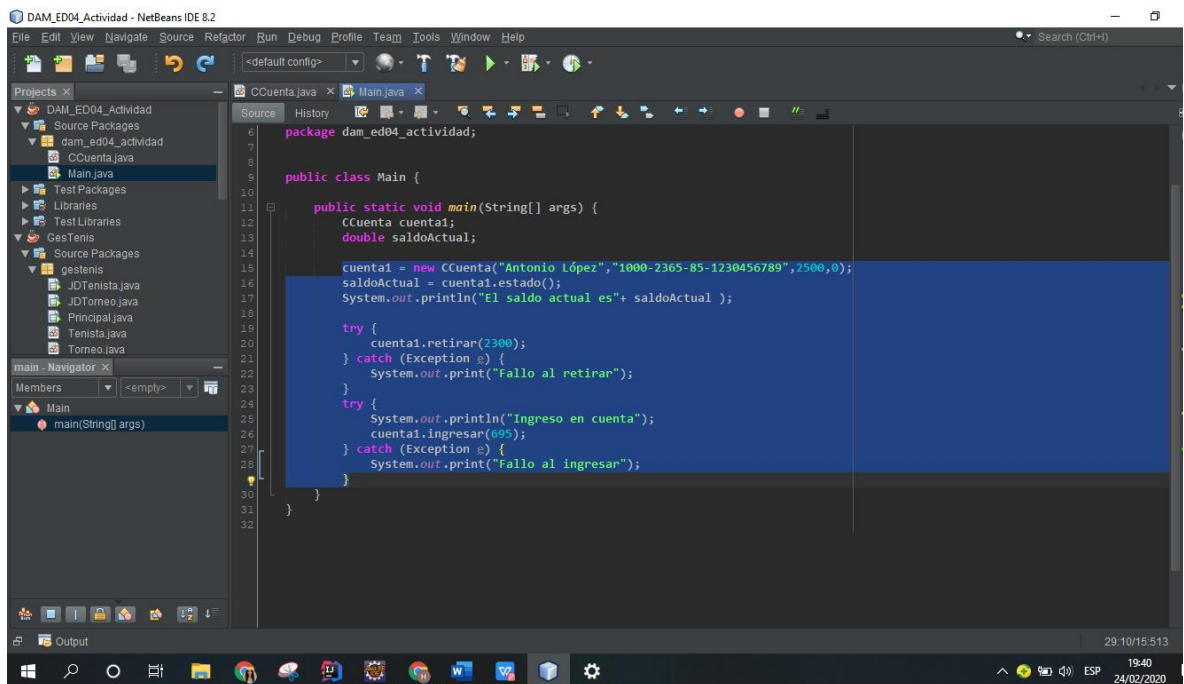
Damos botón derecho y pulsamos dentro de refactor la opción de rename. Ponemos el nombre que queramos y pulsamos aceptar.

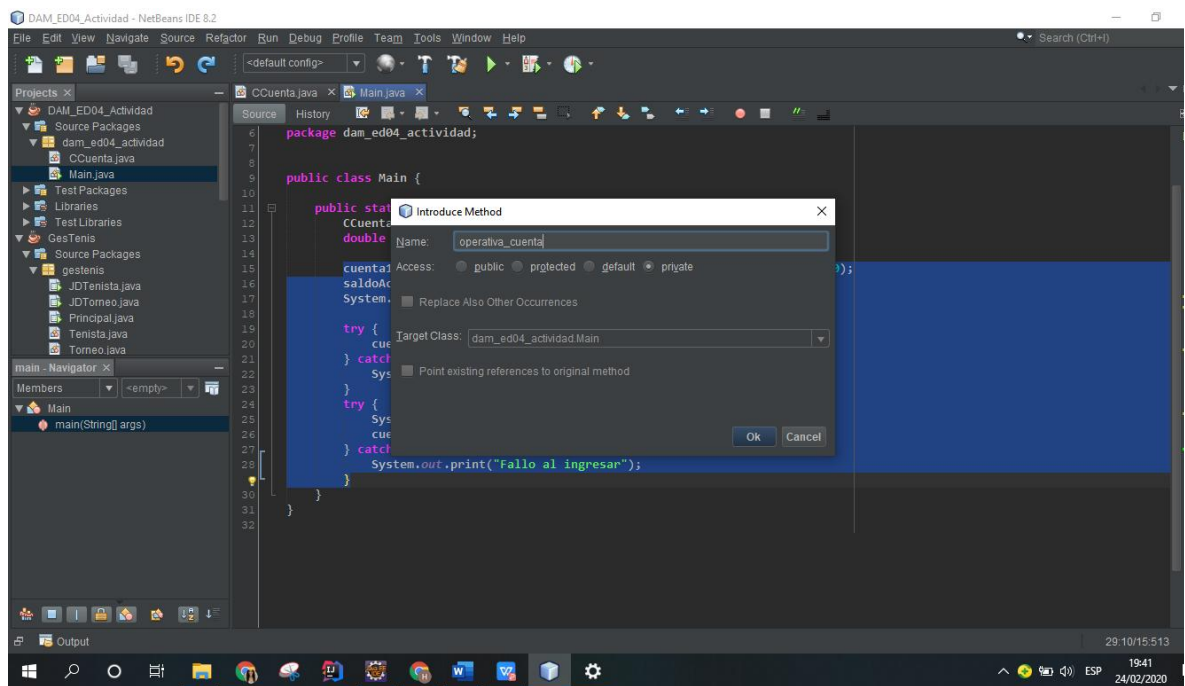
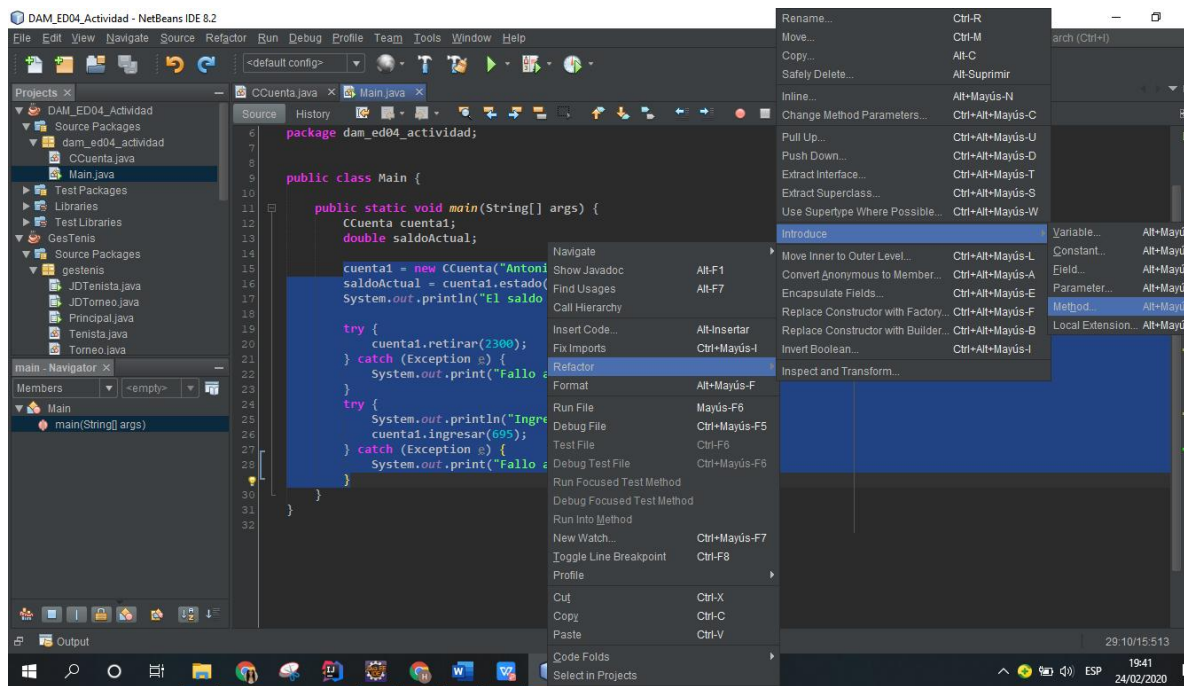


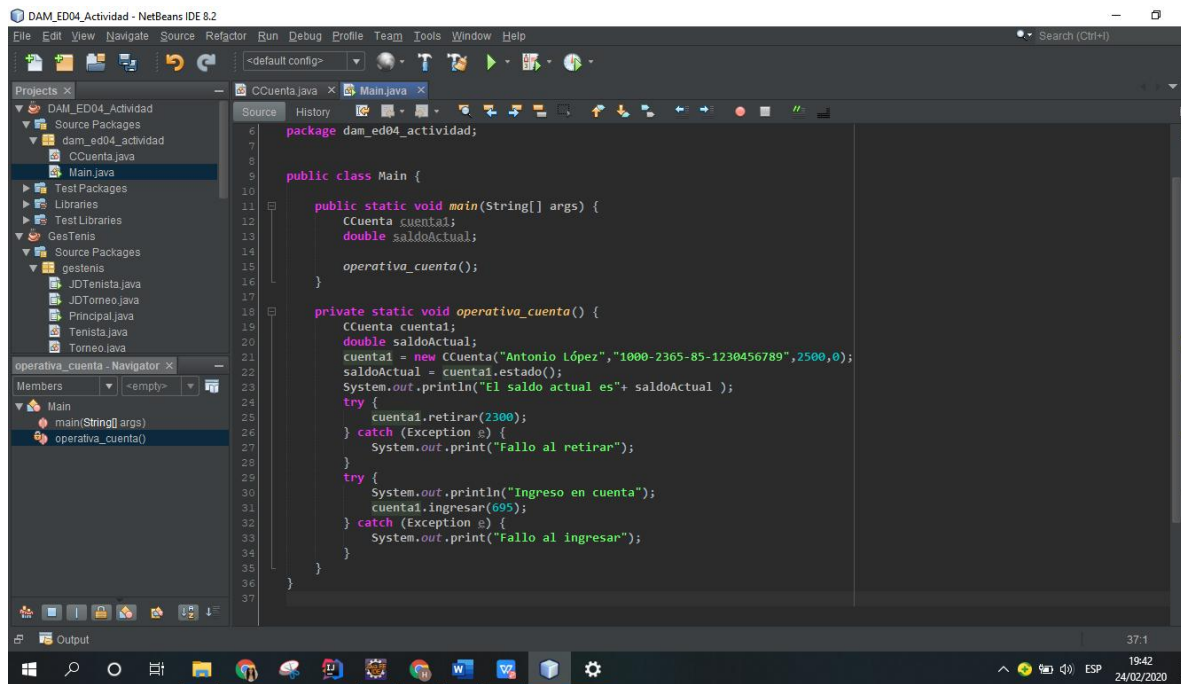


2. Introduce el método **operativa_cuenta**, que englobe las sentencias de la clase **Main** que operan con el objeto **cuenta1**.

Seleccionamos el código que queremos meter dentro de un método. Pulsamos botón derecho, Refactor > Introduce > method. Ponemos el nombre del método que queremos en este caso "operativa_cuenta" y pulsamos aceptar

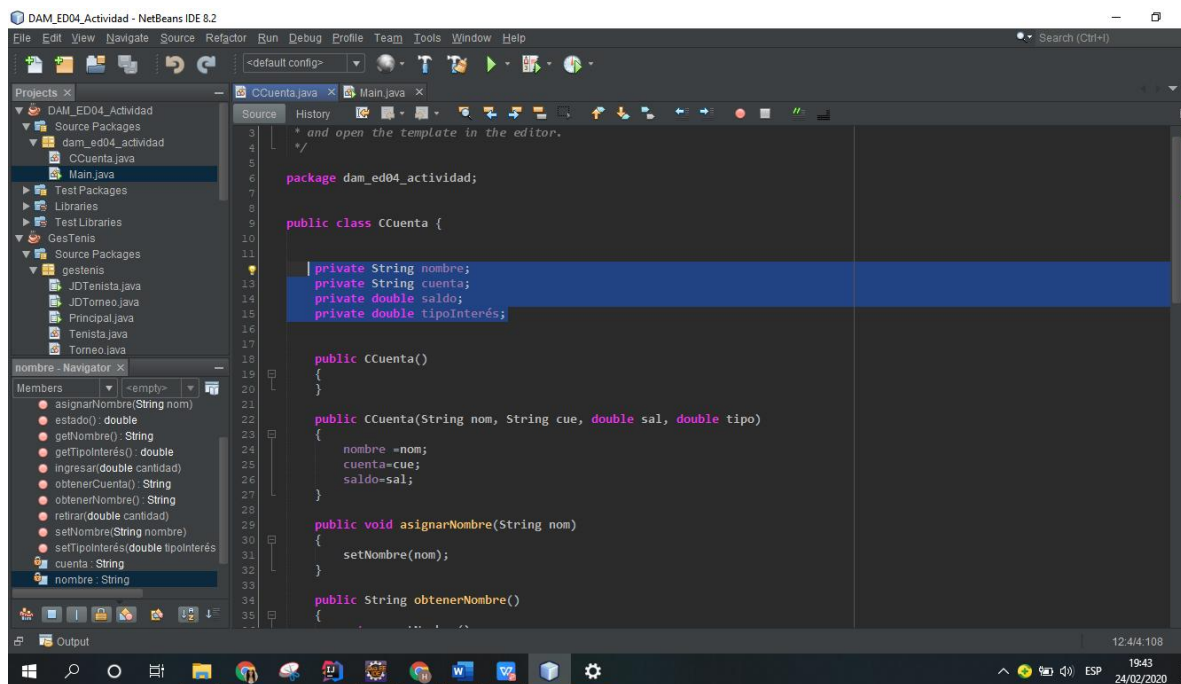


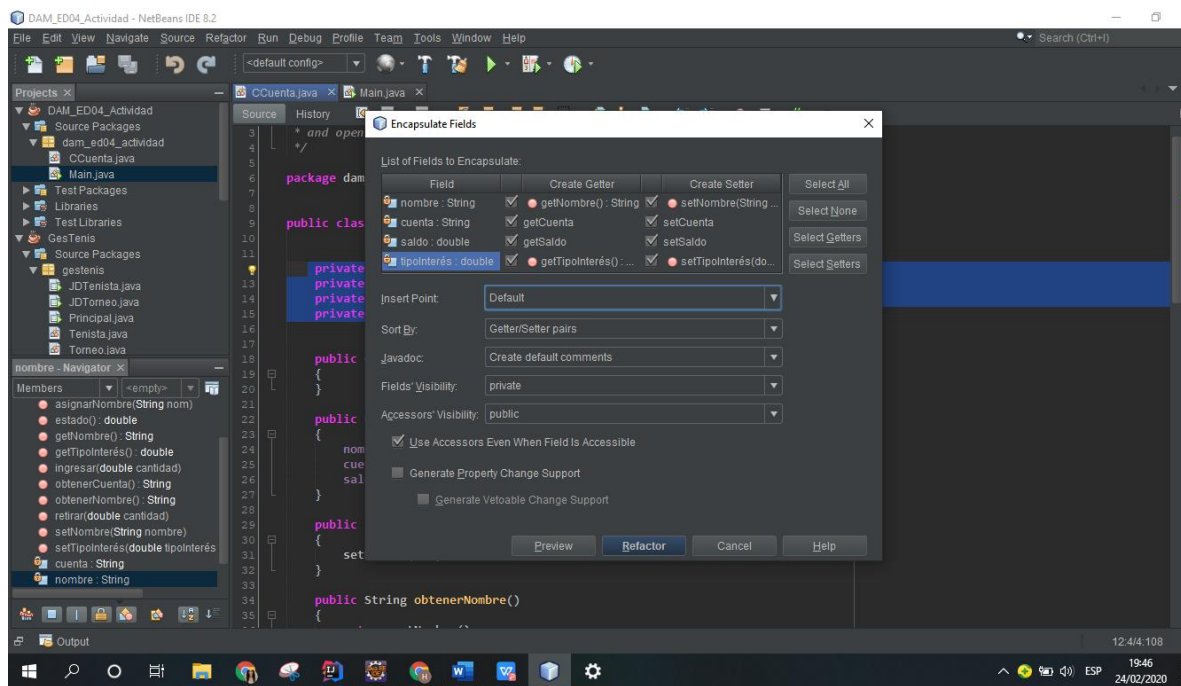
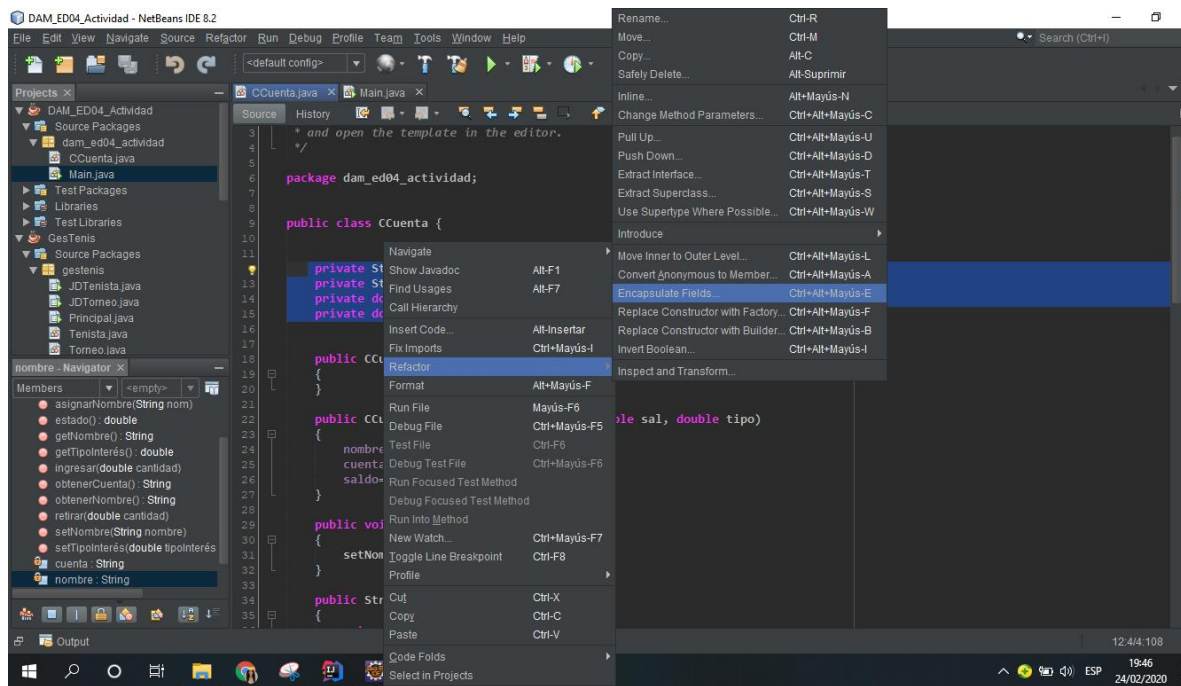


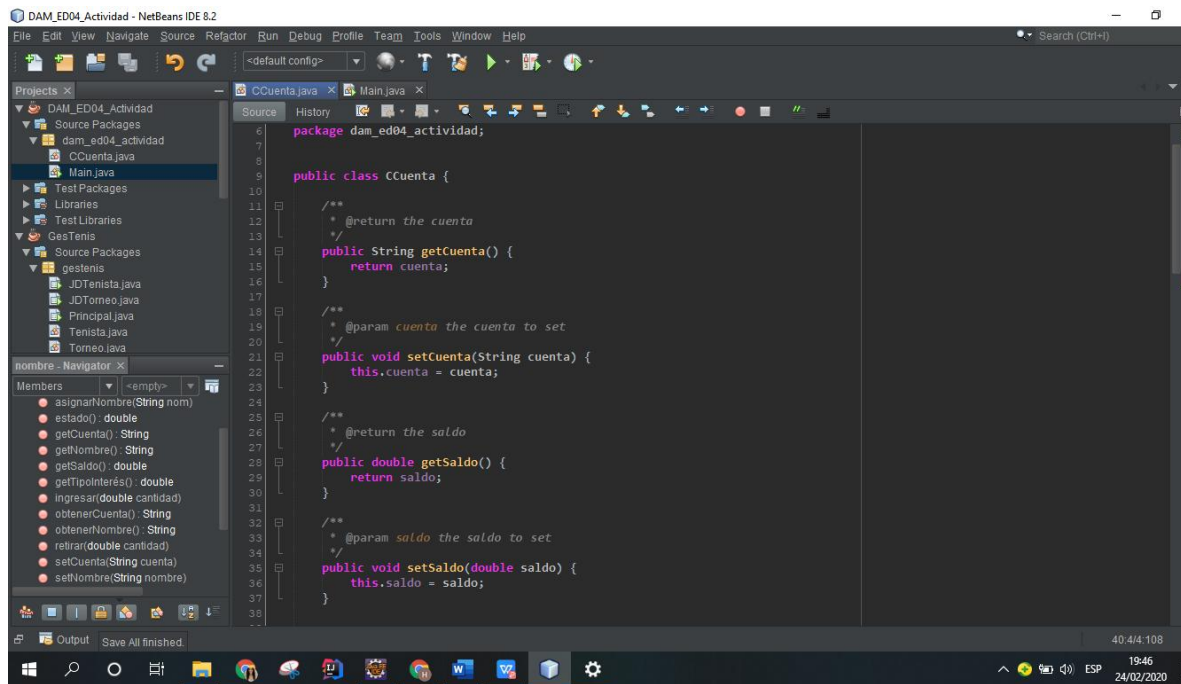


3. Encapsula los cuatro atributos de la clase **CCuenta**.

Seleccionamos los atributos que queremos encapsular. Pulsamos botón derecho y seleccionamos "Encapsulate fields". Comprobamos que los campos "fields visibility" estén en privados y "accessors visibility" en public. Damos aceptar

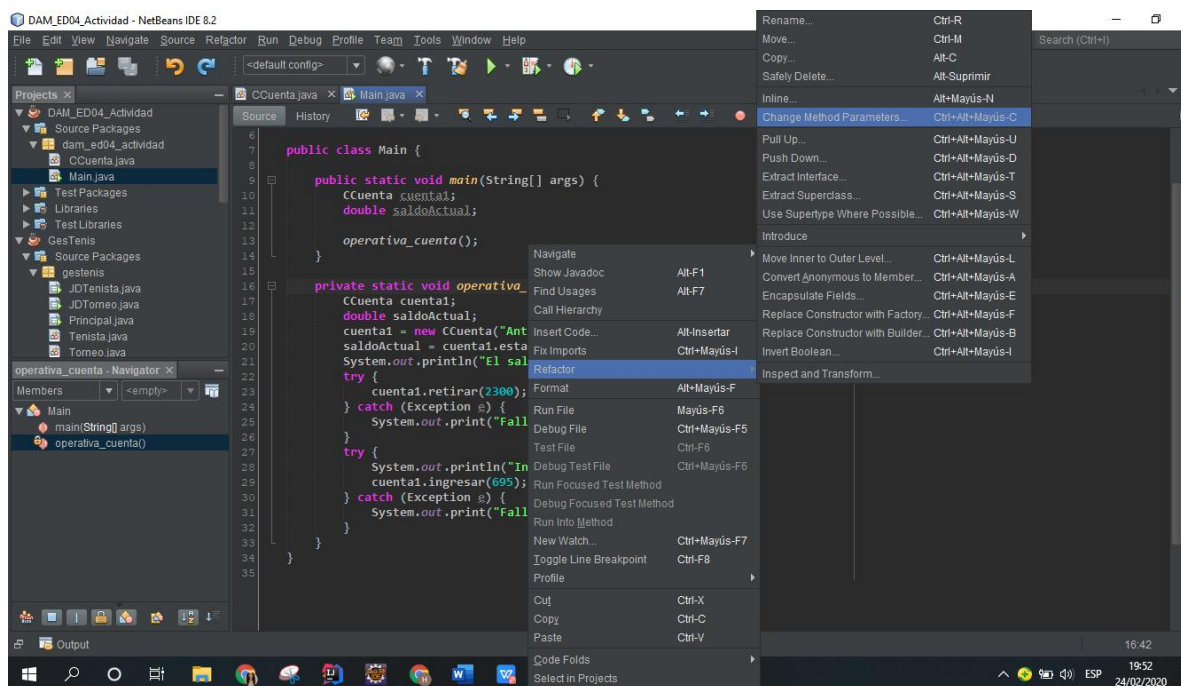


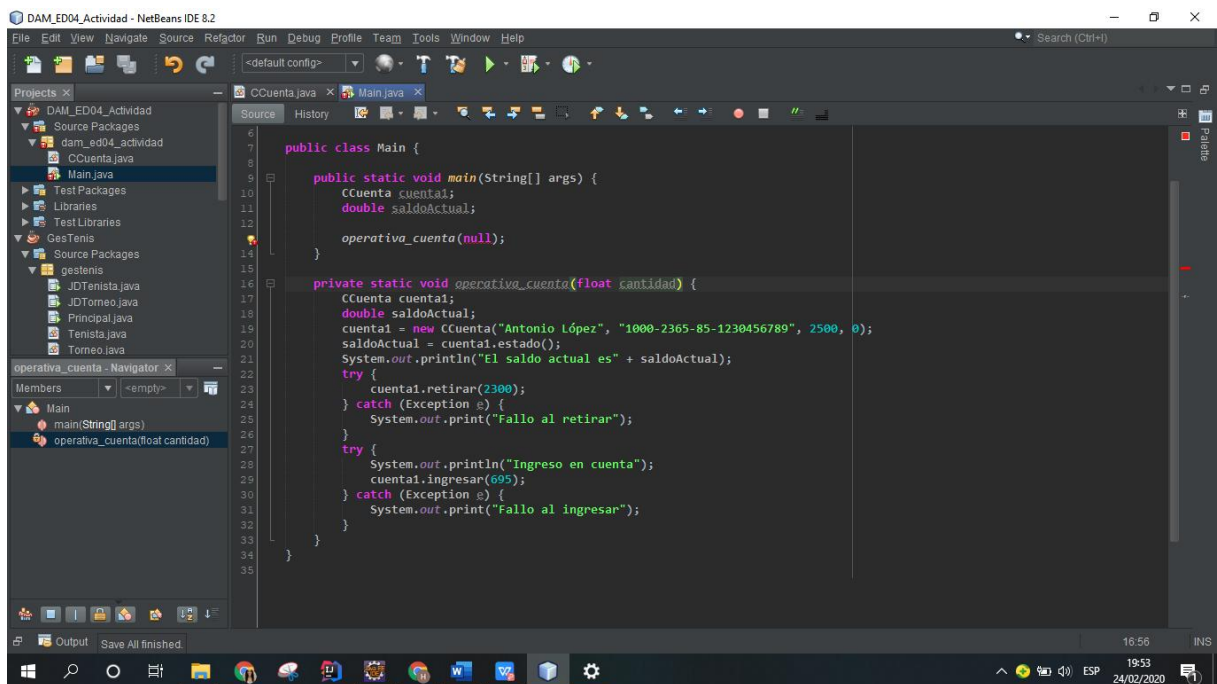
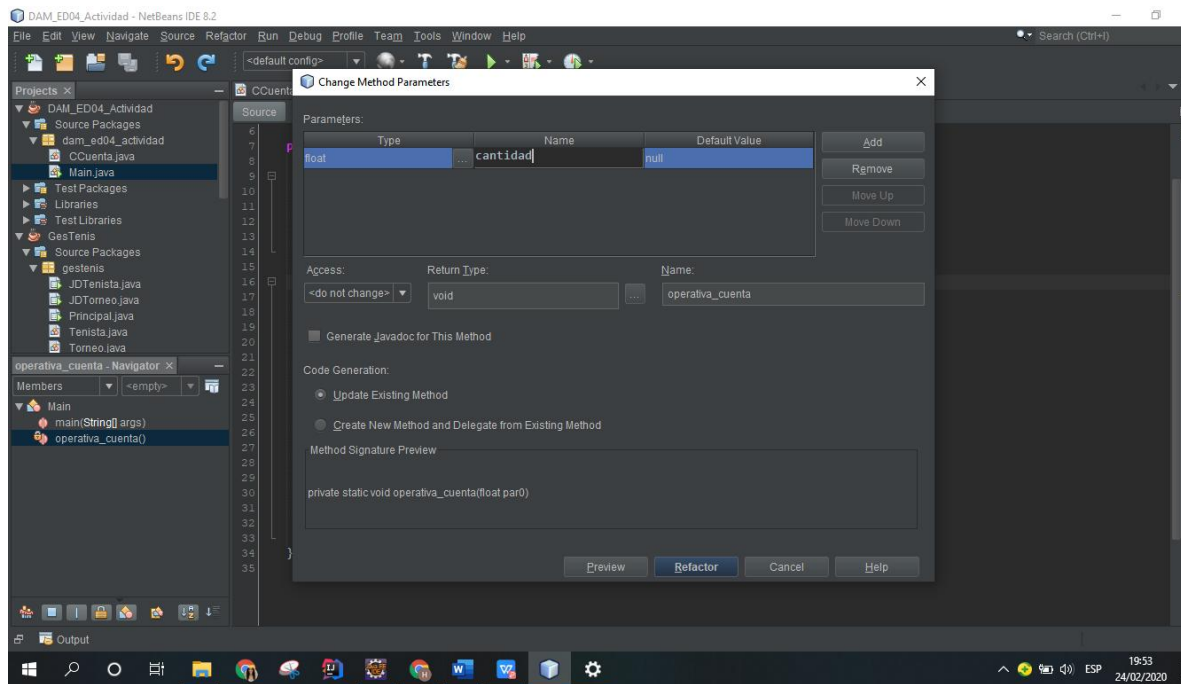


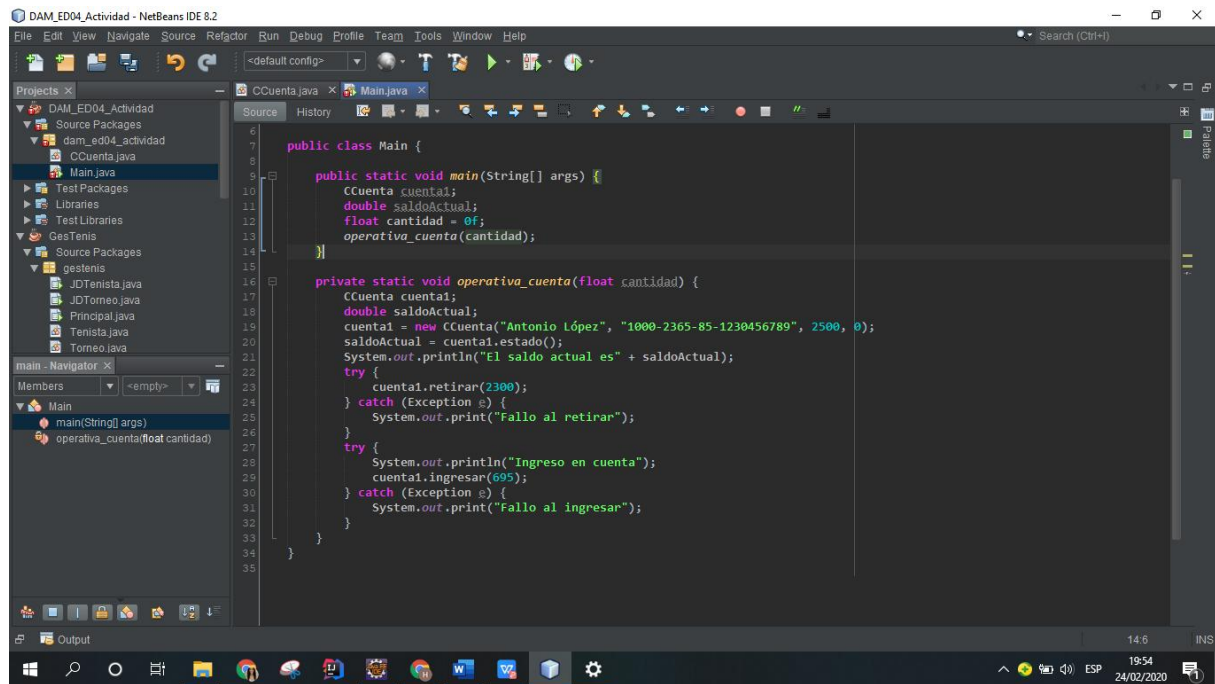


- Añade un nuevo parámetro al método `operativa_cuenta`, de nombre `cantidad` y de tipo `float`.

Pulsamos botón derecho encima del nombre del método. Refactor > change method parameters. En la ventana que se abre pulsamos "add.." y seleccionamos el nuevo parámetro y su tipo, en nuestro caso float cantidad y pulsamos aceptar.





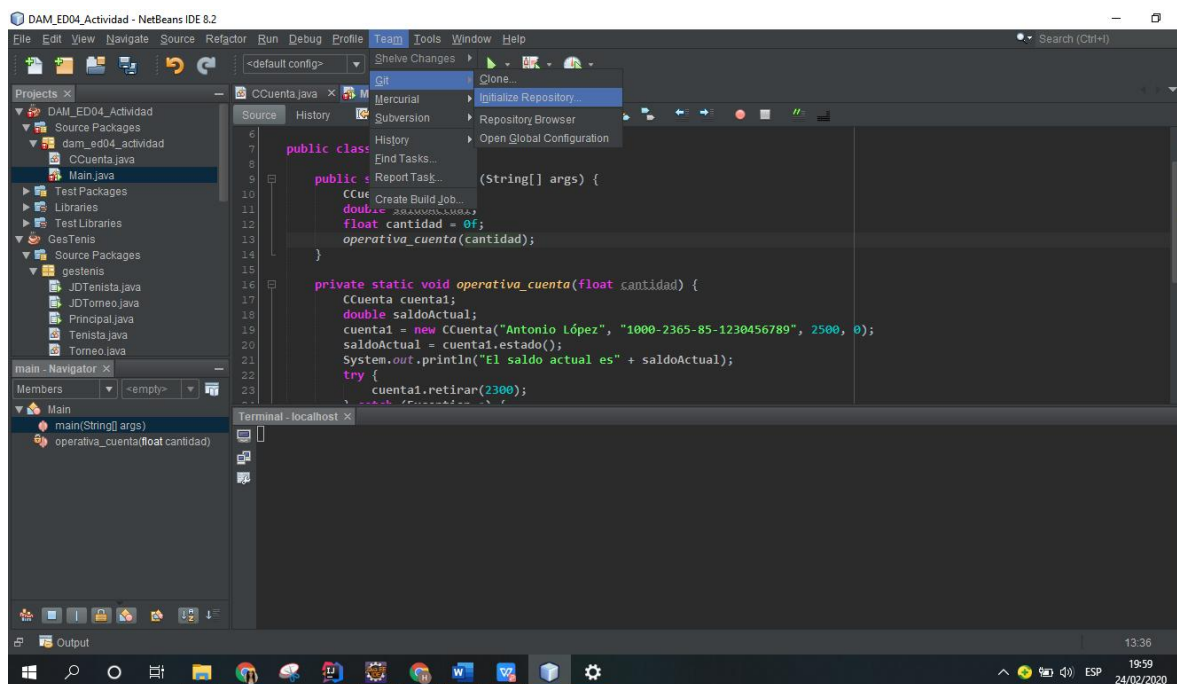


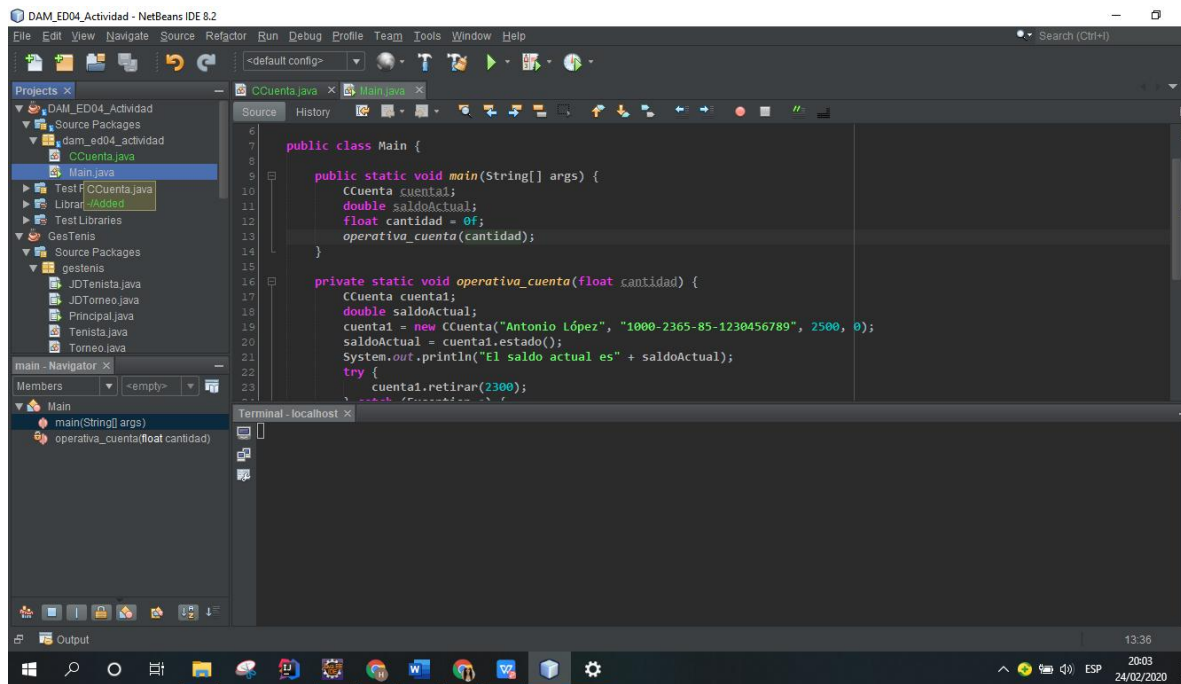
GIT

1. Crea un repositorio Git para el proyecto DAM_ED04_Actividad en la propia carpeta donde lo tengas.

Para crear el repositorio pulsamos Team > Git > Initialize Repository...

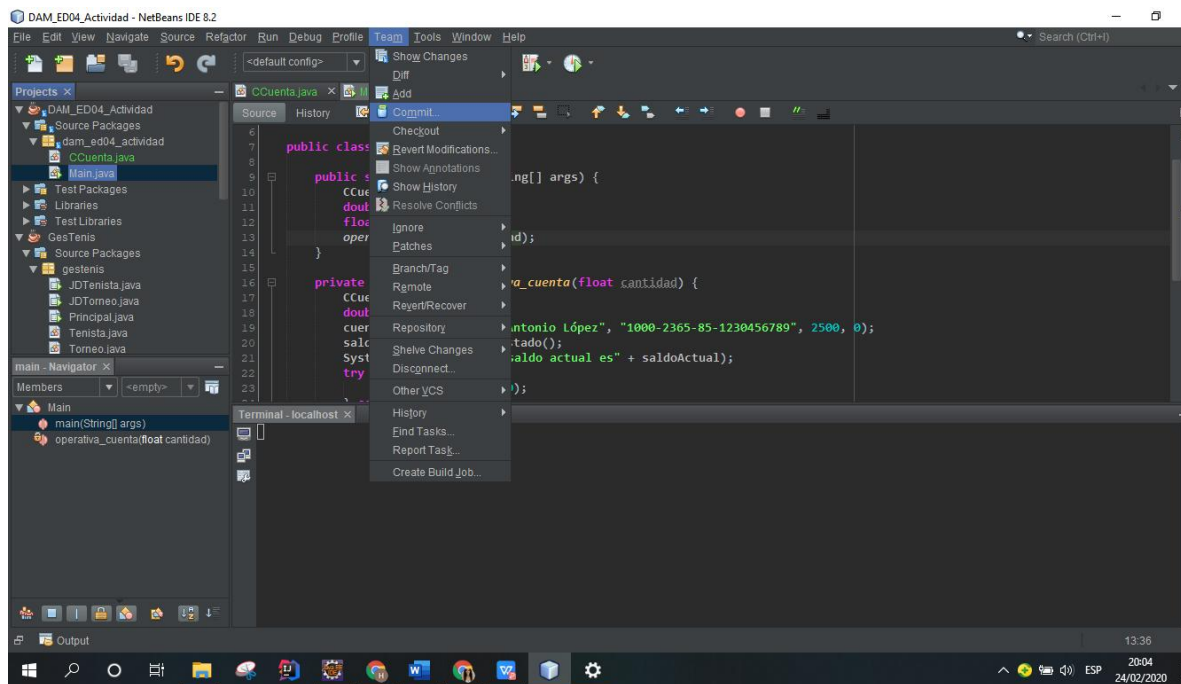
A continuación los nombres de las clases que se han añadido a git aparecerán en verde. En este caso la clase CCuenta y la clase Main

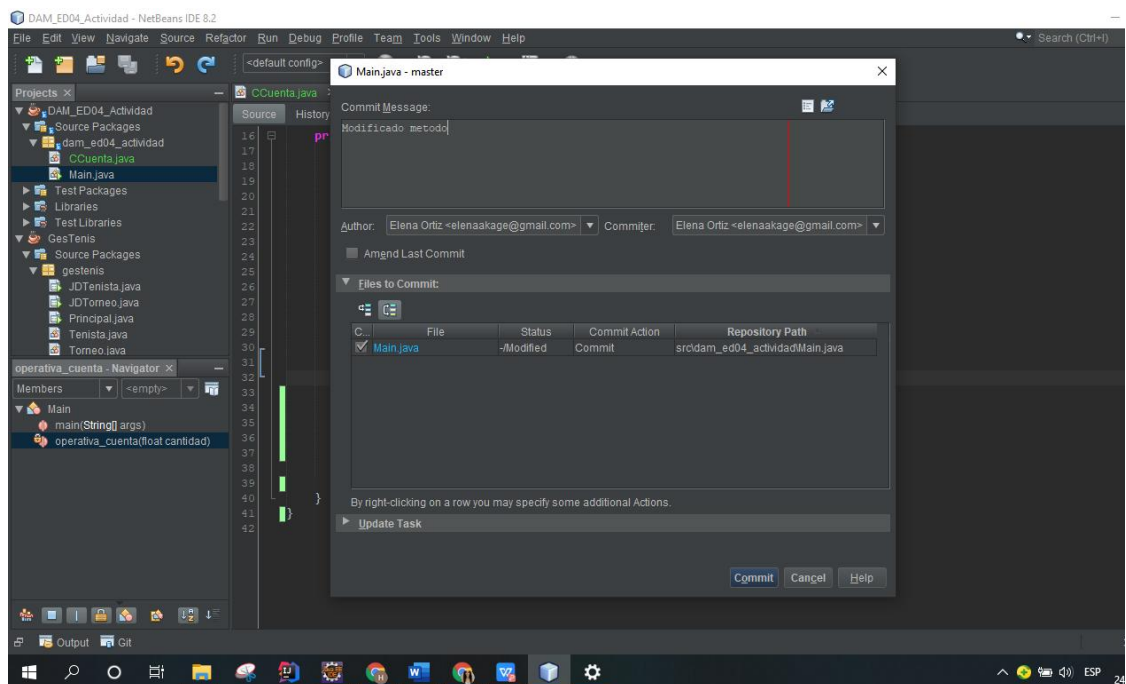
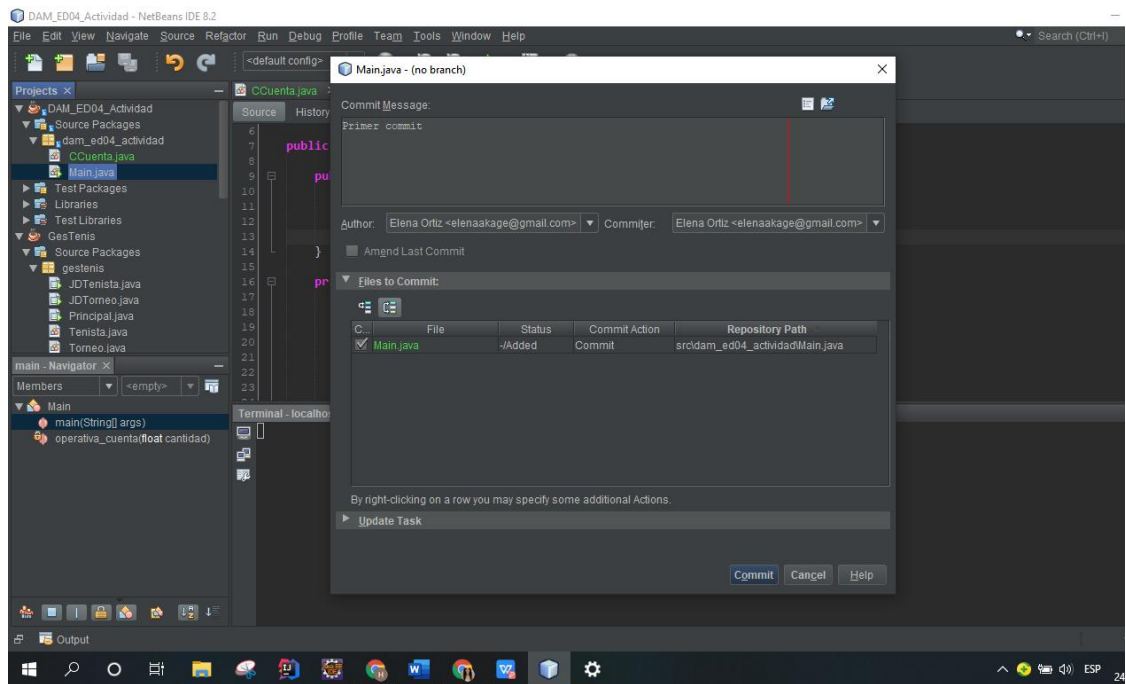




2. Realiza una operación commit.

Pulsamos Team > Commit... y escribimos el mensaje del commit y pulsamos el botón commit





Copia el código siguiente dentro del módulo operativa-cuenta (que se hizo en el punto 2 o en Main si no has hecho este punto) y guarda el proyecto.

```
try {

    System.out.println("Ingreso en cuenta");

    cuenta1.ingresar(1000);

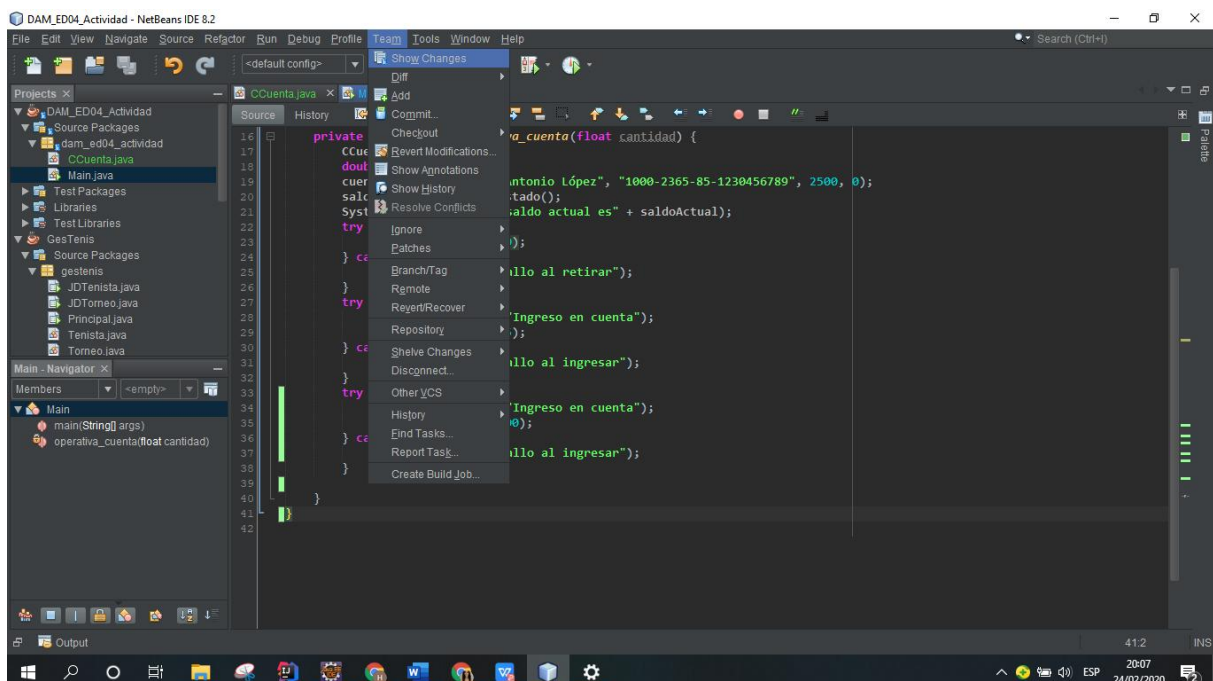
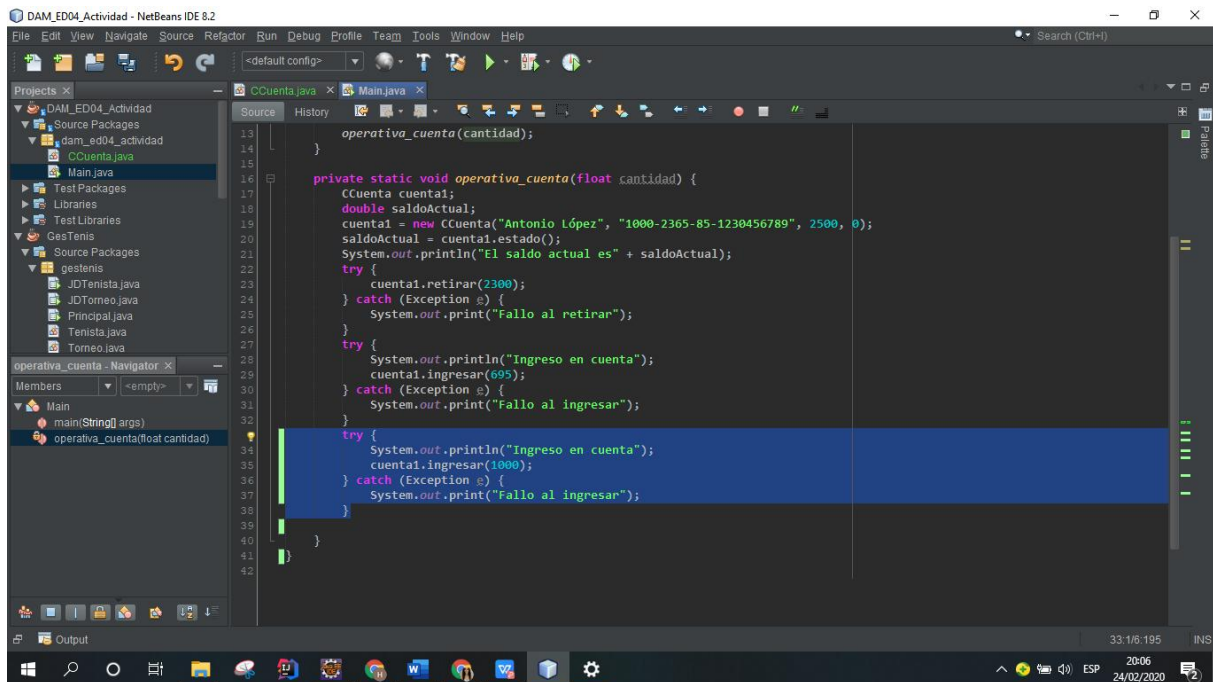
} catch (Exception e) {
```

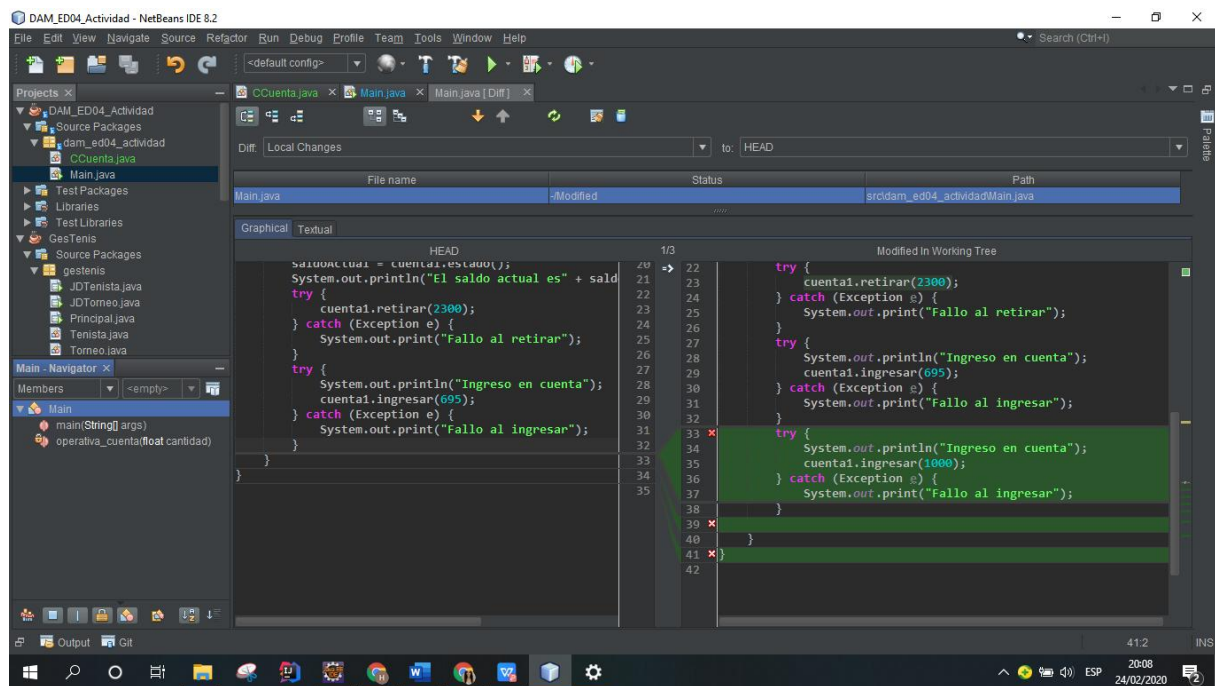
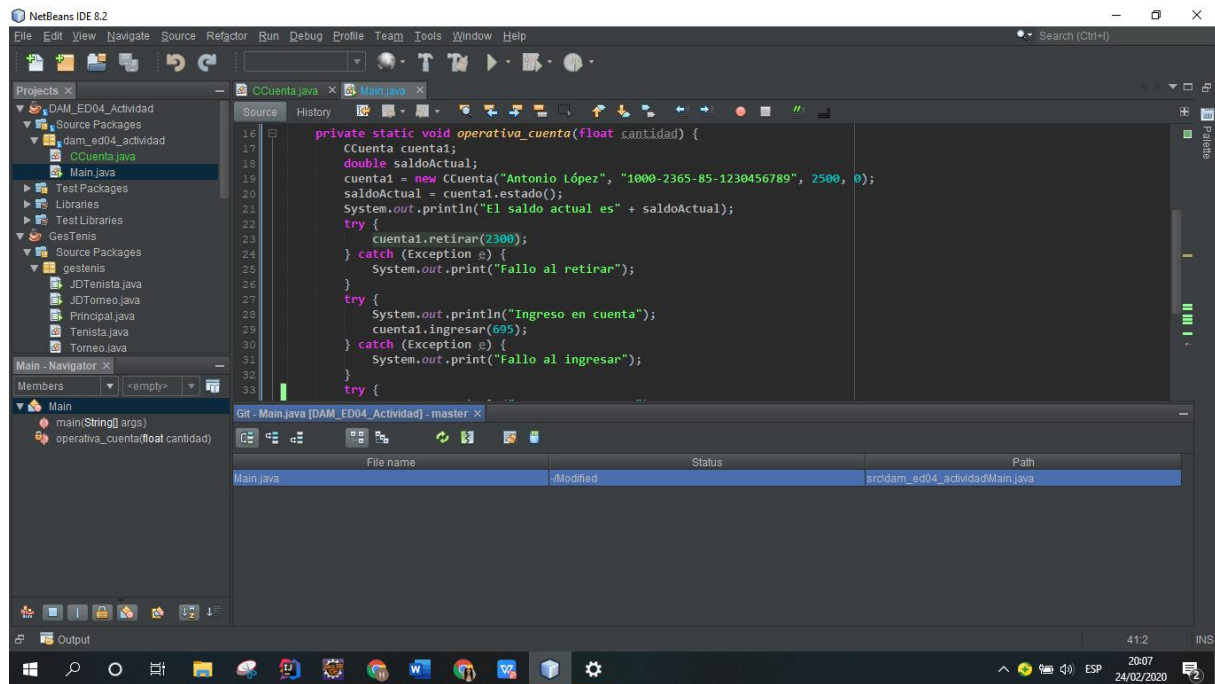
```
System.out.print("Fallo al ingresar");
```

```
}
```

1. Muestra con Git/Show-Changes los cambios producidos en el fichero Main.java y explora y comenta el resultado.

Pegamos el código y pulsamos Team > Show changes y nos mostrará los cambios que se han producido desde el último commit

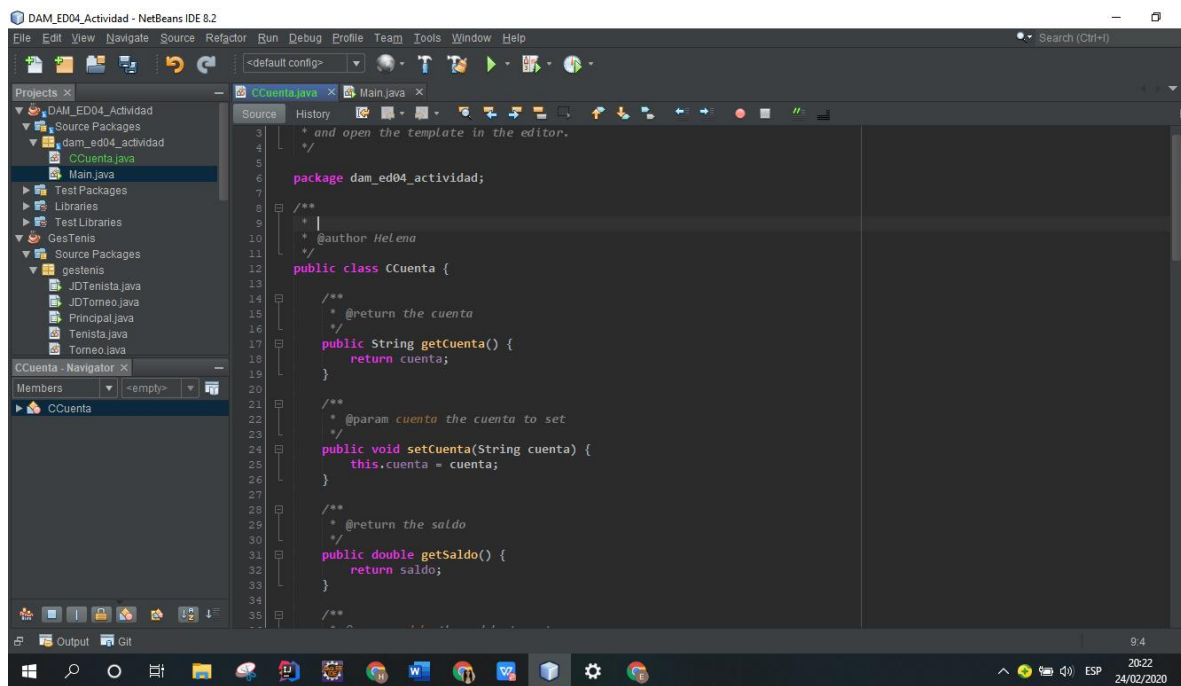
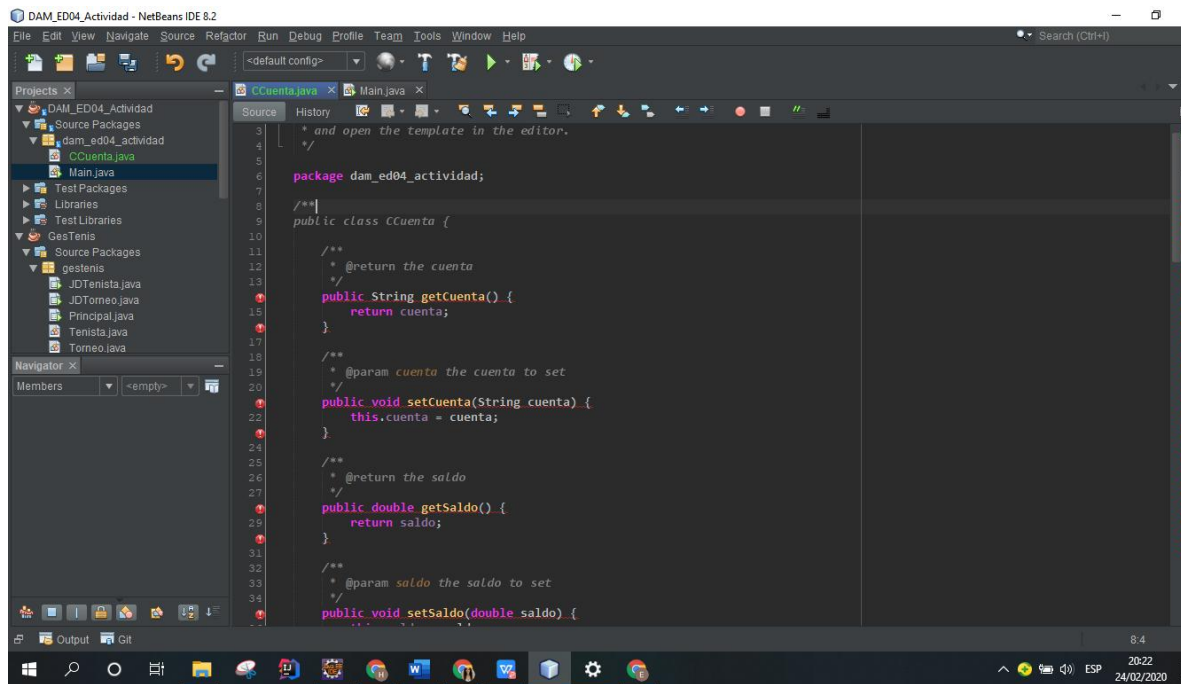


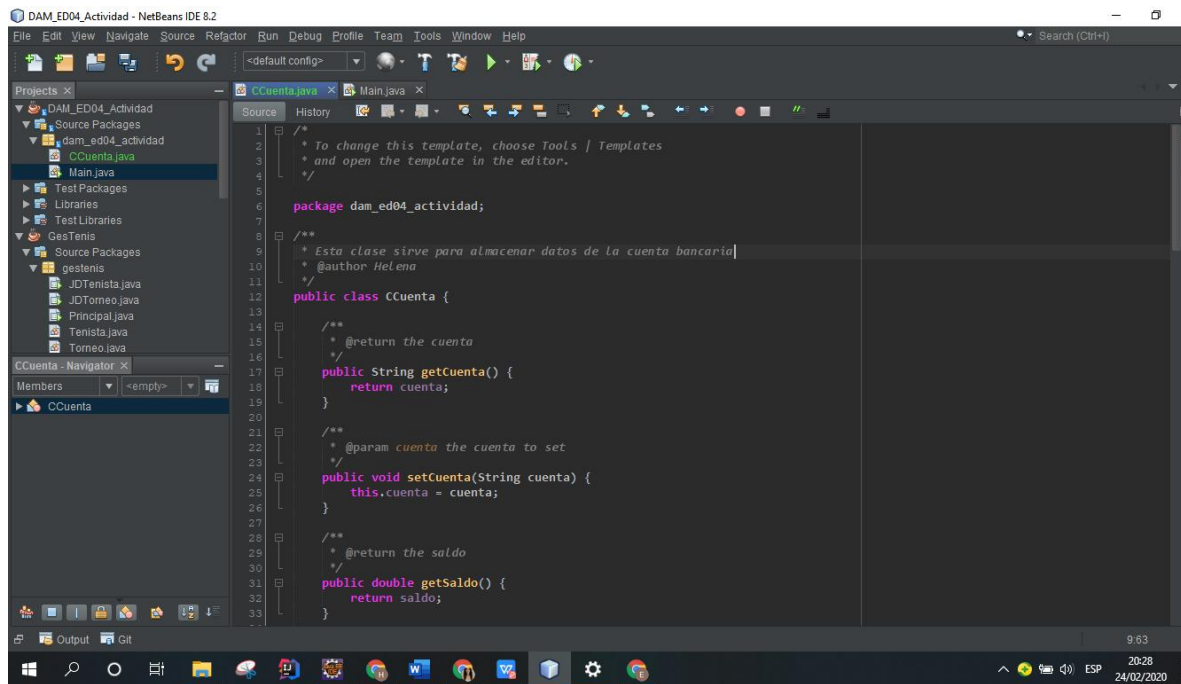


JAVADOC

1. Inserta comentarios Javadoc en la clase **CCuenta**.

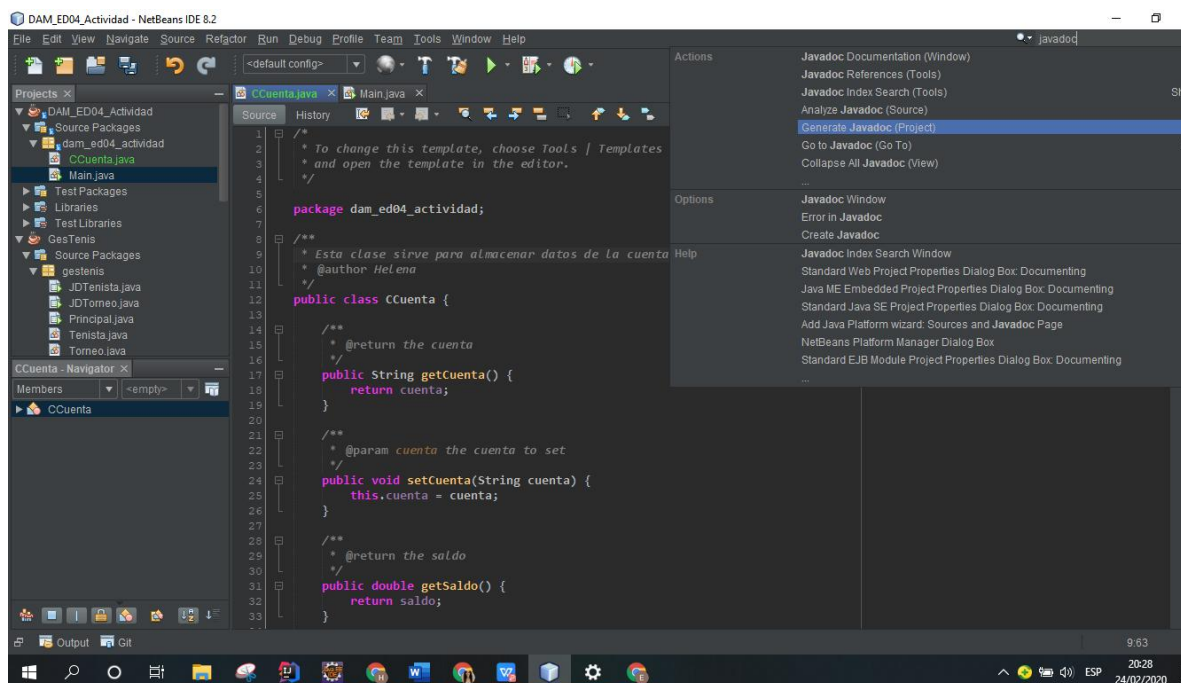
Para insertar Javadoc ponemos encima de la clase `/**` y pulsamos la tecla enter y automáticamente NetBeans genera el Javadoc

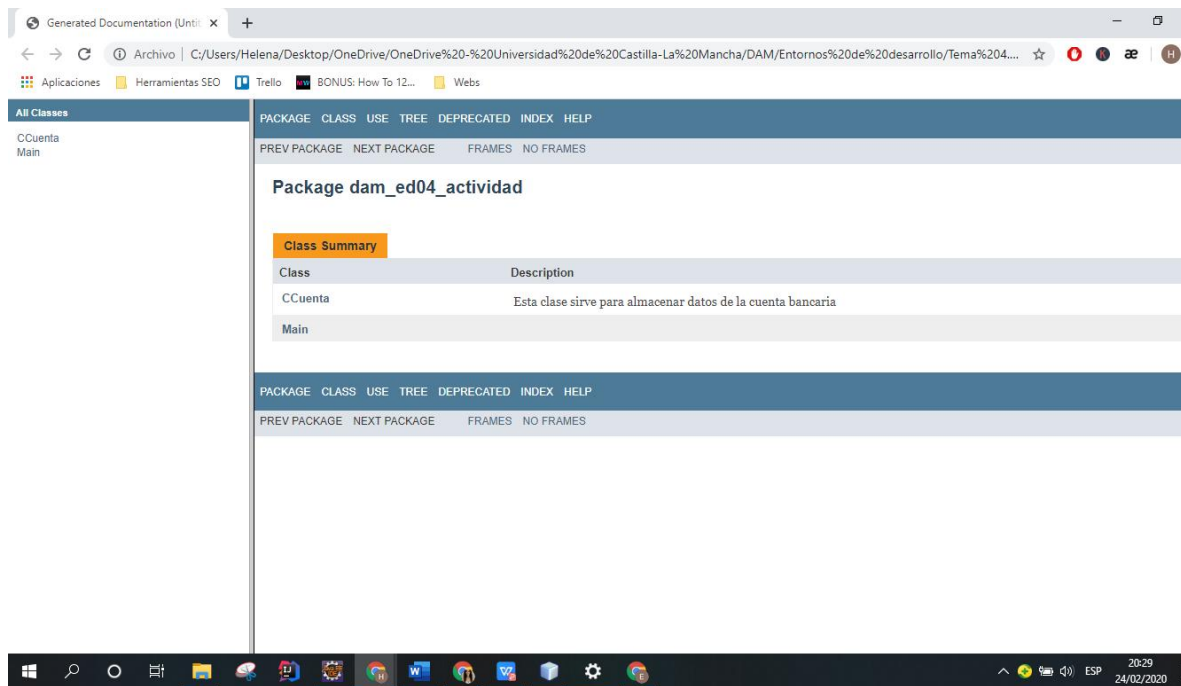




2. Genera documentación Javadoc para todo el proyecto.

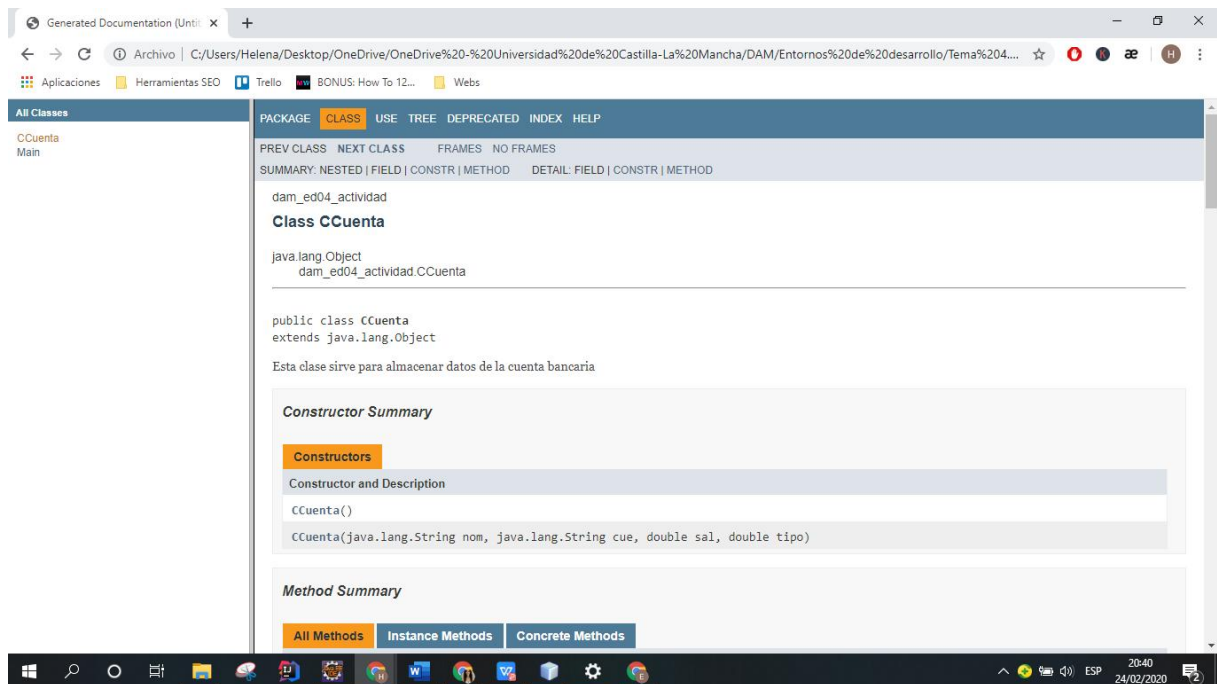
Buscamos en la barra java doc y pulsamos la opción Generate Javadoc (Project) y automáticamente nos genera un Javadoc de todo el proyecto





3. Comprueba que la documentación generada por Javadoc, abarca todos los métodos y atributos de la clase **CCuenta**.

A continuación se adjuntan capturas del Javadoc generado de la clase CCuenta para comprobar que se ha generado correctamente.



Generated Documentation (Untitled) x +

← → ↻ Archivo | C:/Users/Helena/Desktop/OneDrive/OneDrive%20-%20Universidad%20de%20Castilla-La%20Mancha/DAM/Entornos%20de%20desarrollo/Tema%204... ☆ 🔍 🌐 🌐 🌐

Aplicaciones Herramientas SEO Trello BONUS: How To 12... Webs

All Classes

- CCuenta
- Main

Method Summary

All Methods	Instance Methods	Concrete Methods
void	asignarNombre(java.lang.String nom)	
double	estado()	
java.lang.String	getCuenta()	
java.lang.String	getNombre()	
double	getSaldo()	
double	getTipoInterés()	
void	ingresar(double cantidad)	
java.lang.String	obtenerCuenta()	
java.lang.String	obtenerNombre()	
void	retirar(double cantidad)	
void	setCuenta(java.lang.String cuenta)	
void	setNombre(java.lang.String nombre)	
void	setSaldo(double saldo)	
void	setTipoInterés(double tipoInterés)	

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Generated Documentation (Untitled) x +

← → ↻ Archivo | C:/Users/Helena/Desktop/OneDrive/OneDrive%20-%20Universidad%20de%20Castilla-La%20Mancha/DAM/Entornos%20de%20desarrollo/Tema%204... ☆ 🔍 🌐 🌐 🌐

Aplicaciones Herramientas SEO Trello BONUS: How To 12... Webs

All Classes

- CCuenta
- Main

void setSaldo(double saldo)

void setTipoInterés(double tipoInterés)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

CCuenta

public CCuenta()

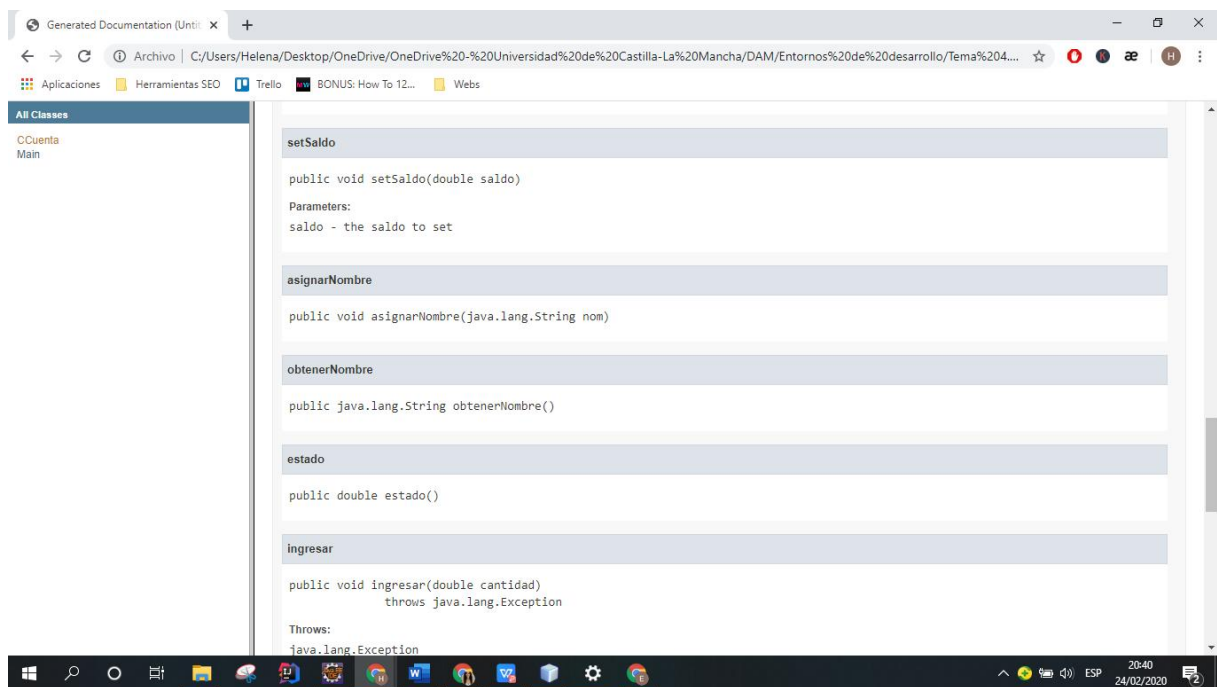
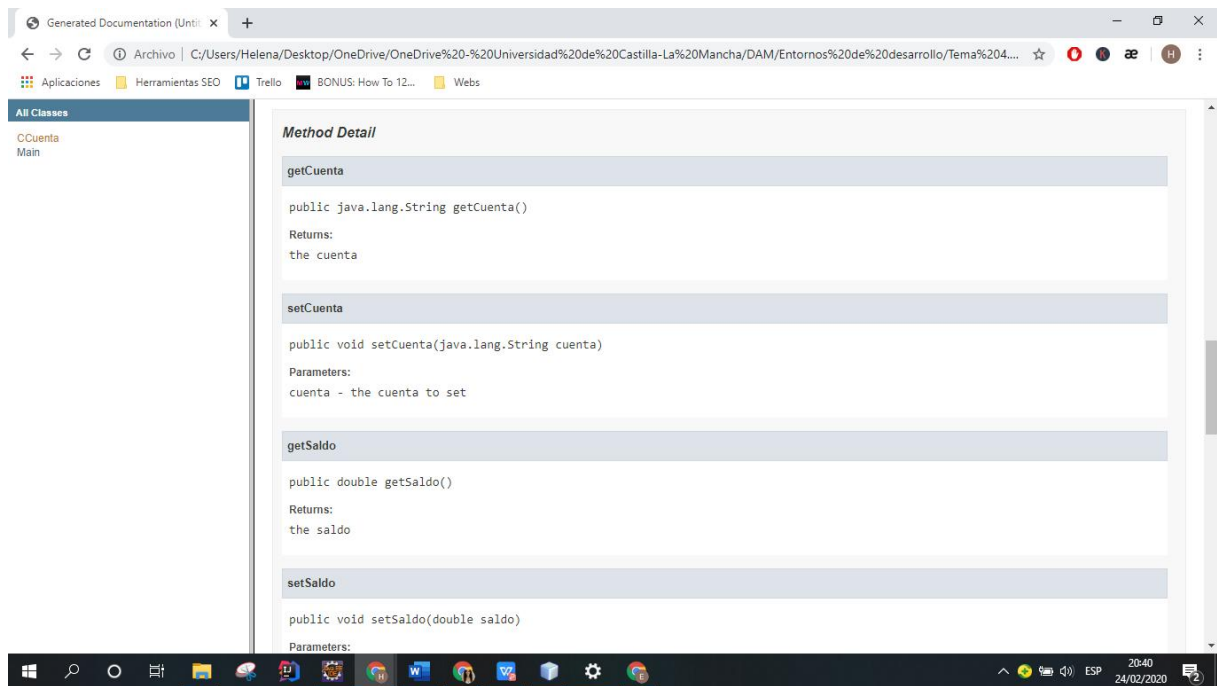
CCuenta

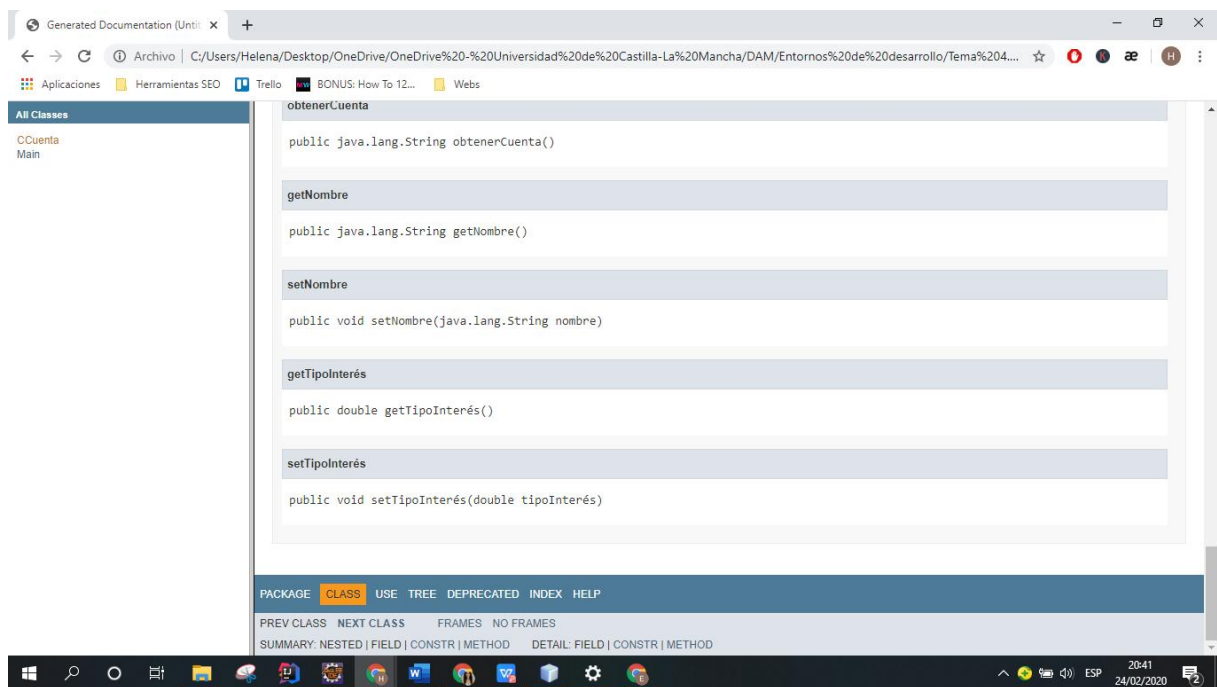
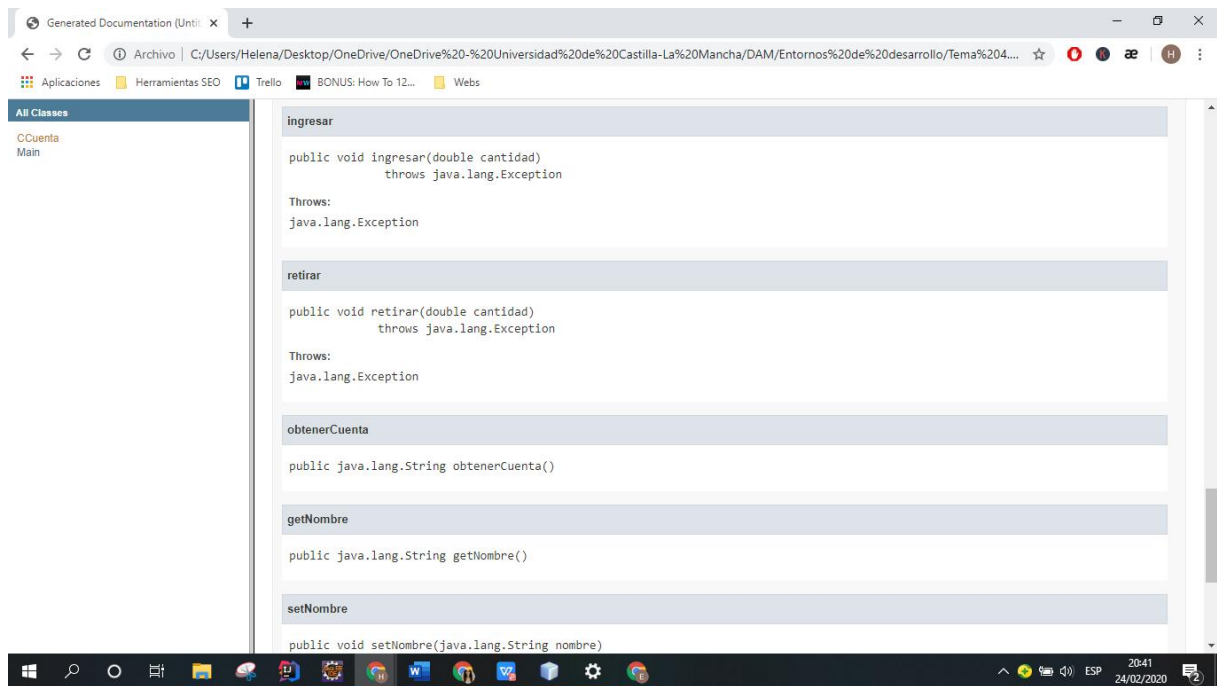
public CCuenta(java.lang.String nom,
java.lang.String cue,
double sal,
double tipo)

Method Detail

getCuenta

file:///C:/Users/Helena/Desktop/OneDrive - Universidad de Castilla-La Mancha/DAM/Entornos de desarrollo/Tema 4. Optimización y documentación/DAM_ED04_Actividad/dist/javadoc/dam_ed04_actividad/CCuenta.html#obtenerCuenta...





LINK AL CÓDIGO

<https://bit.ly/EntornosDesarrollo04>