Enunciado.

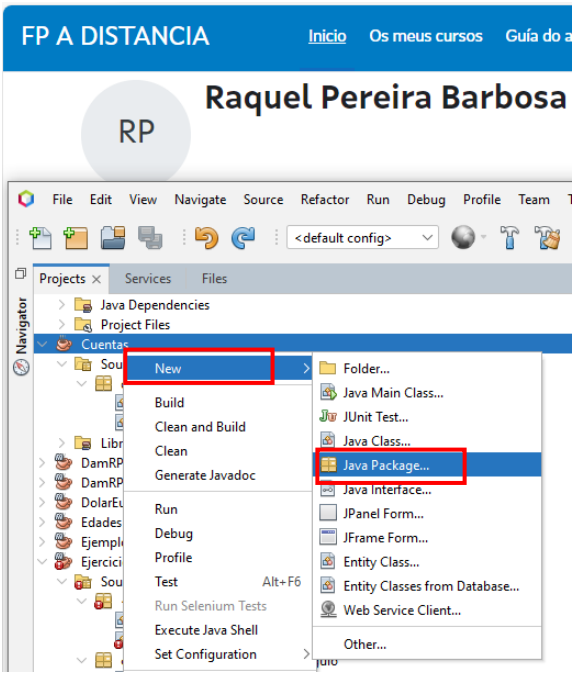
En el proyecto Java "Deposito", hay definida una Clase llamada CCuenta, que tiene una serie de atributos y métodos. El proyecto cuenta asimismo con una Clase Main, donde se hace uso de la clase descrita.

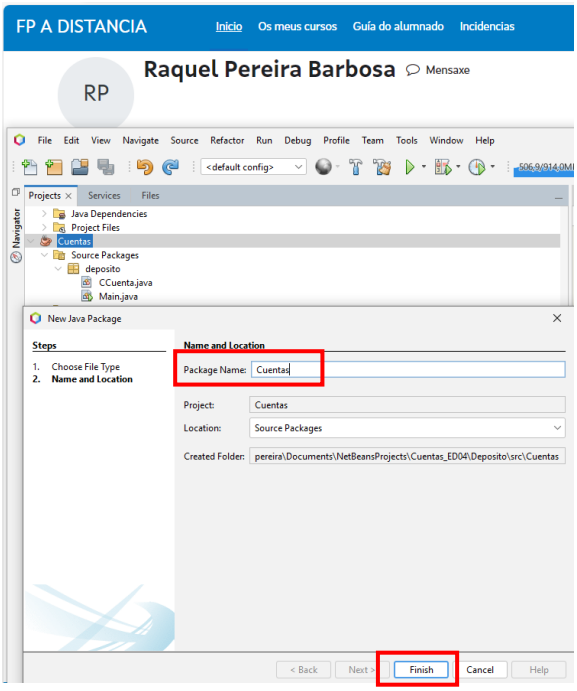
Basándonos en ese proyecto, vamos a realizar las siguientes actividades.

REFACTORIZACIÓN

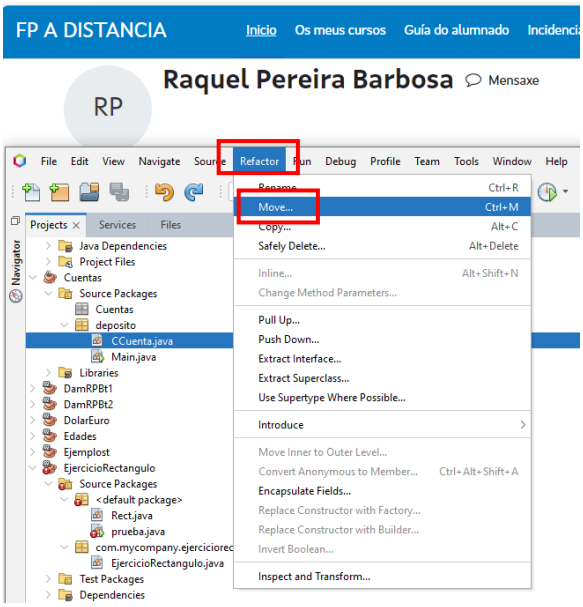
1. Las clases deberán formar parte del paquete cuentas.

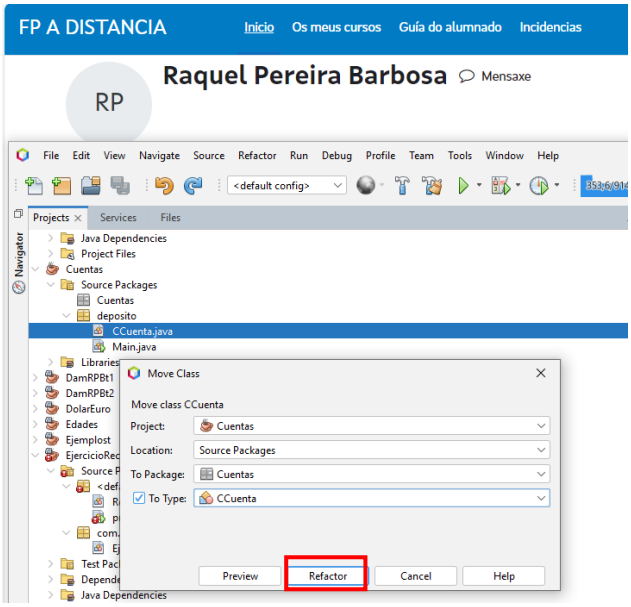
Crearemos un nuevo paquete llamado cuentas pulsando sobre el proyecto botón derecho-Nuevo Paquete y lo llamaremos Cuentas

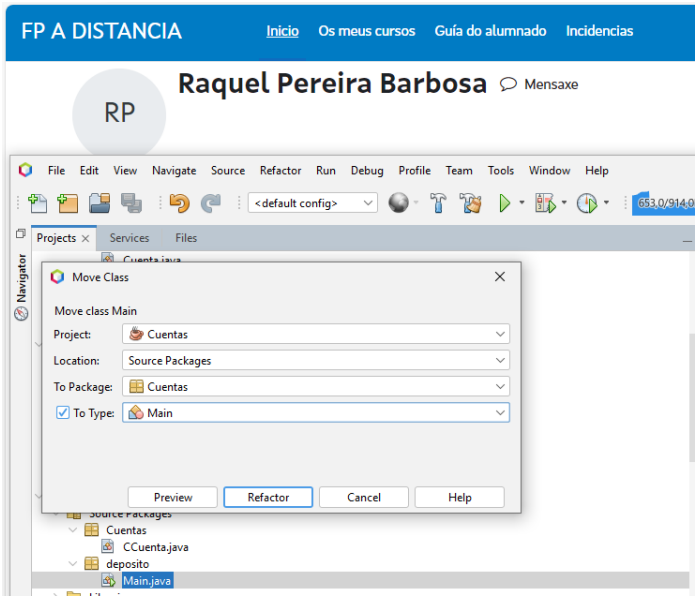




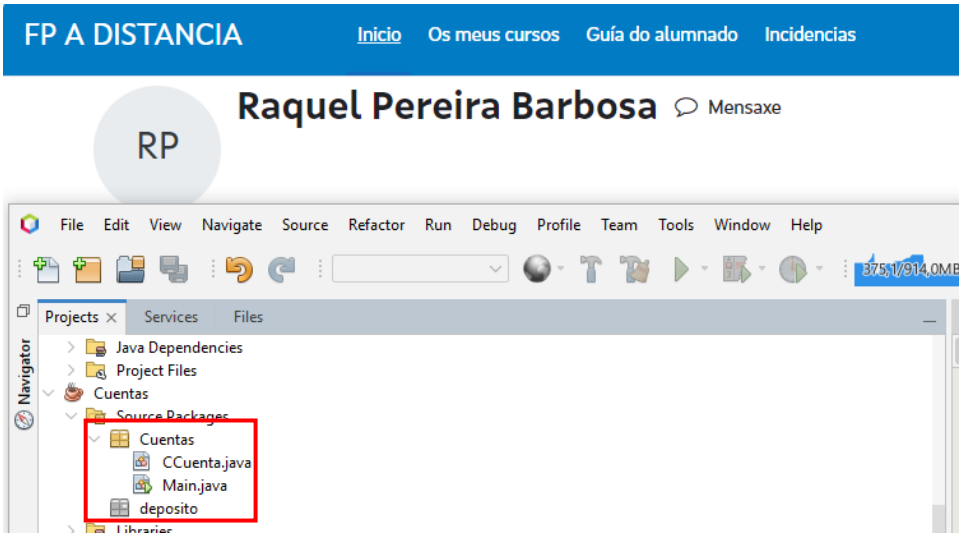
Con refactorización movemos las clases existentes al paquete Cuentas y lo repetimos para ambas clases



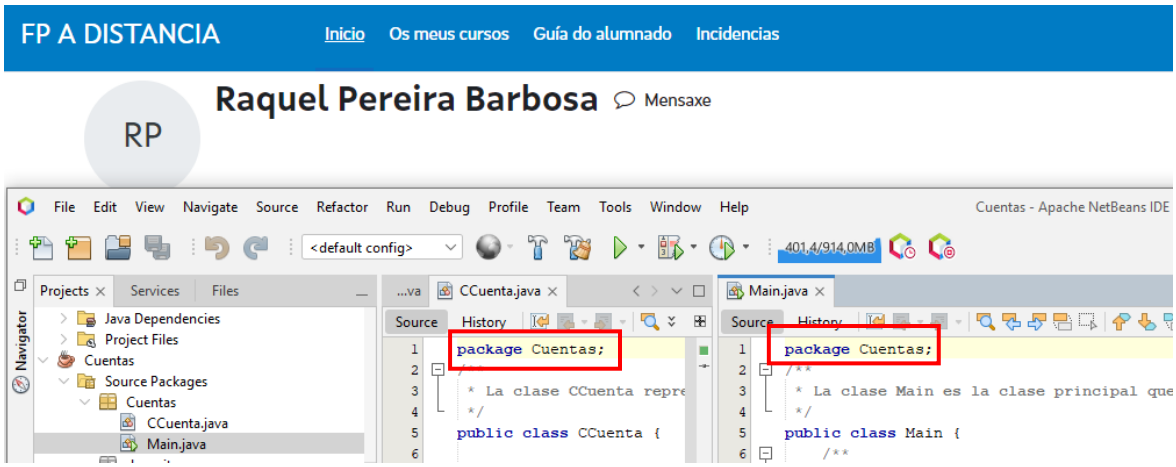




Vemos que se han movido al paquete cuentas y que depósito está vacío

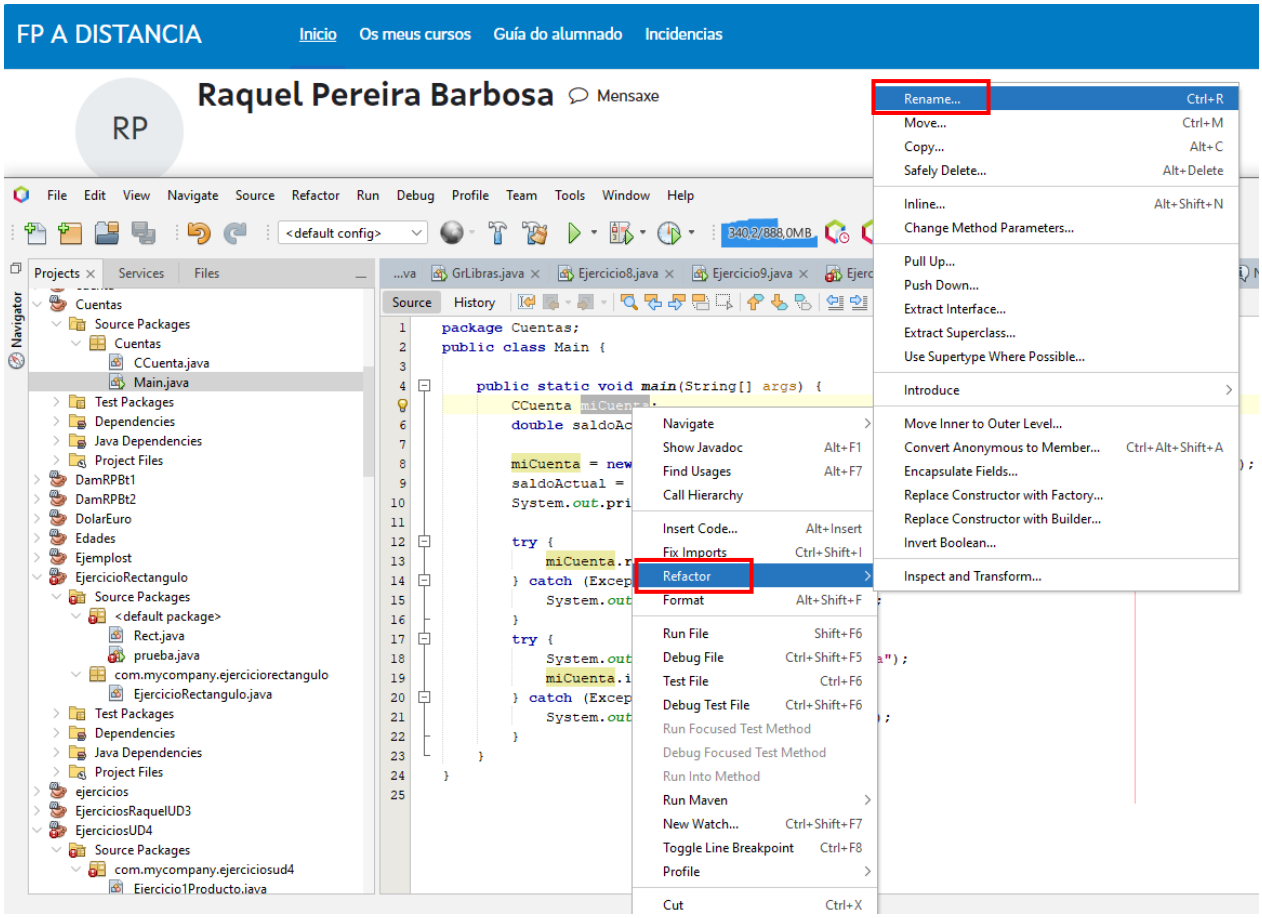


También podemos observar que al refactorizar el paquete ya se cambia automático en el código:

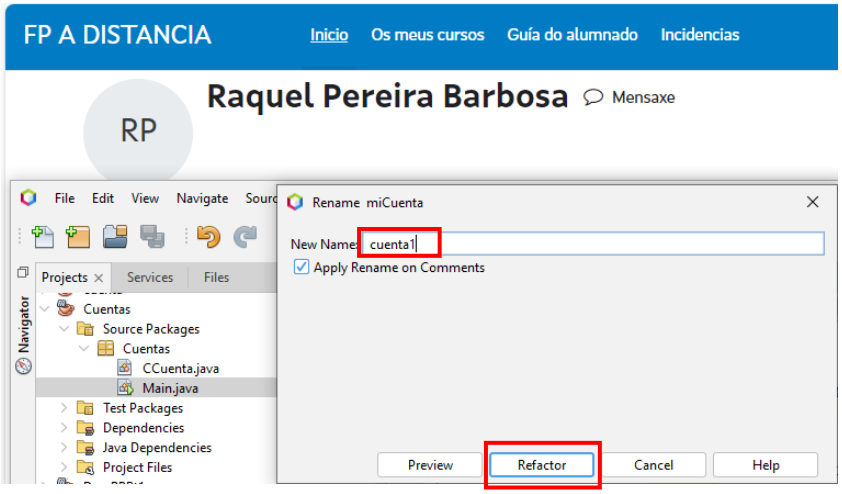


2. Cambiar el nombre de la variable "miCuenta" por "cuenta1".

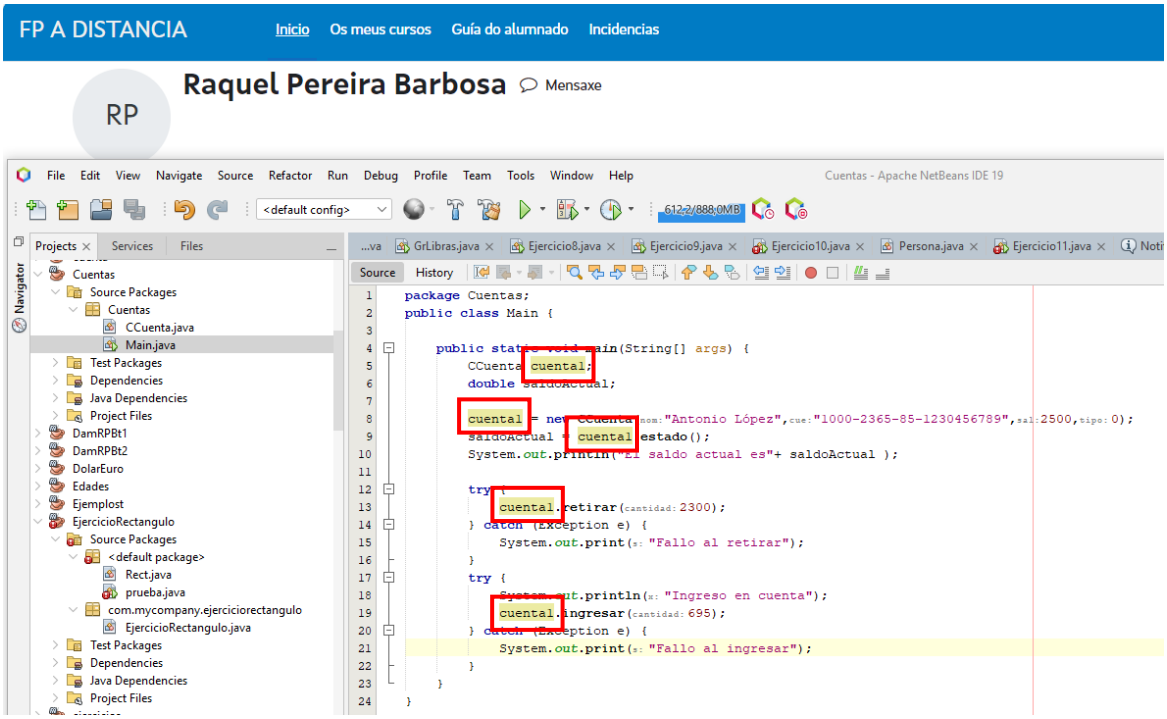
Para hacer este cambio marcaremos el primer miCuenta y pulsaremos botón derecho, ahí iremos a Refactorizar - Renombrar:



A continuación, ponemos el nuevo nombre “cuenta1” y le damos a refactorizar:

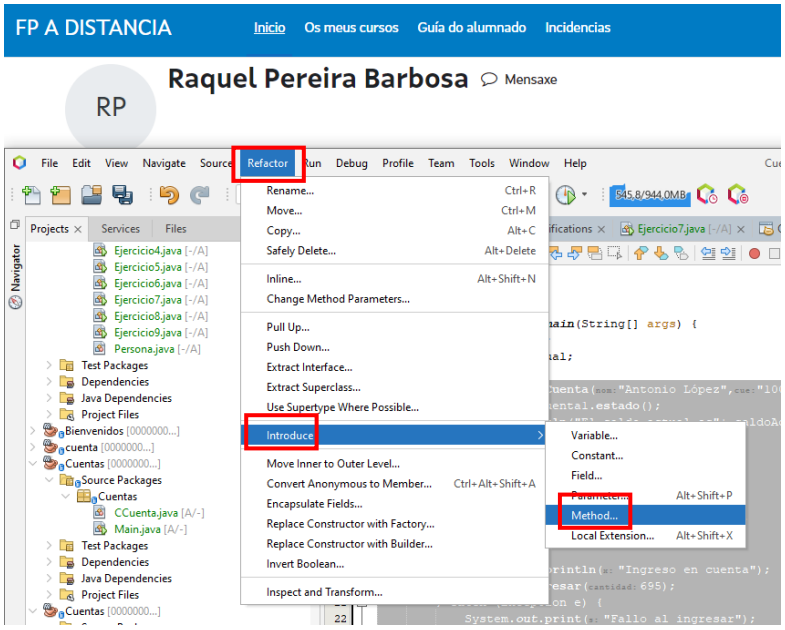


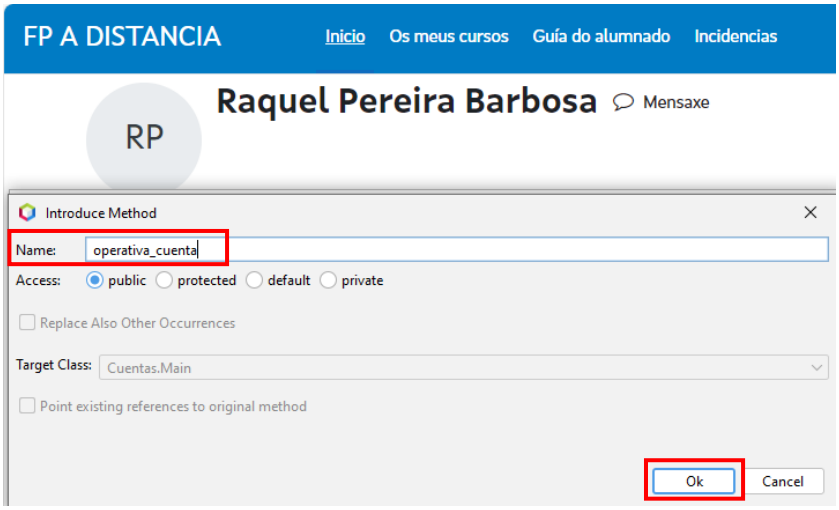
Y con esto nos cambia todos los nombres de la variable a la vez



3. Introducir el método operativa_cuenta, que englobe las sentencias de la clase Main que operan con el objeto cuenta1.

Para esta parte seleccionaremos la parte del código donde queremos introducir el método y le damos a refactorizar/introducir/método y le ponemos el nombre

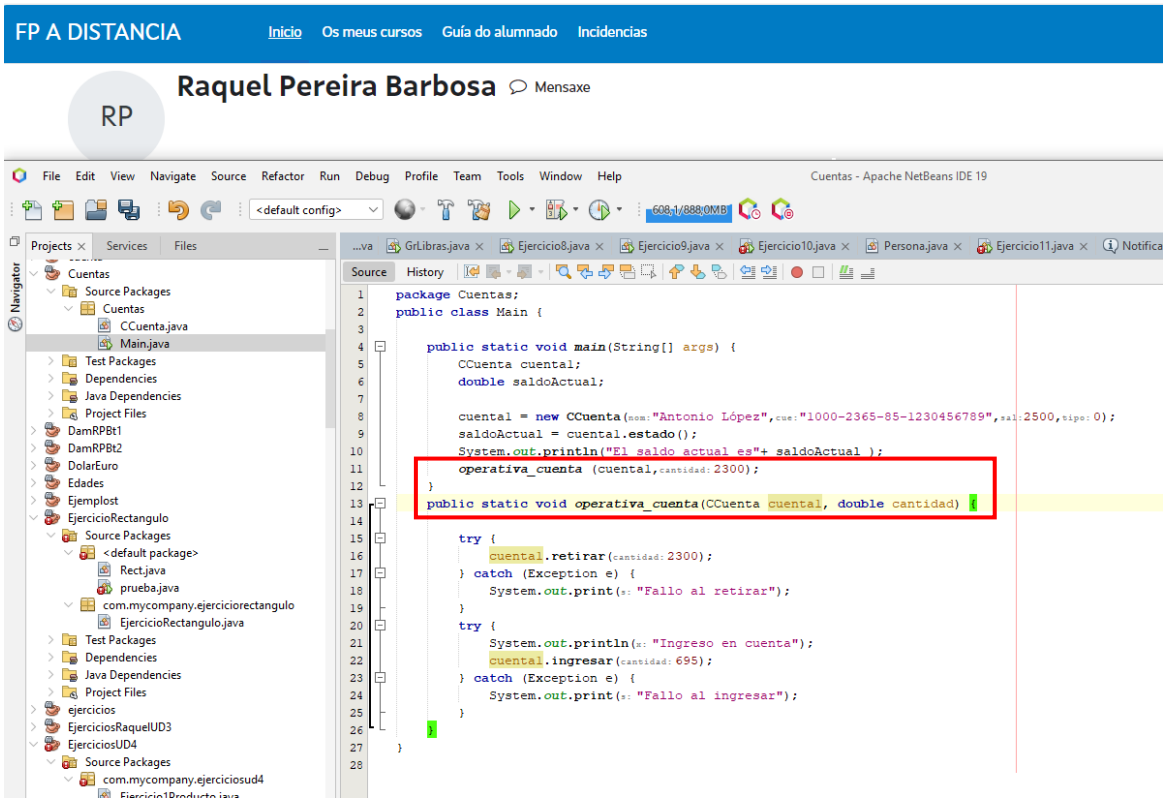




Nos ha creado lo solicitado

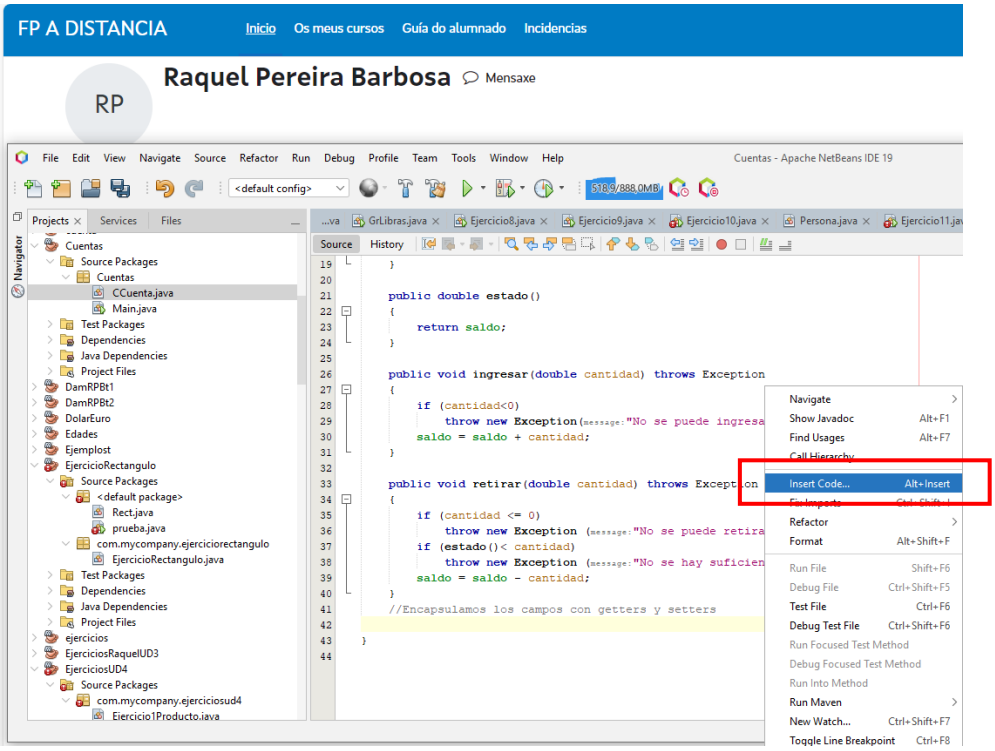
Igualmente, también en este caso que no es mucho código podríamos hacerlo a mano para ajustarlo a nuestro gusto. Para esto en este código, se ha introducido el método operativa_cuenta que recibe como parámetros una instancia de CCuenta llamada cuenta y un valor cantidad de tipo double.

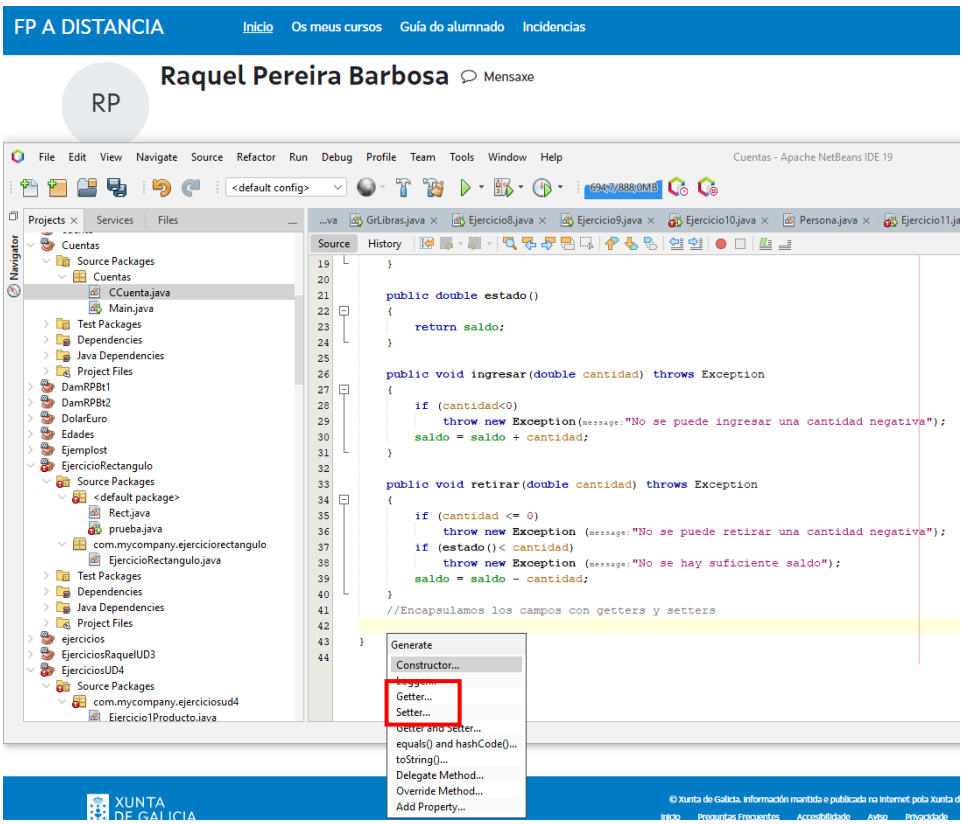
Luego, en el método main, se invoca operativa_cuenta pasando la instancia cuenta1 y la cantidad 2300.



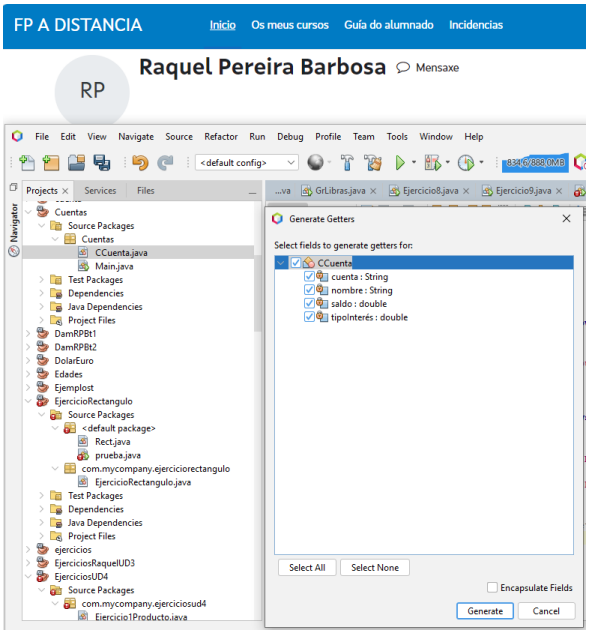
4. Encapsular los atributos de la clase CCuenta.

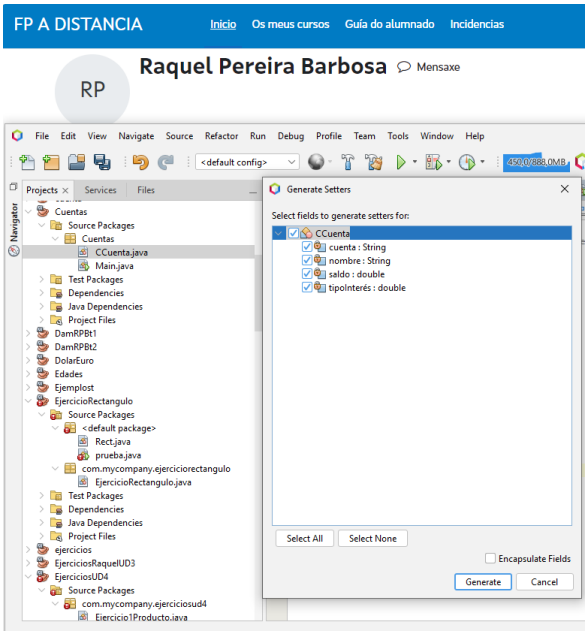
Para esto crearemos los métodos de asignación y de consulta getters y setters para los campos de la clase.
Hay dos formas de realizarlo:
1.-Lo haremos de forma automática, nos posicionaremos donde queremos incluirlo, botón derecho, añadir código:



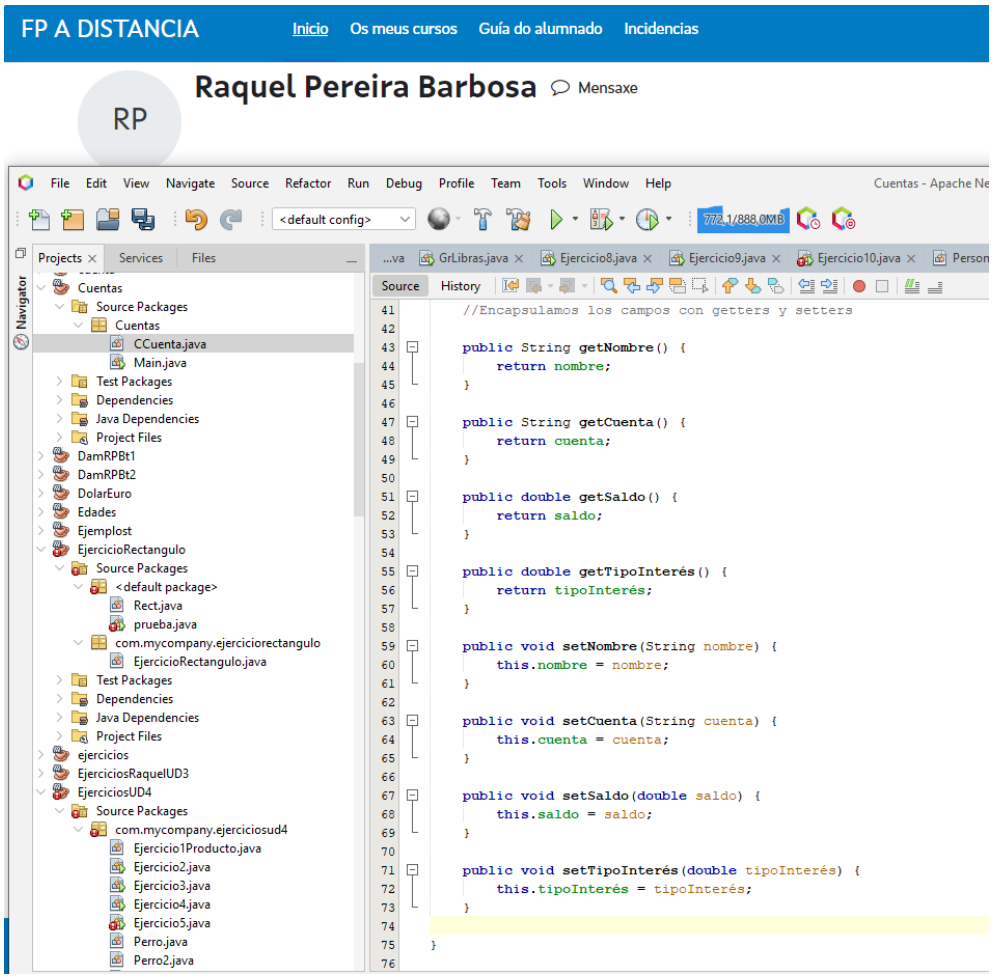


Seleccionamos getter y setter y lo aplicamos a toda CCuenta y le damos a generar

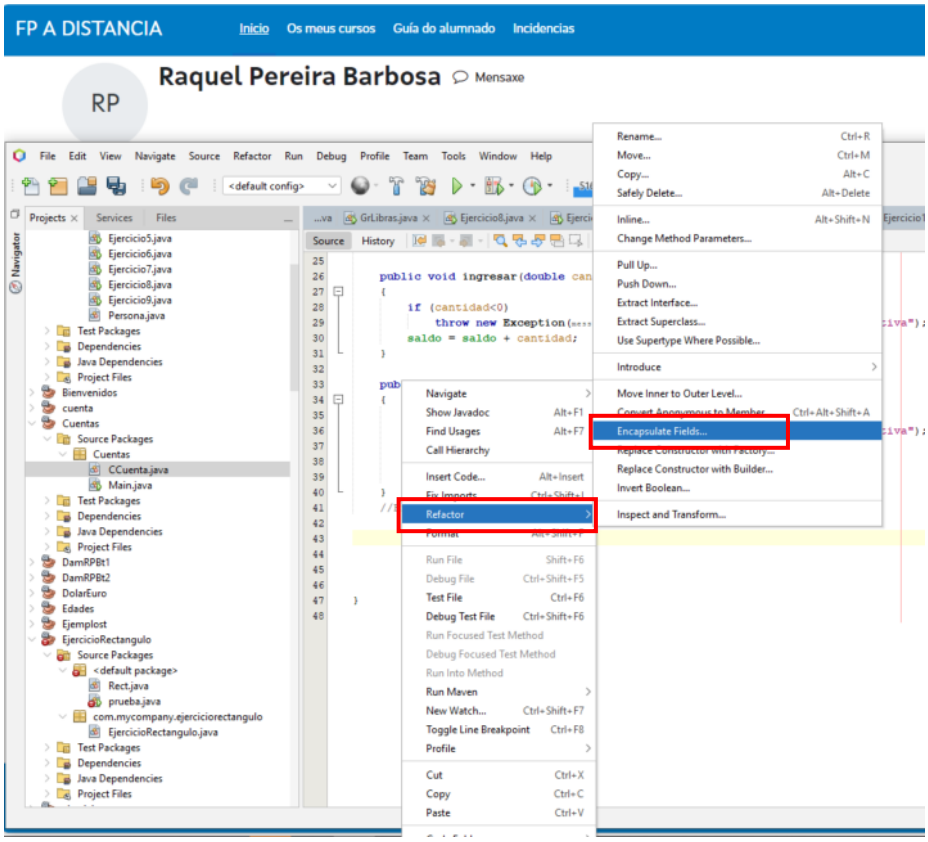




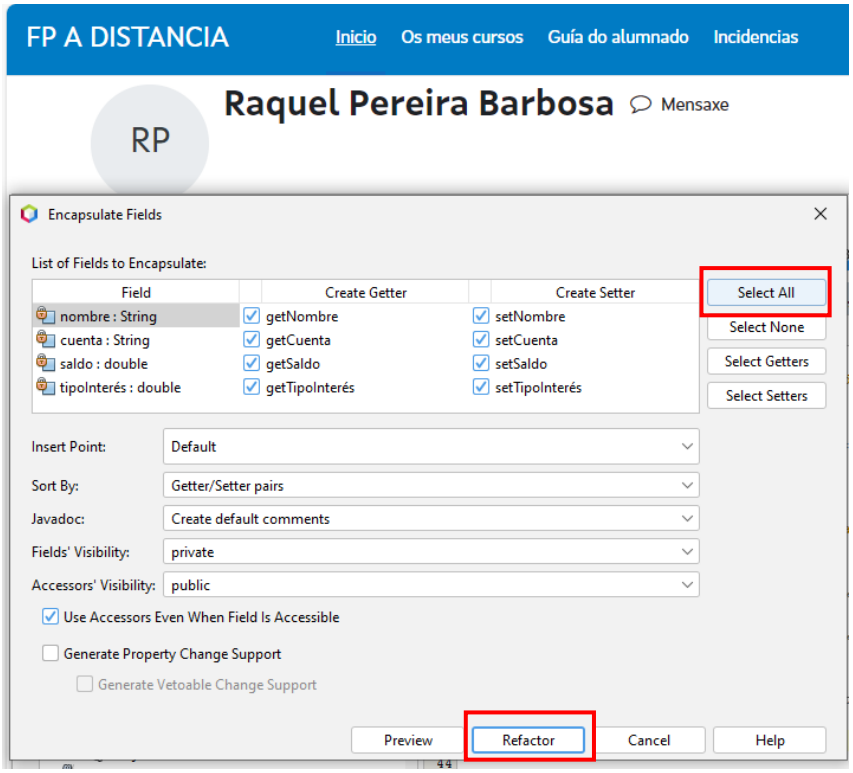
Esto generará todo el código automáticamente:



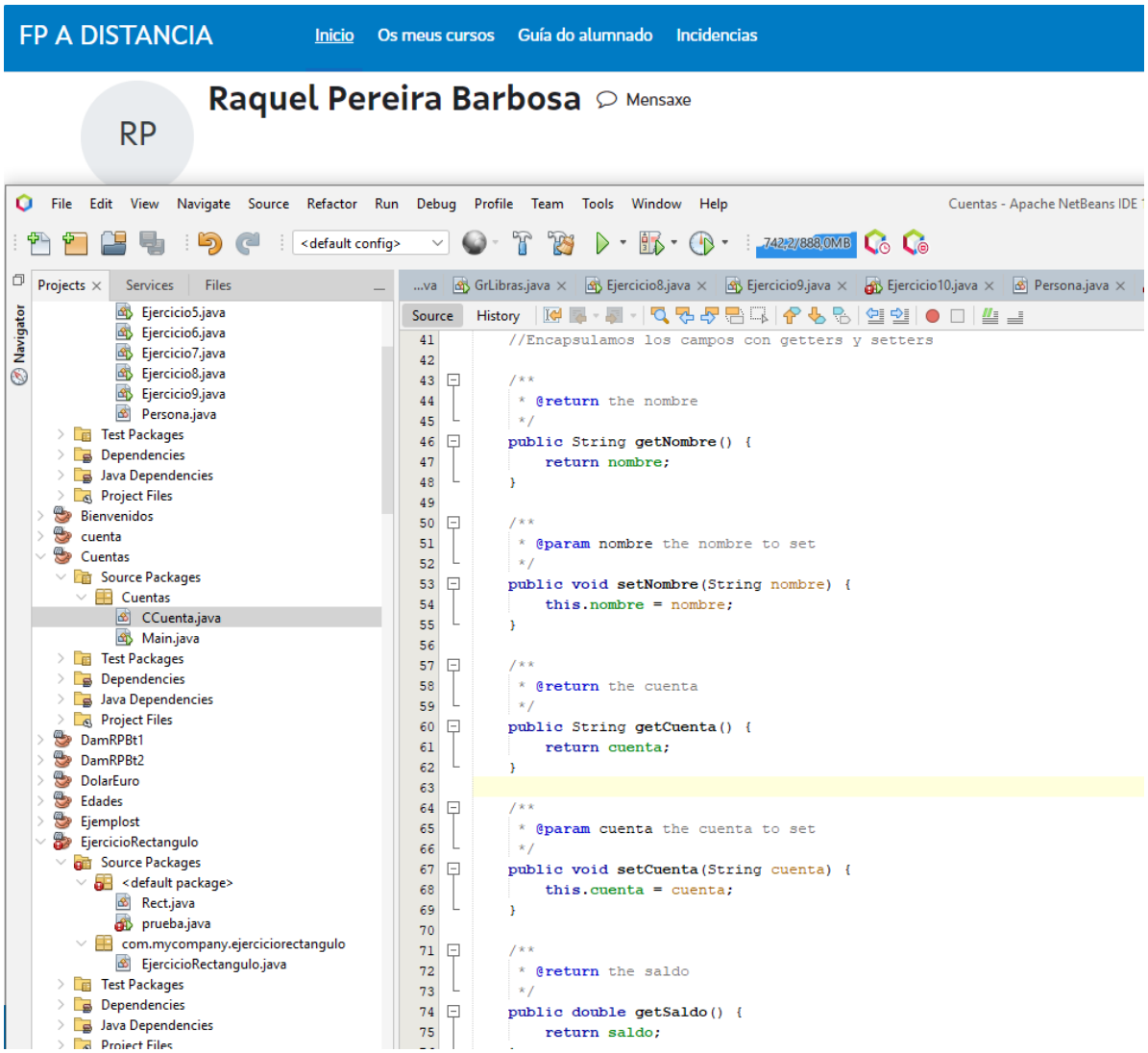
2.-Otra manera de hacerlo es a través de la opción de refactorización, nos posicionamos donde queremos ubicarlo y botón derecho refactorizar, encapsular campos:



Seleccionamos todo y le damos a refactorizar:



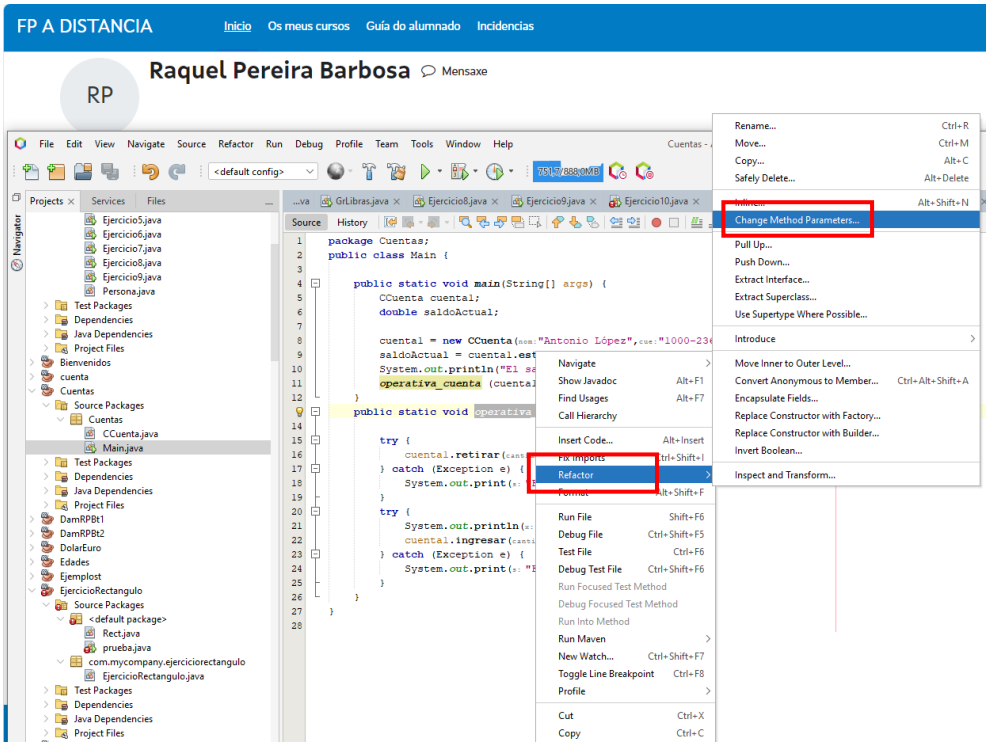
De esta manera nos insertará comentarios para poder entender el código:



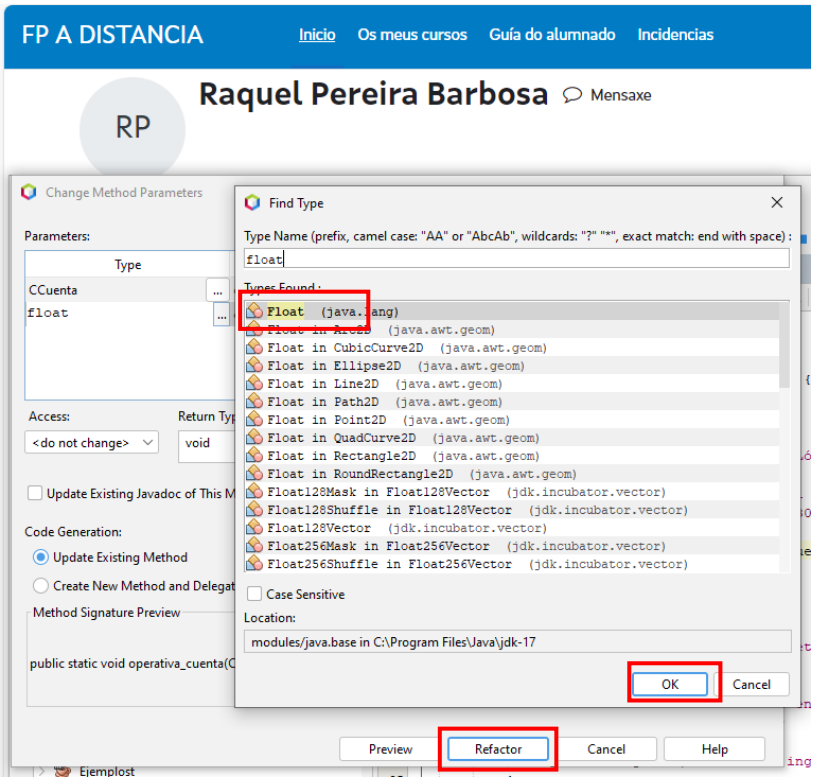
(Esto es una muestra ya que el código es más largo y no entra en un solo pantallazo)

5. Añadir un nuevo parámetro al método operativa_cuenta, de nombre cantidad y de tipo float.

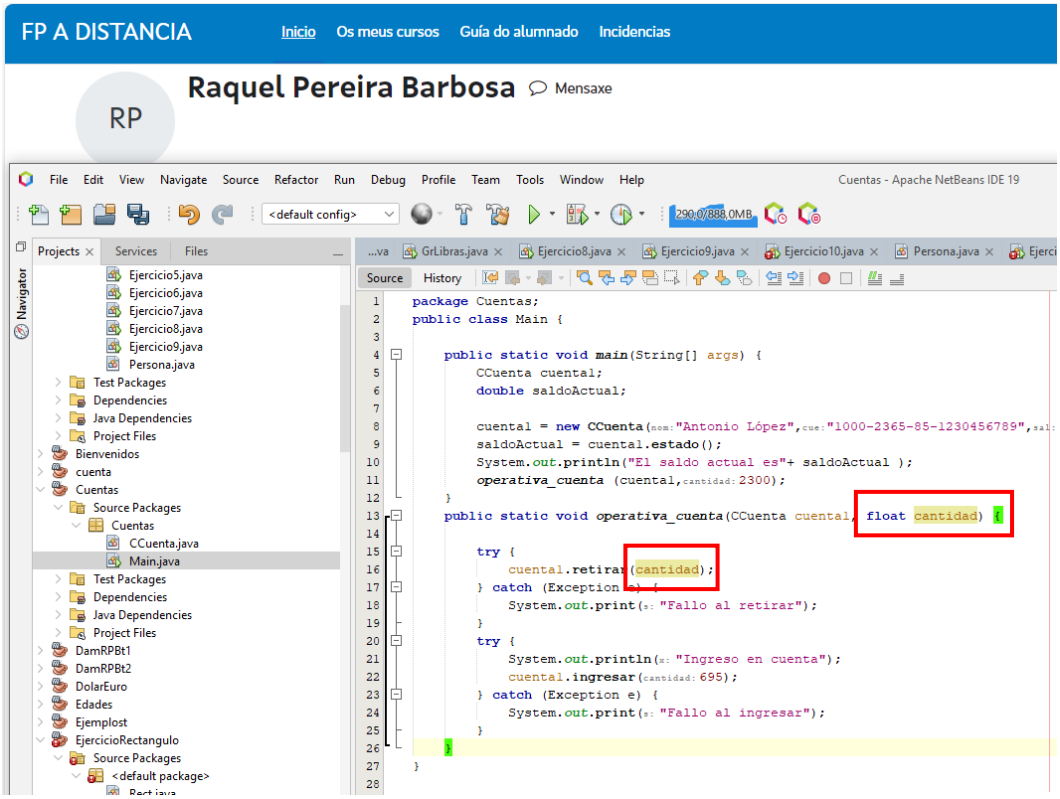
Para hacer este cambio en vez de hacerlo directamente sobre el código también lo haremos a través de la opción de refactorizar. Seleccionamos en nombre del método “operativa_cuenta”, botón derecho refactorizar, cambiar los parámetros del método:



Cambiamos el double por float y refactorizamos



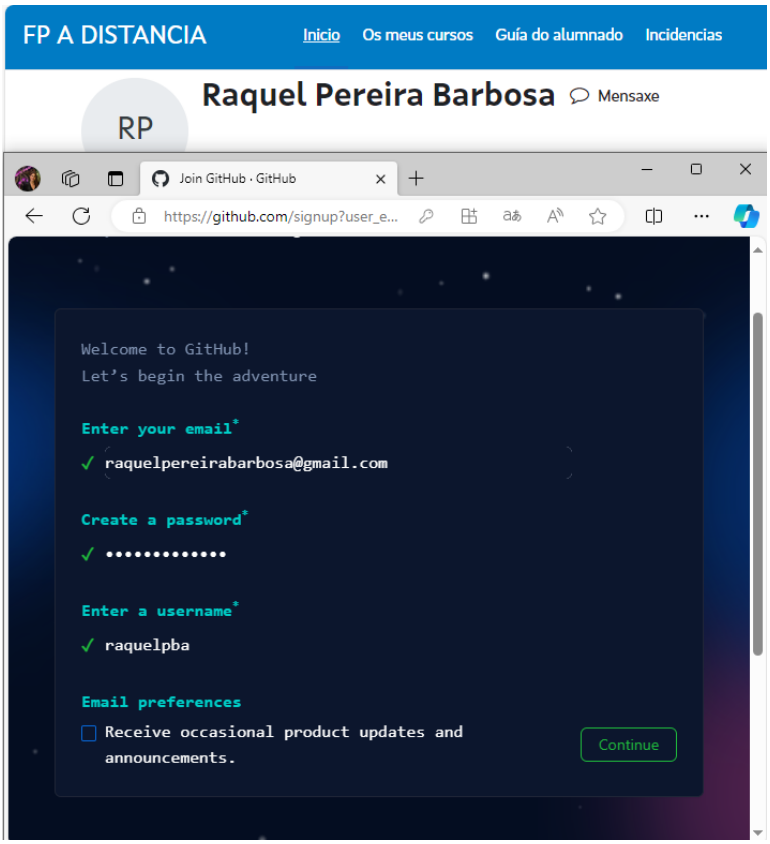
Ahora actualizamos la referencia al parámetro



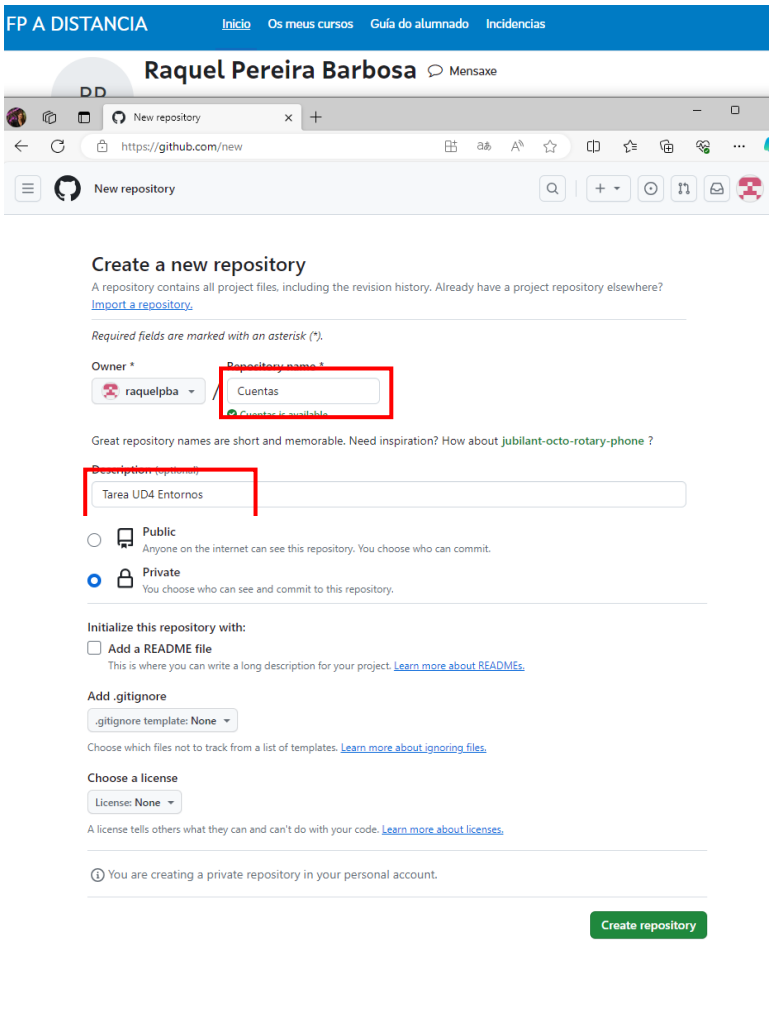
GIT

1. Configurar GIT para el proyecto. Crear un repositorio público en GitHub.

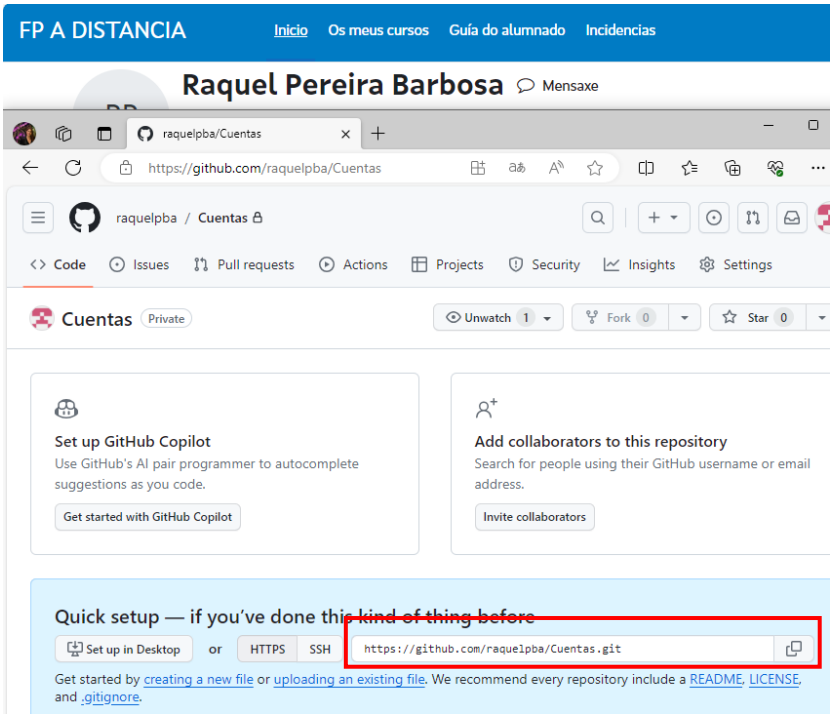
Para crear un repositorio en GitHub, primero daremos de alta una cuenta:



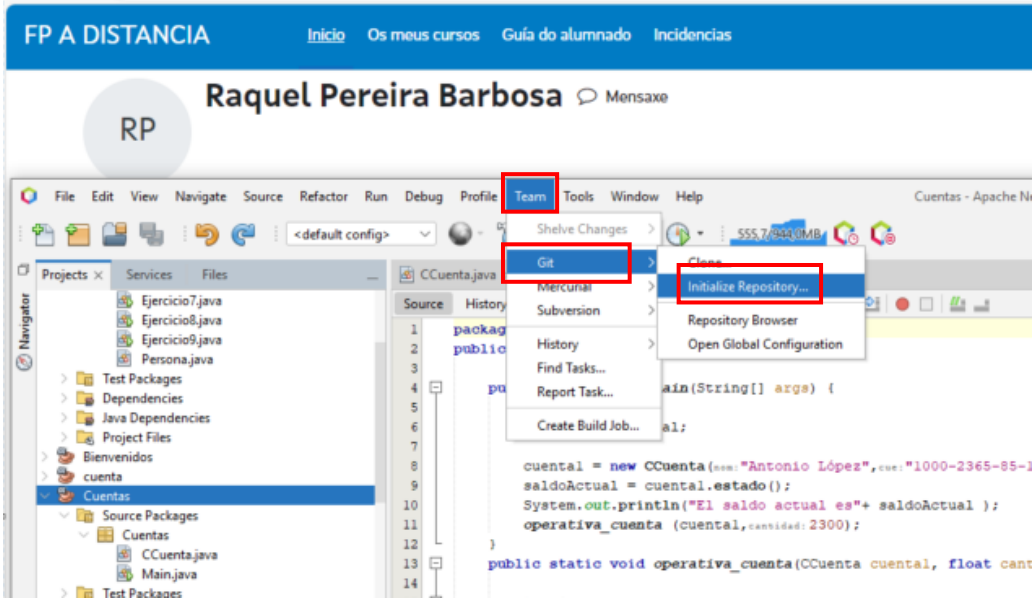
Una vez que ya tenemos una cuenta creamos un repositorio que llamaremos cuentas como nuestro proyecto



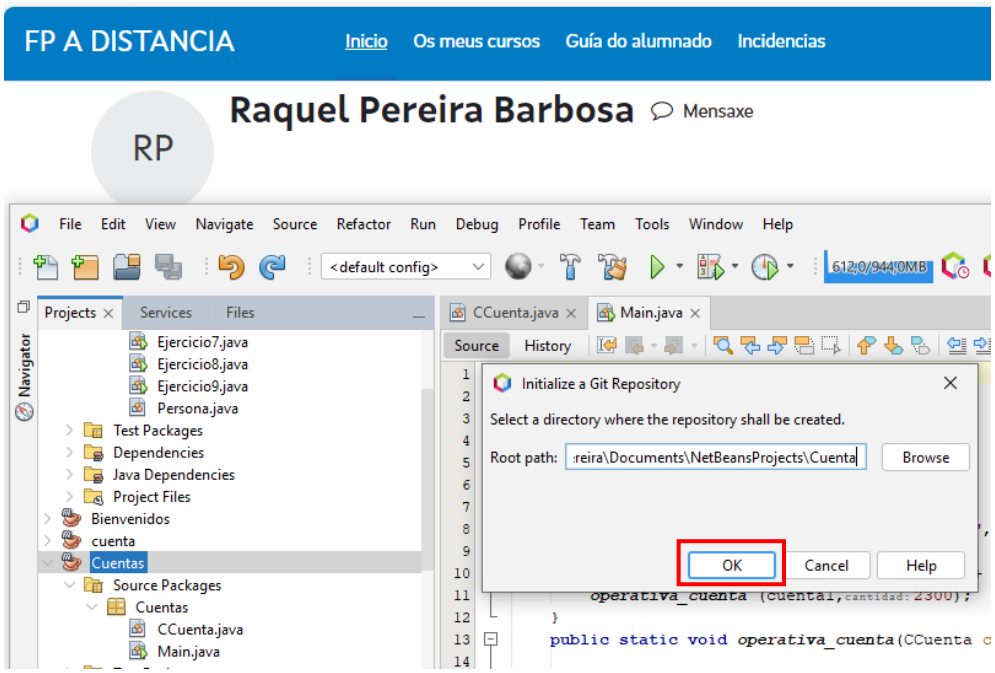
Una vez creado copiamos la URL generada https://github.com/raquelpba/Cuentas.git (es privada, me hace falta el usuario de github del profesor para que tenga acceso)



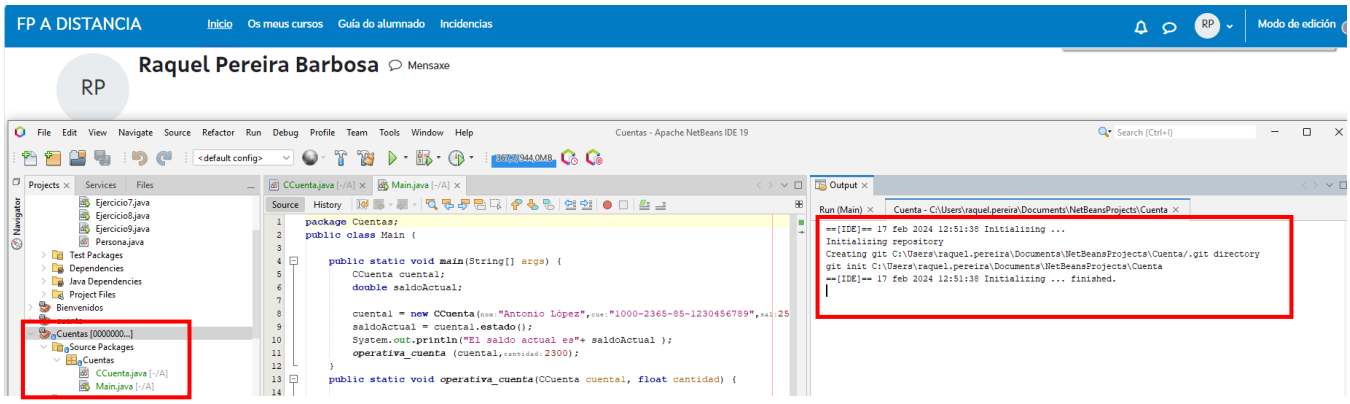
Ahora nos vamos a Netbeans y vamos a Team-Git-Inicializar repositorio



Aquí nos sale la ubicación de nuestro proyecto y es la ubicación que mantendremos así que le decimos ok



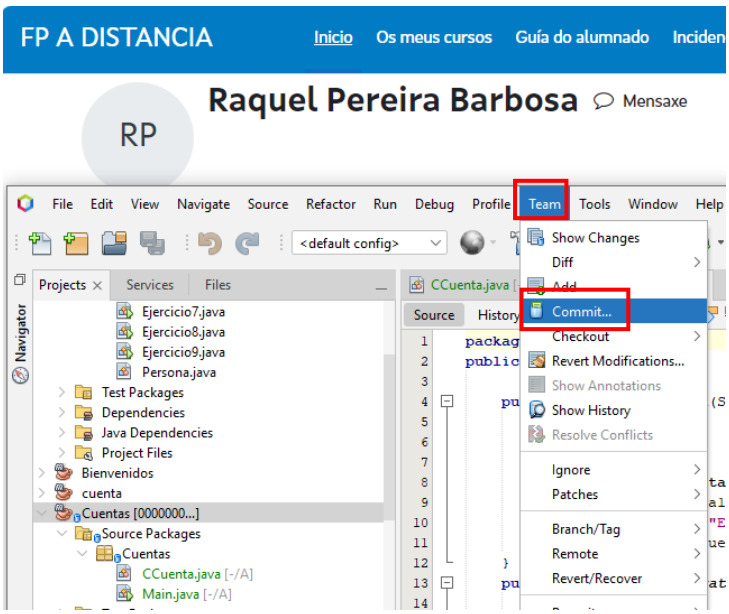
Podemos comprobar que nos ha creado el repositorio en el panel de salida. También nos fijamos que salen con el icono del repositorio nuestro proyecto.

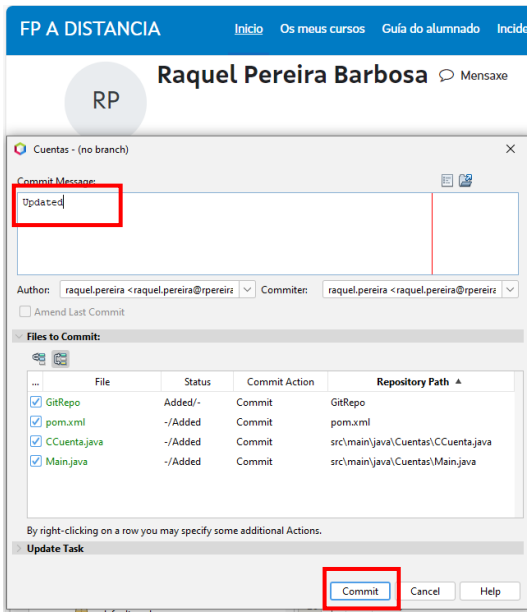


Con esto nos queda configurado GIT para el proyecto y creado el repositorio público

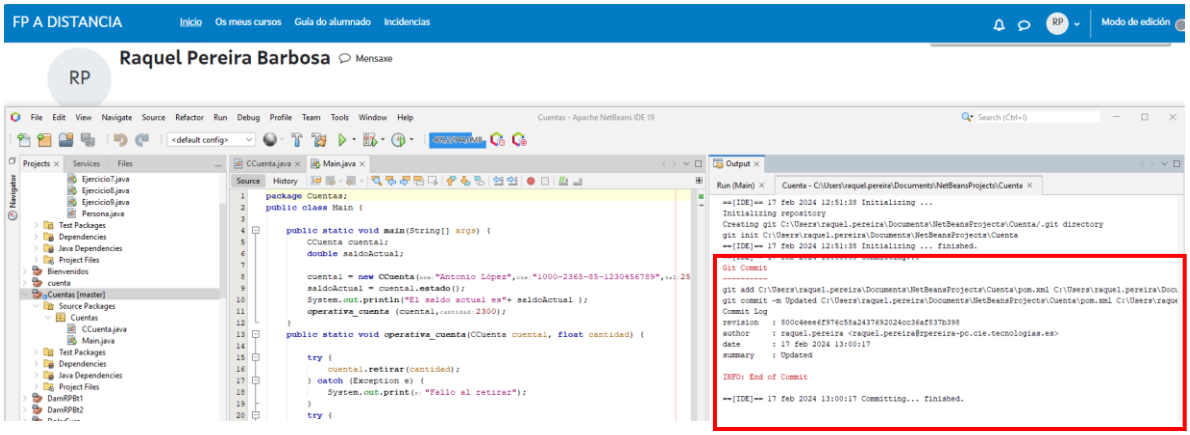
2. Realizar, al menos, una operación commit. Comentando el resultado de la ejecución.

Ahora nos dirigiremos a Team, commit para indicar un mensaje para que se puedan identificar los cambios antes de subirlos al repositorio

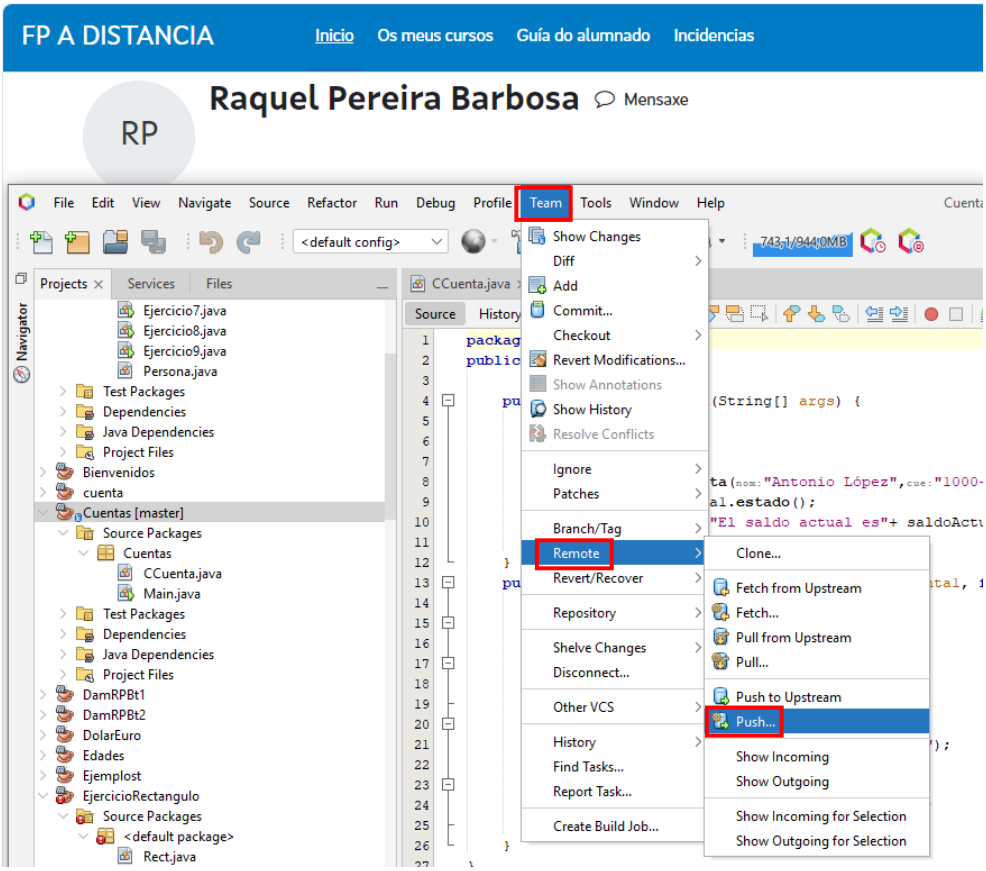




Podemos ver en el panel de salida el inicio y el fin del commint

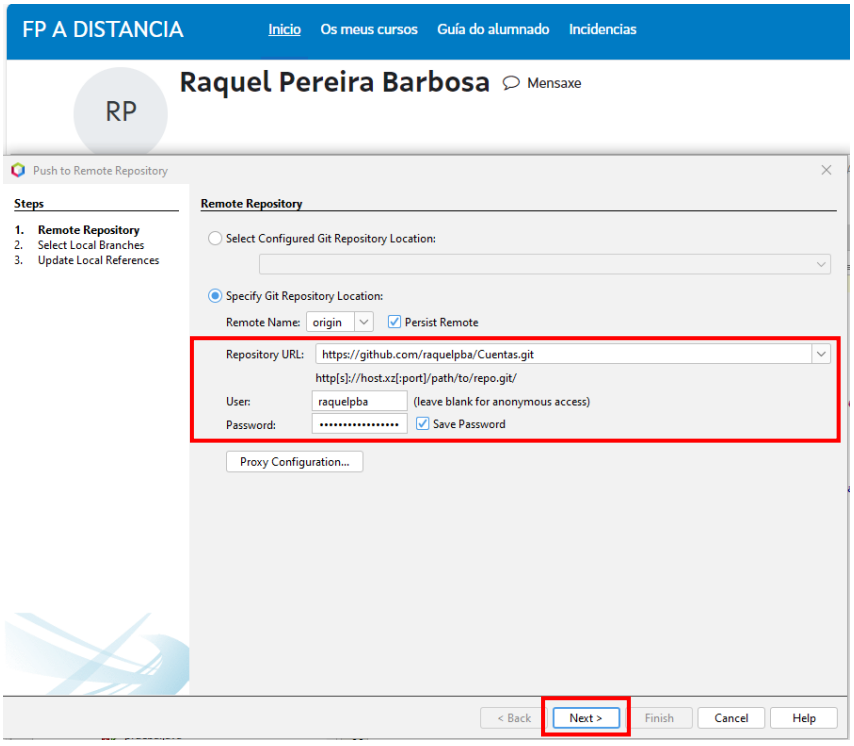


Para que esto tenga sentido subiremos los cambios al repositorio, para eso vamos a Team, Remoto, Push

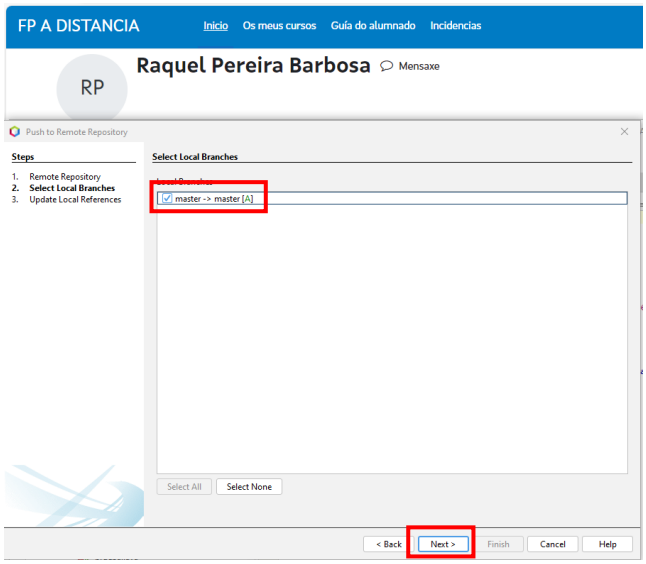


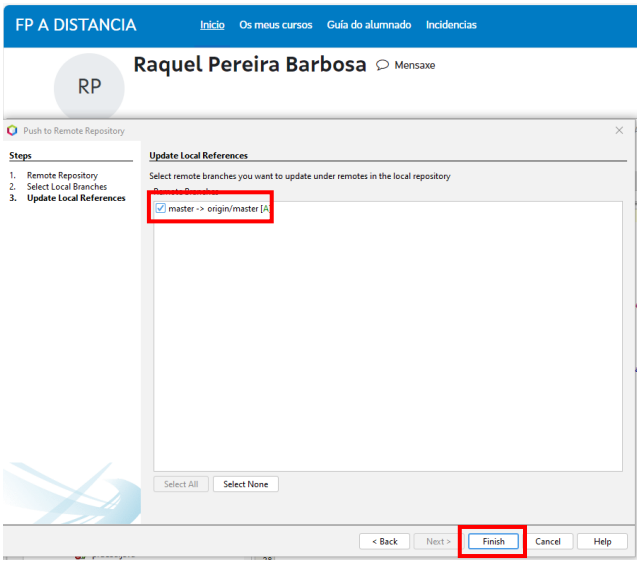
Aquí nos pedirá la URL de nuestro repositorio y nos solicitará nuestras credenciales.

OJO!!!! En las credenciales no valdrá con nuestra contraseña de github, sino que hay que generar un token de seguridad.

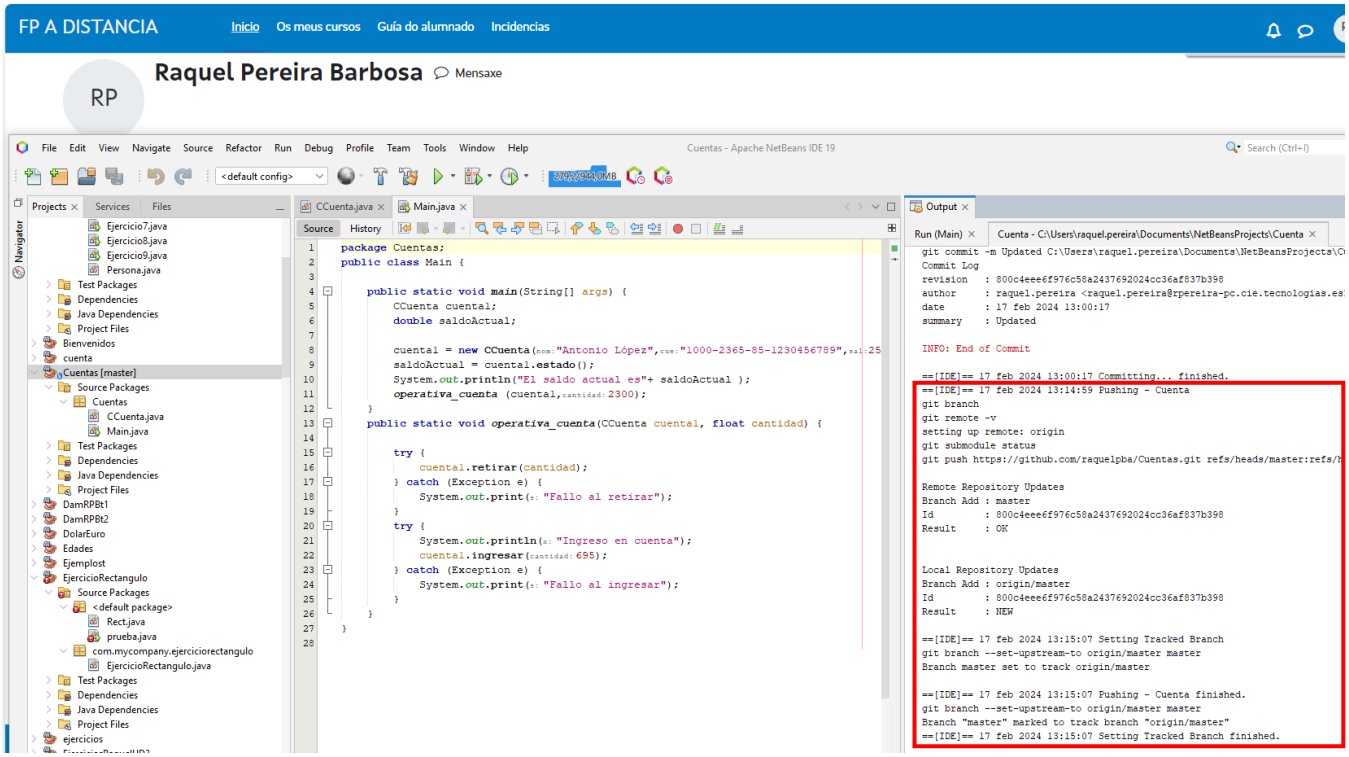


Seleccionamos nuestra rama master y continuamos, actualizamos la rama y finalizamos



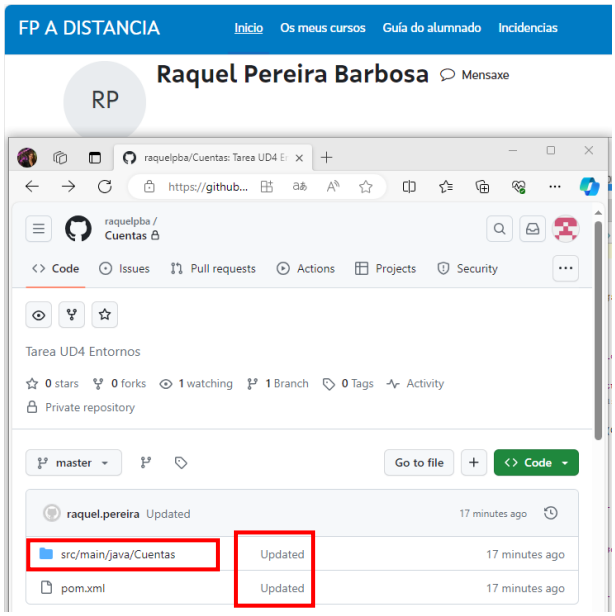


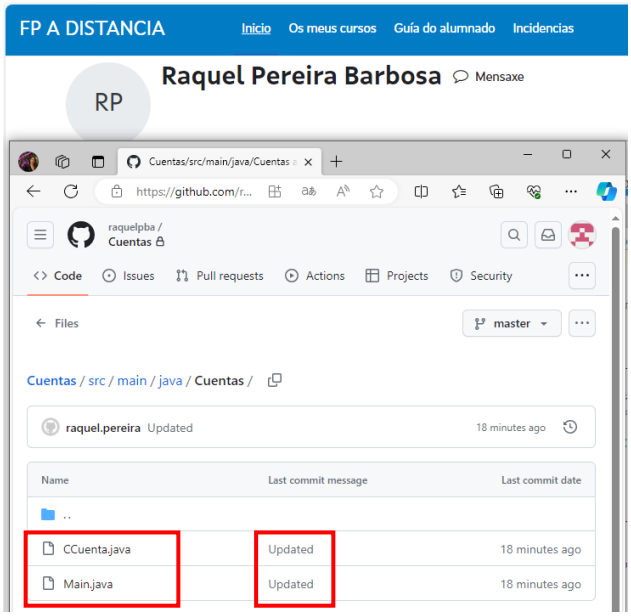
Ahora esperamos a que salga el mensaje en el panel de salida de que se ha subido correctamente



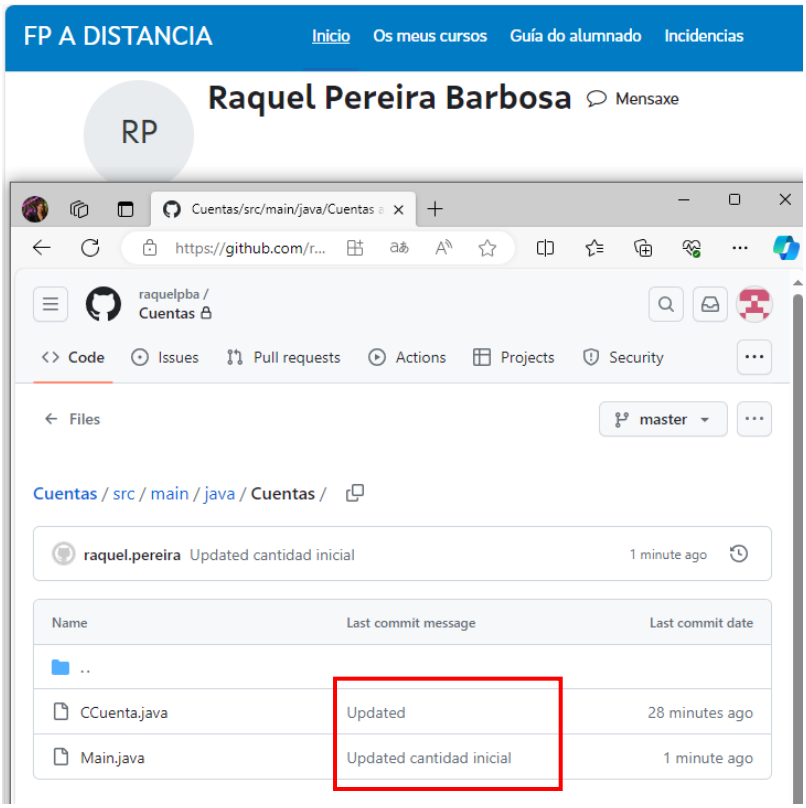
Ahora comprobaremos en nuestro Github que nuestro proyecto está subido, y vemos que así es

Nos fijamos que el mensaje que pusimos en commit aparece en la columna “Last commit message”



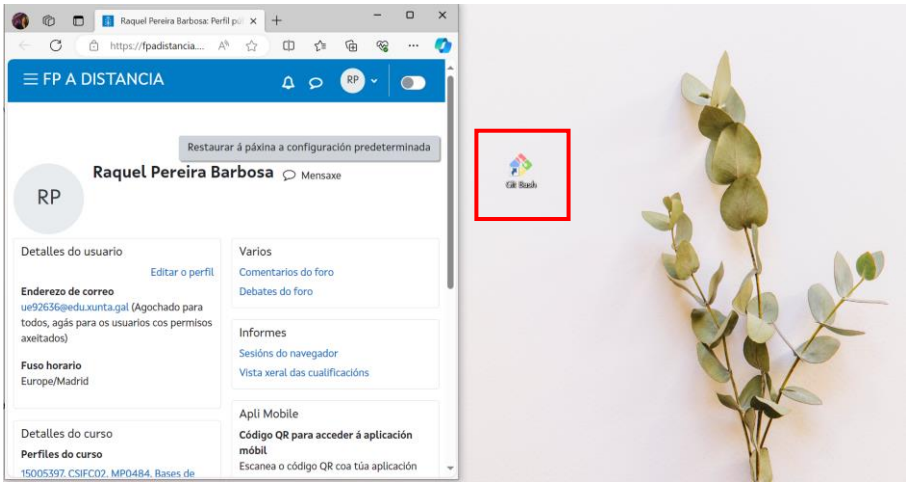


Hacemos otro commit y push para hacer una prueba de que nos indica el cambio



3. Mostrar el historial de versiones para el proyecto mediante un comando desde consola.

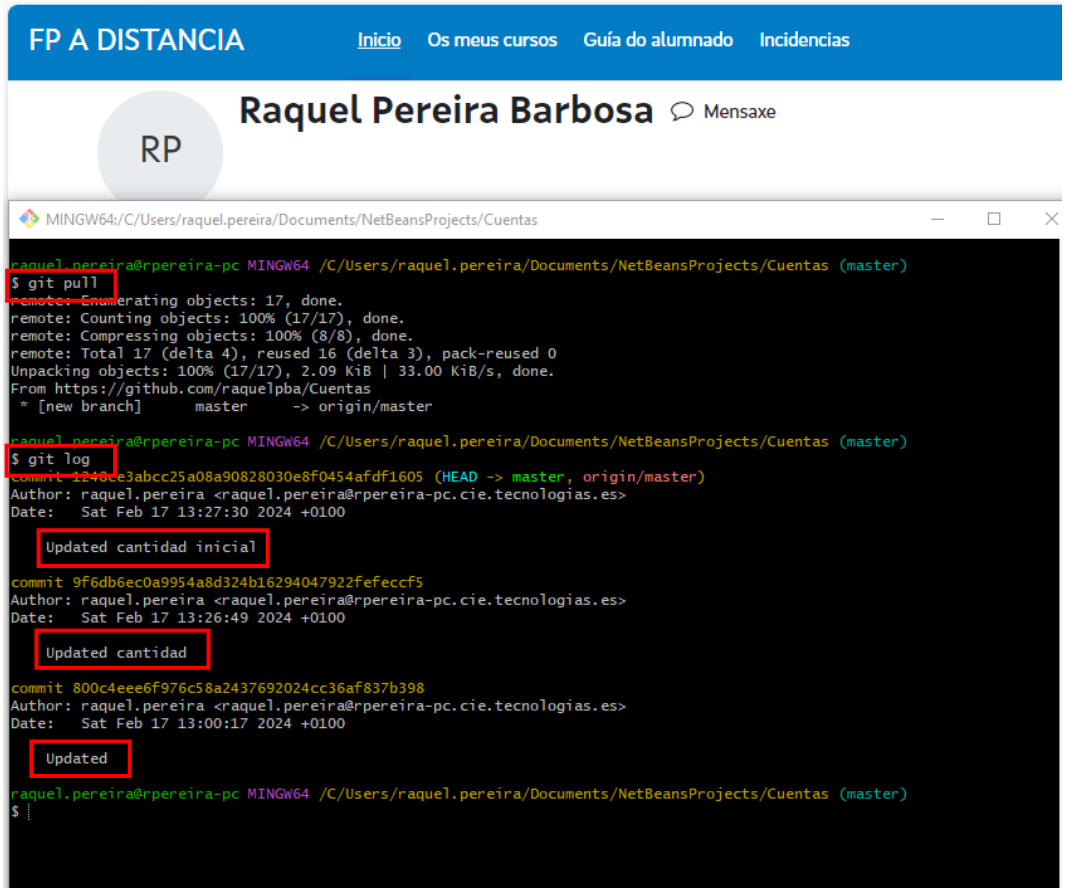
Para esto hemos instalado GIT durante la revisión de la unidad:



Lo abrimos y nos ubicamos en nuestra carpeta donde está el proyecto y donde está nuestro repositorio local:

1. Sincronizamos nuestro repositorio remoto con el local, para ello hacemos un git pull
2. Para obtener el historial de versiones una vez sincronizado utilizaremos el comando git log

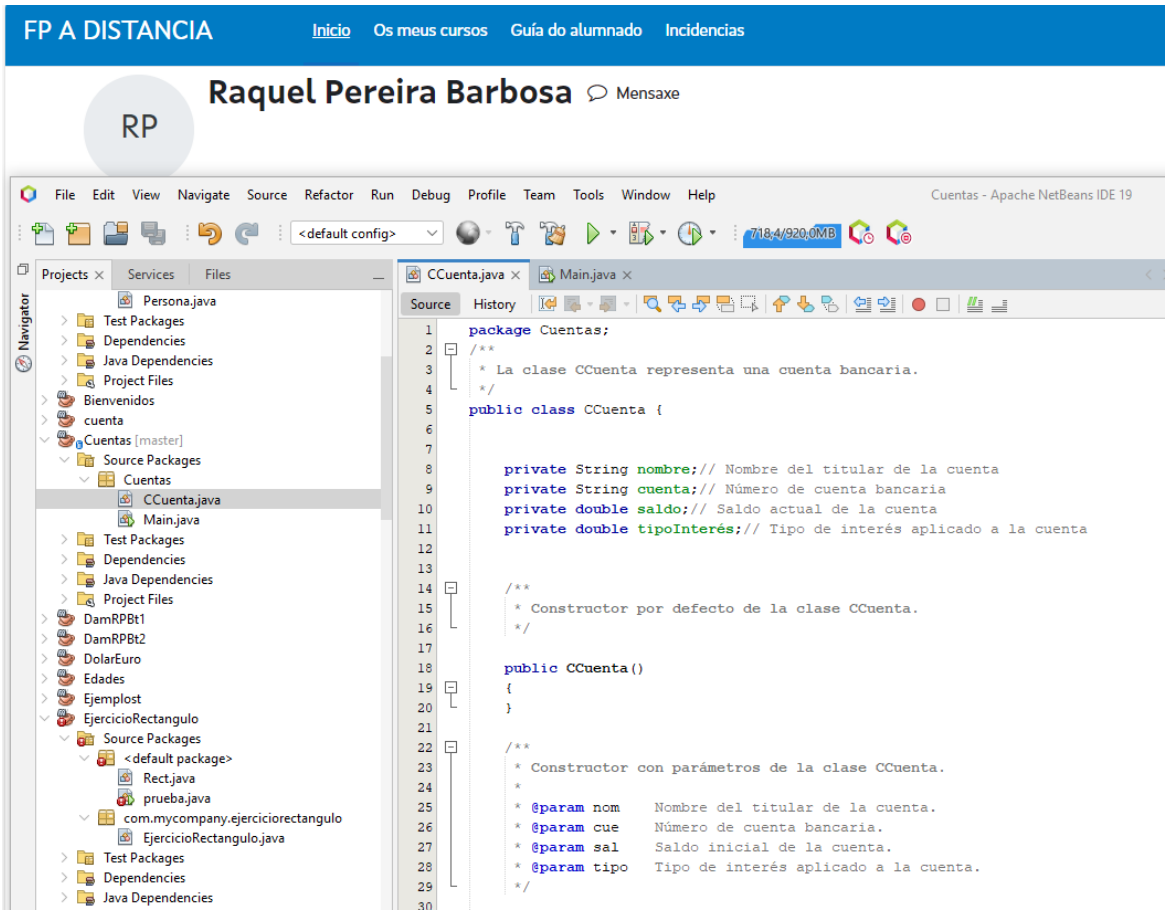
Y con esto podemos ver todos los commits que hemos realizado:



JAVADOC

1. Insertar comentarios JavaDoc en la clase CCuenta.

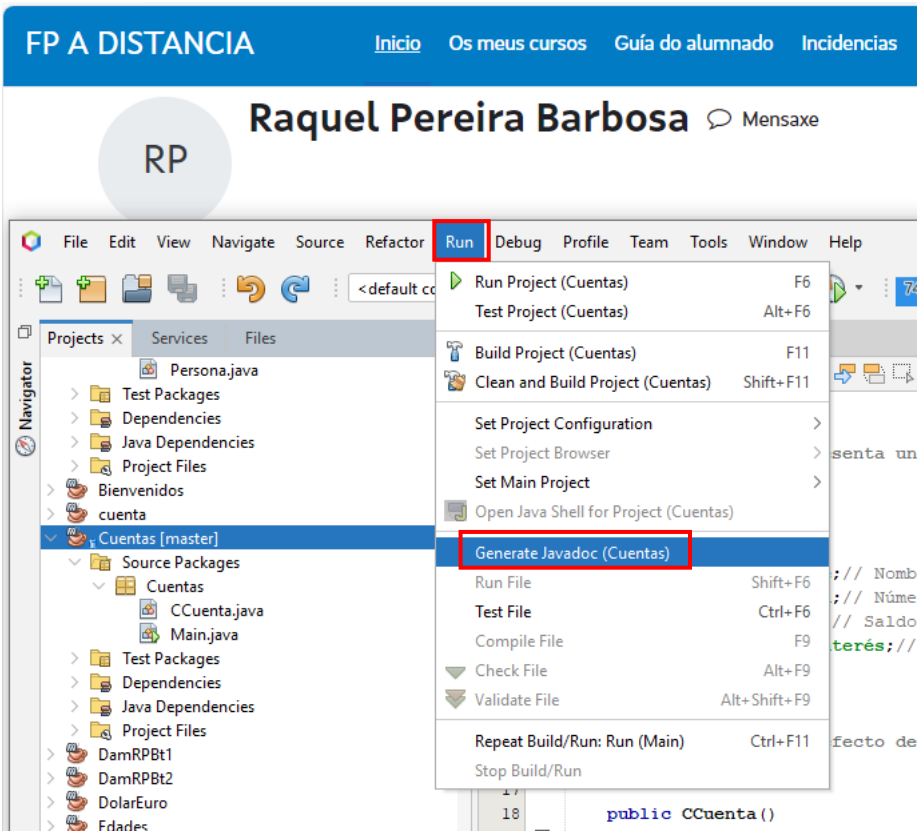
Añadimos algún comentario a mayores de los generados en la refactorización de getters y setters, y generamos comentarios para documentar parámetros de métodos y constructores.



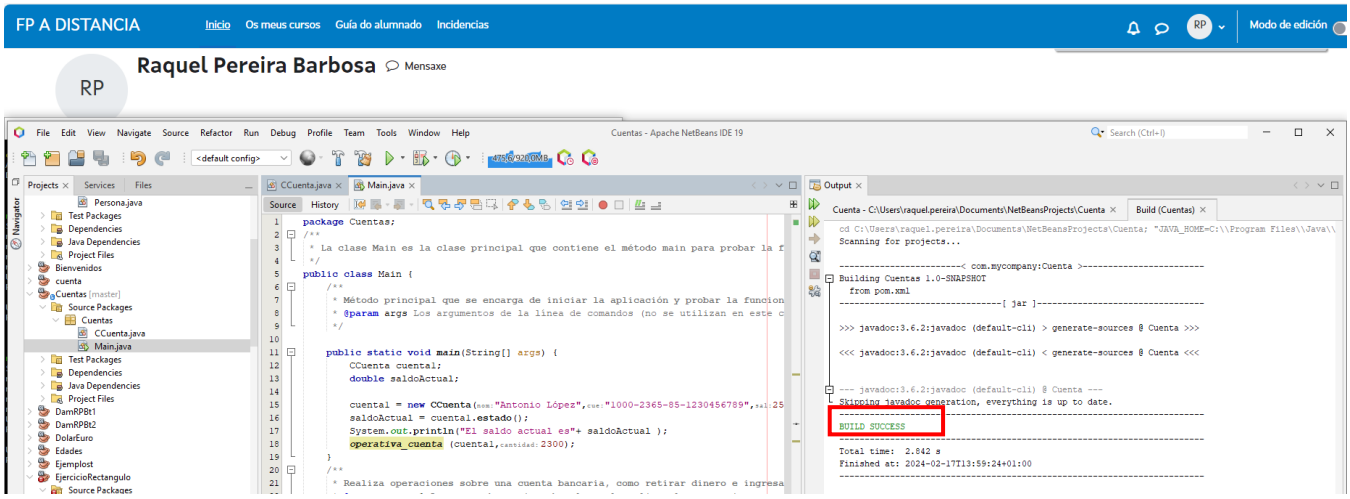
(Como adjuntaremos el proyecto a la tarea ahí se podrán ver todos los comentarios)

2. Generar documentación JavaDoc para todo el proyecto y comprueba que abarca todos los métodos y atributos de la clase CCuenta.

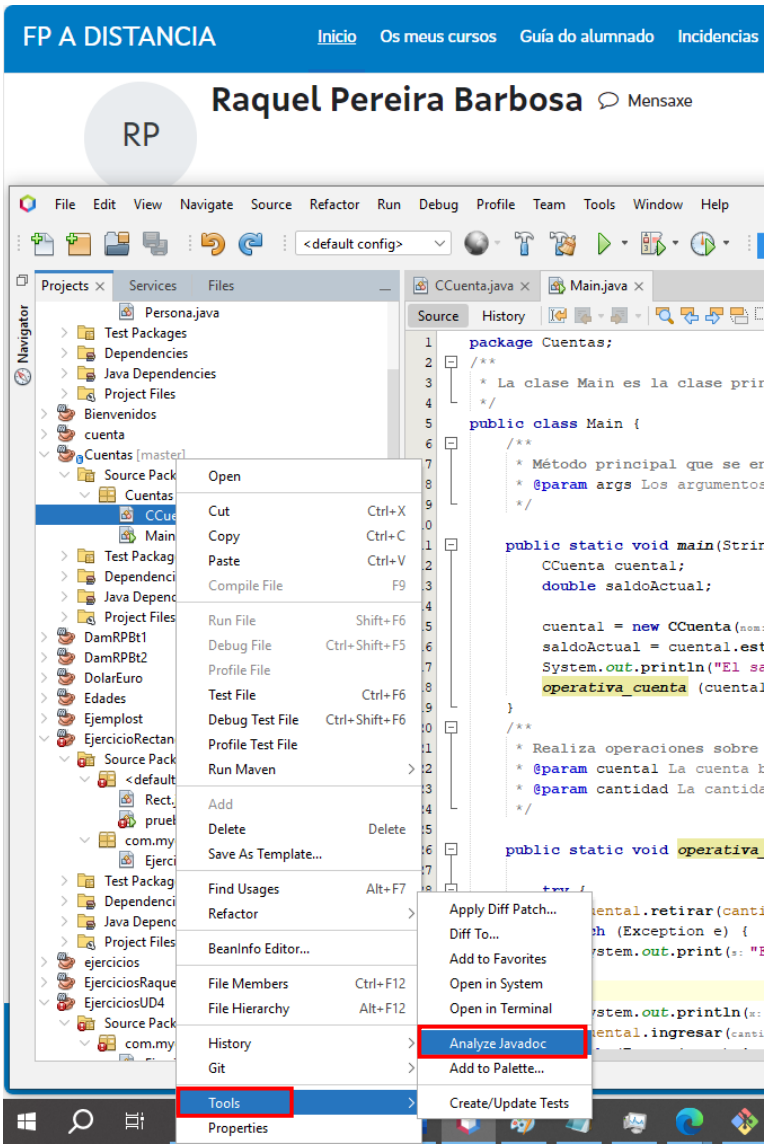
Para generar el JavaDoc nos dirigimos a la pestaña Run. Generar JavaDoc



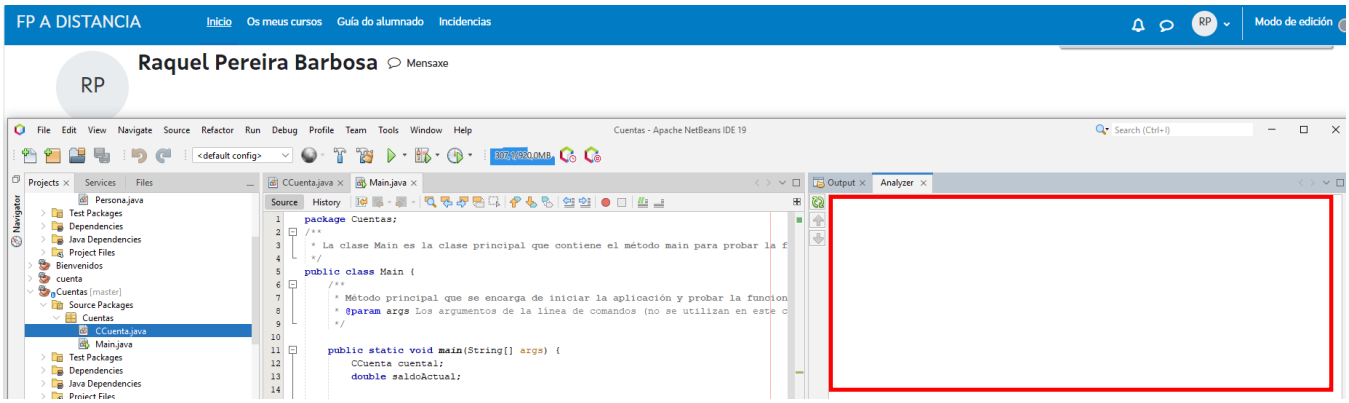
Vemos que abarca todos los métodos y atributos de la clase cuenta ya que lo genera sin ningún error:



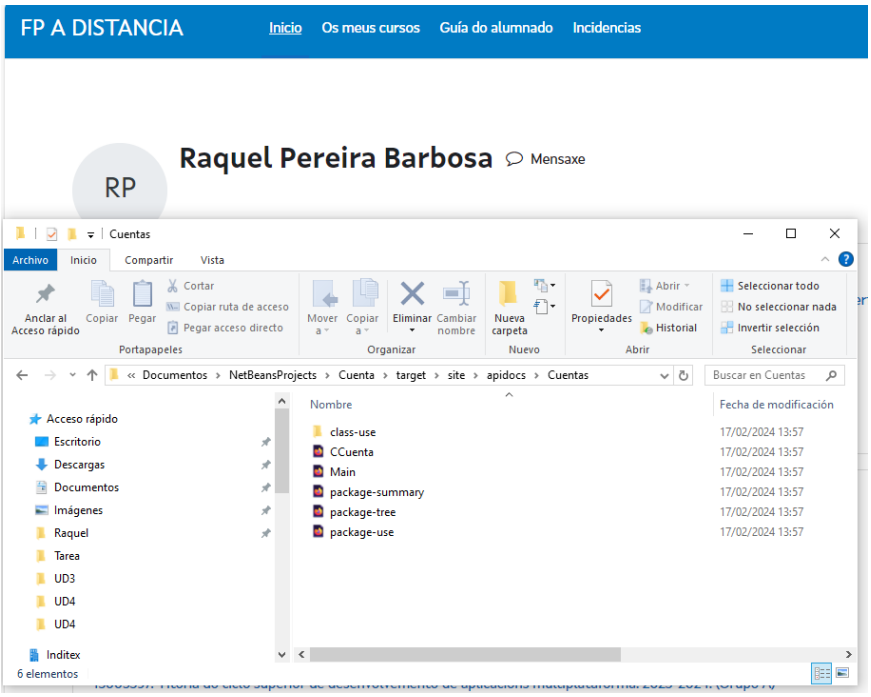
Además, si vamos a nuestro fichero CCuentas y analizamos el JavaDoc no nos da ninguna opción que nos falte, por tanto, está todo incluido:



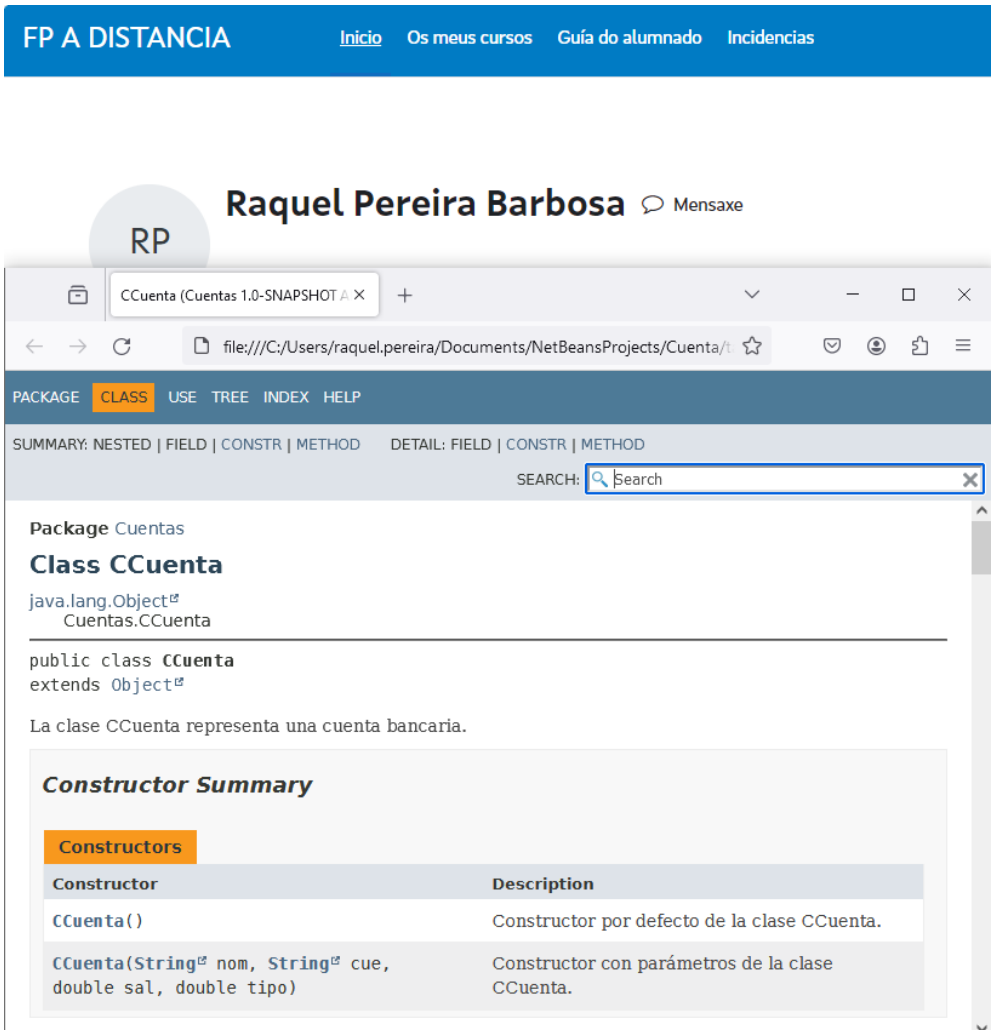
No da opciones de cambio por lo que está todo correcto



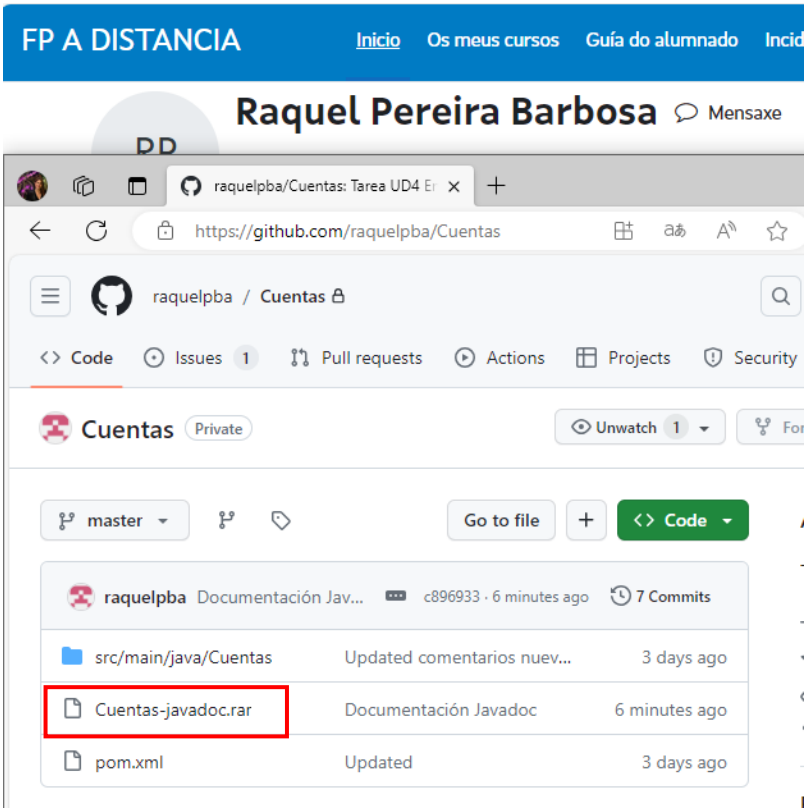
Nuestro Javadoc lo podemos encontrar en la carpeta que ha generado netbeans para nuestro proyecto dentro de apidocs



Lo abrimos y vemos que abre perfectamente, lo adjuntamos en esta entrega y lo podemos encontrar también en nuestro GitHub



Nuestro JavaDoc lo hemos subido a GitHub en un comprimido con todo lo necesario para que se pueda visualizar correctamente:



Hemos añadido una issue para explicarlo correctamente: