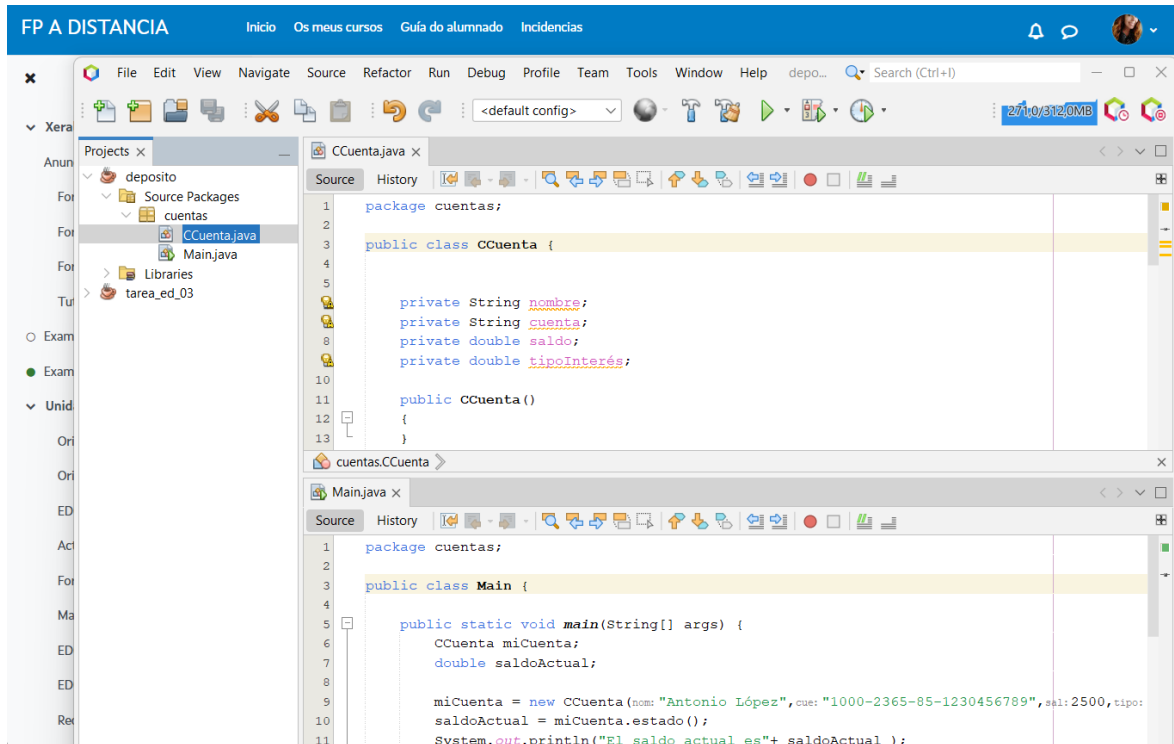


ED04 TAREA

REFACTORIZACIÓN

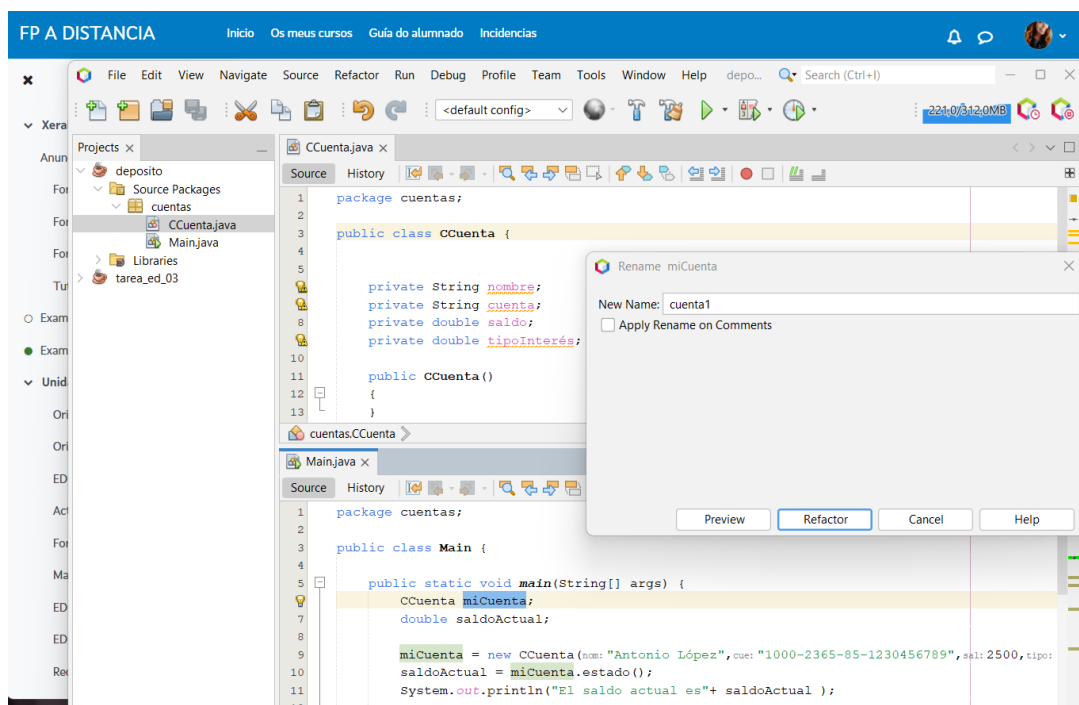
1. Las clases deberán formar parte del paquete cuentas.

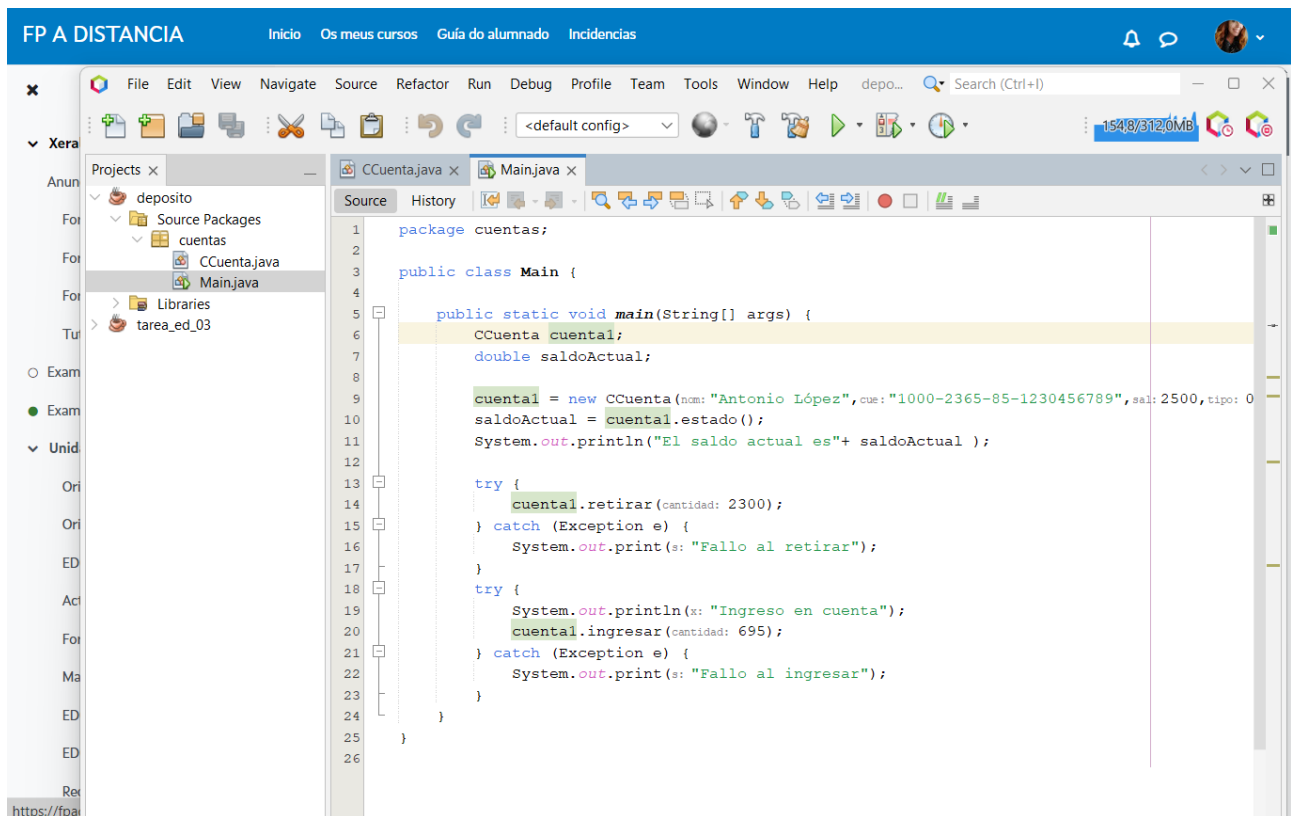
Seleccionamos cada clase → Refactor → Move... → To Package cuentas



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

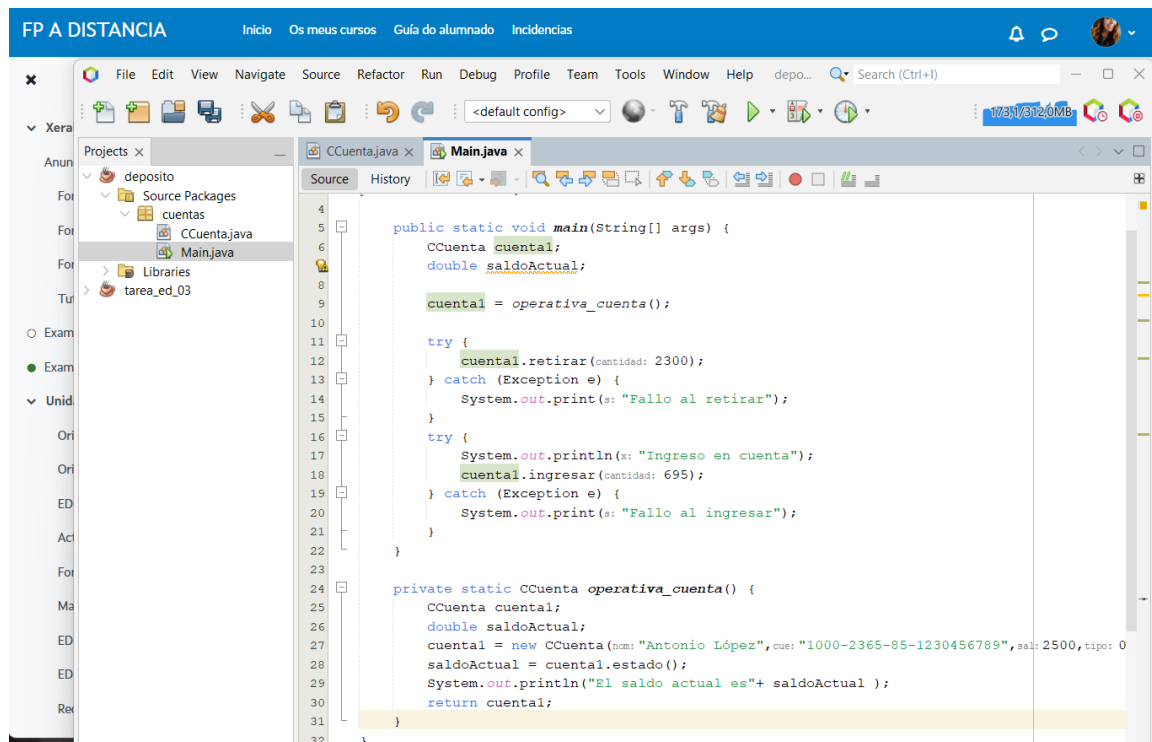
Seleccionamos la variable "miCuenta" → Refactor → Rename → "cuenta1"





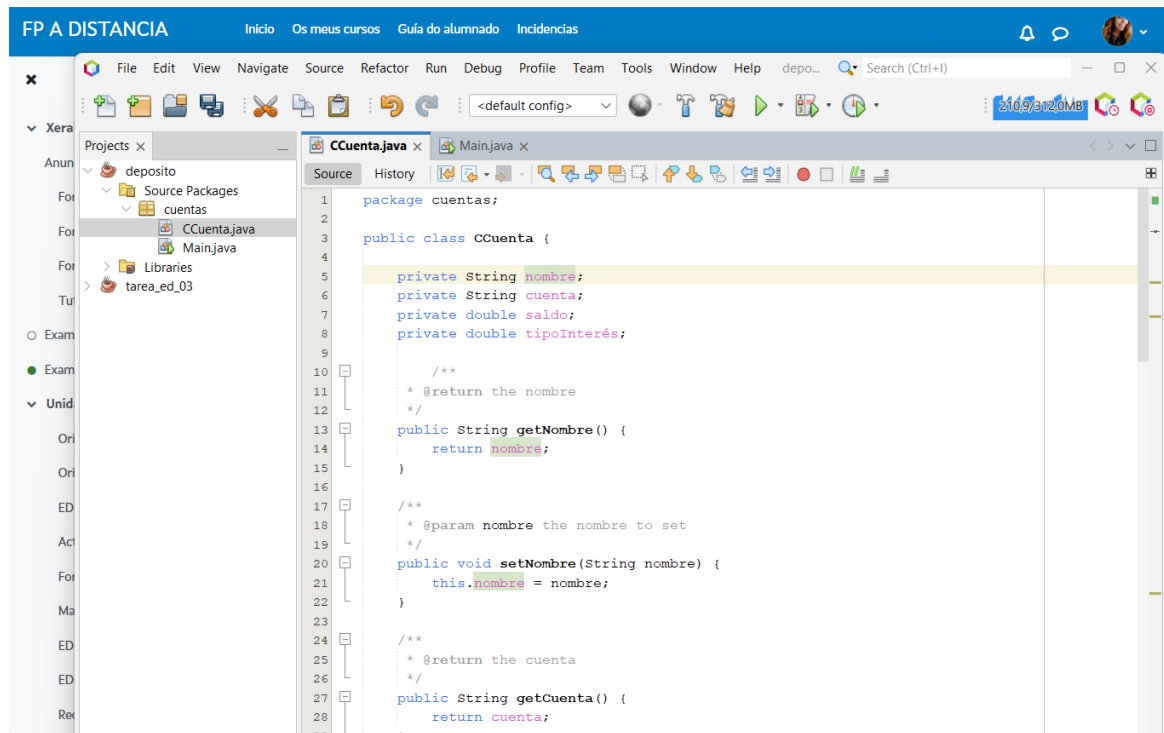
3. Introducir el método operativa_cuenta, que englobe las sentencias de la clase Main que operan con el objeto cuenta1.

Seleccionamos las sentencias de la clase Main que operan con el objeto cuenta1 → Refactor → Introduce → Method → “operativa_cuenta”

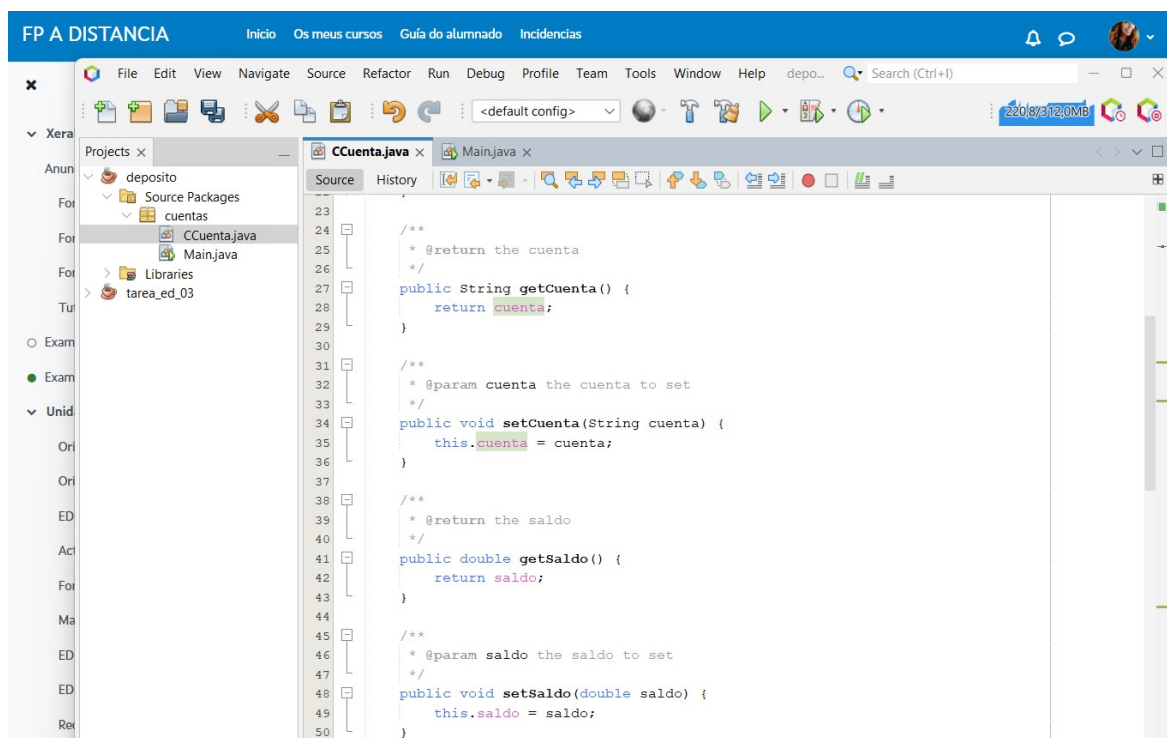


4. Encapsular los atributos de la clase CCuenta.

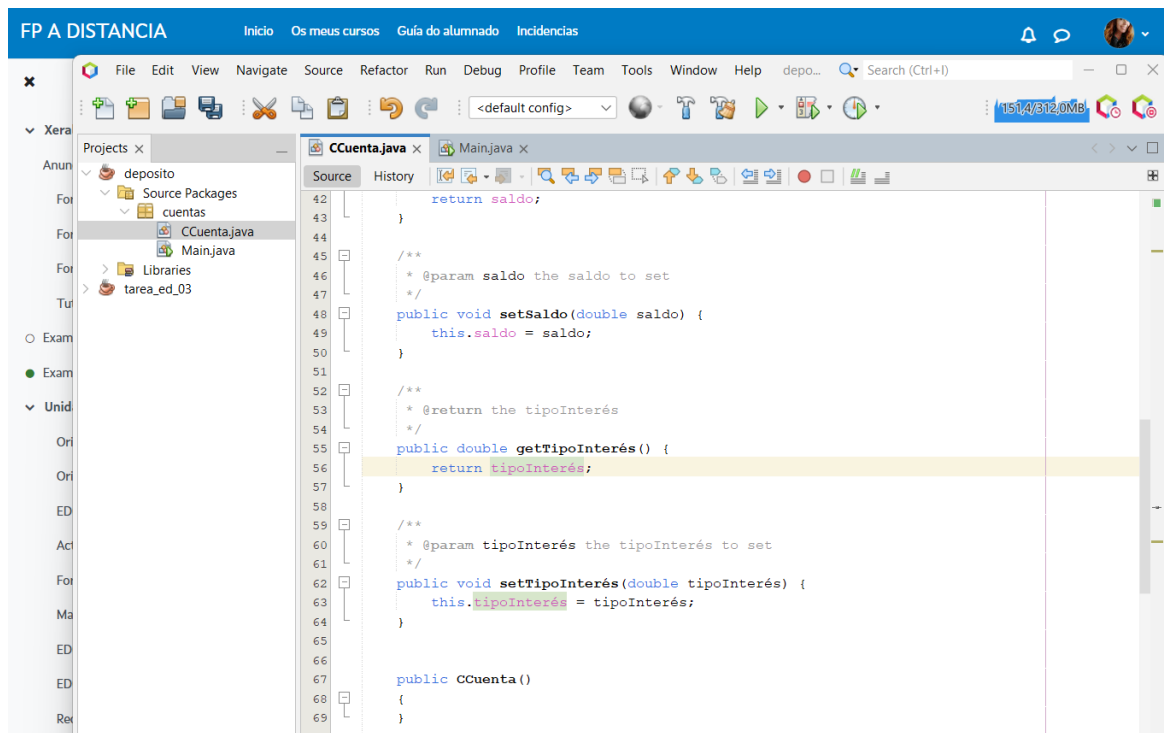
Seleccionar los atributos → Refactor → Encapsulate Fields



```
1 package cuentas;
2
3 public class CCuenta {
4
5     private String nombre;
6     private String cuenta;
7     private double saldo;
8     private double tipoInterés;
9
10    /**
11     * @return the nombre
12     */
13    public String getNombre() {
14        return nombre;
15    }
16
17    /**
18     * @param nombre the nombre to set
19     */
20    public void setNombre(String nombre) {
21        this.nombre = nombre;
22    }
23
24    /**
25     * @return the cuenta
26     */
27    public String getCuenta() {
28        return cuenta;
29    }
30
31    /**
32     * @param cuenta the cuenta to set
33     */
34    public void setCuenta(String cuenta) {
35        this.cuenta = cuenta;
36    }
37
38    /**
39     * @return the saldo
40     */
41    public double getSaldo() {
42        return saldo;
43    }
44
45    /**
46     * @param saldo the saldo to set
47     */
48    public void setSaldo(double saldo) {
49        this.saldo = saldo;
50    }
51 }
```

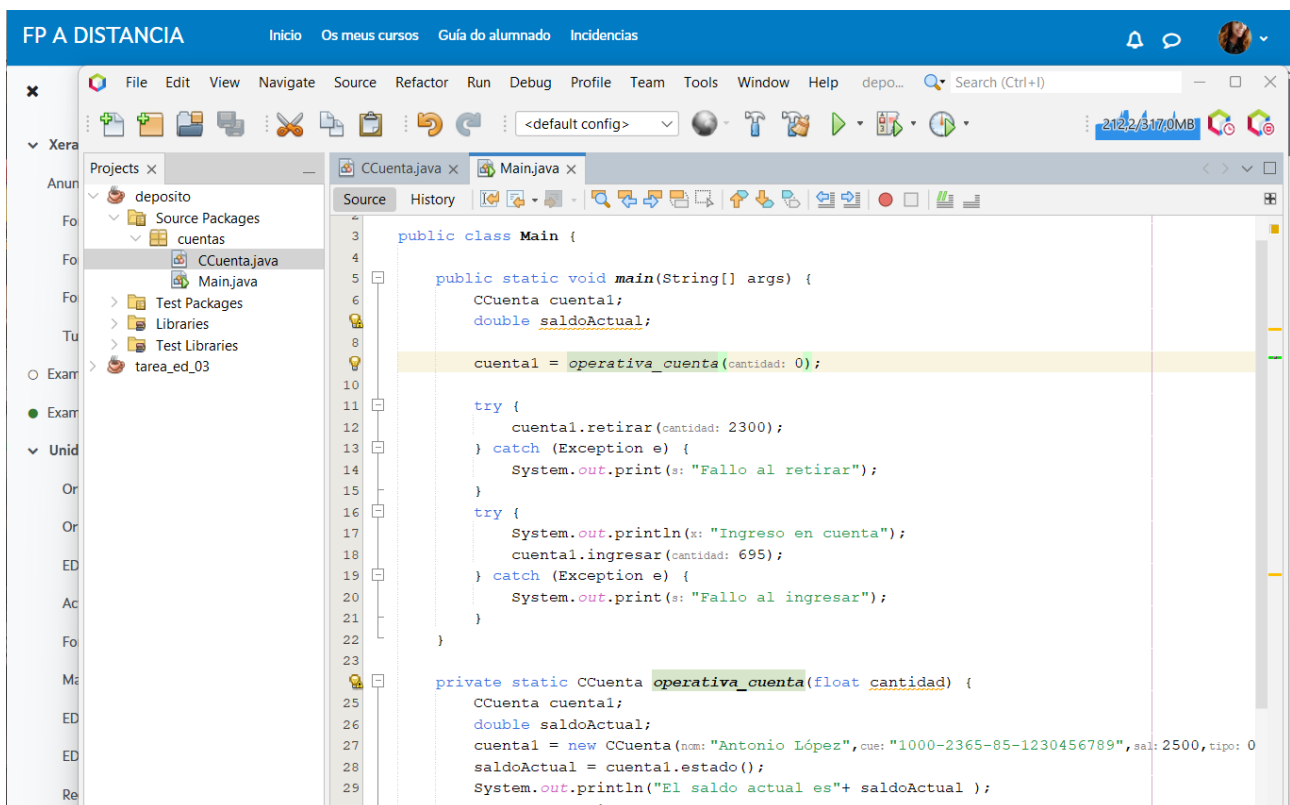


```
23
24 /**
25  * @return the cuenta
26  */
27 public String getCuenta() {
28     return cuenta;
29 }
30
31 /**
32  * @param cuenta the cuenta to set
33  */
34 public void setCuenta(String cuenta) {
35     this.cuenta = cuenta;
36 }
37
38 /**
39  * @return the saldo
40  */
41 public double getSaldo() {
42     return saldo;
43 }
44
45 /**
46  * @param saldo the saldo to set
47  */
48 public void setSaldo(double saldo) {
49     this.saldo = saldo;
50 }
51 }
```



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

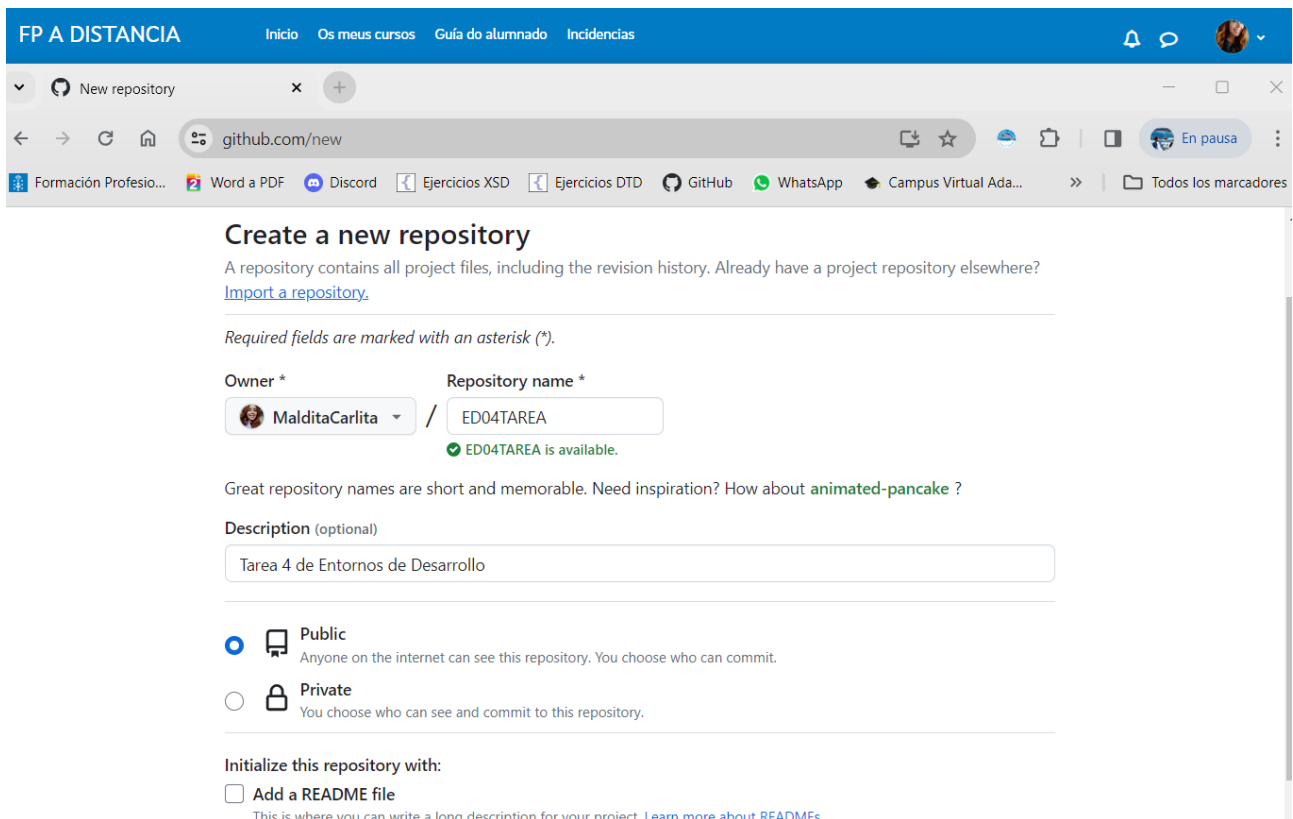
Seleccionamos el método → Refactor → Change Method parameters → tipo “float”, nombre “cantidad”, valor por defecto 0



GIT

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

Me registro en GitHub, inicio sesión y creo un repositorio nuevo y público.



FP A DISTANCIA Inicio Os meus cursos Guía do alumnado Incidencias

New repository

github.com/new

Formación Profesio... Word a PDF Discord Ejercicios XSD Ejercicios DTD GitHub WhatsApp Campus Virtual Ada... Todos los marcadores

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

MalditaCarlita / ED04TAREA

ED04TAREA is available.

Great repository names are short and memorable. Need inspiration? How about [animated-pancake](#) ?

Description (optional)

Tarea 4 de Entornos de Desarrollo

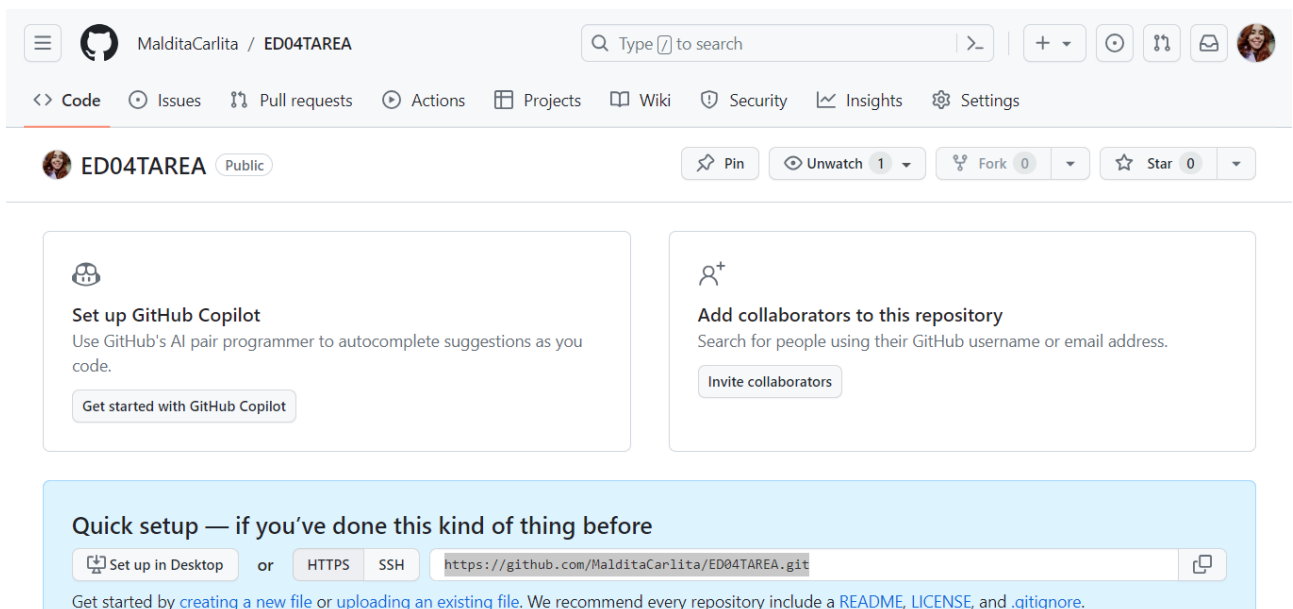
☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:

☒ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Copiamos la URL del repositorio creado.



MalditaCarlita / ED04TAREA

Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

ED04TAREA Public

Pin Unwatch 1 Fork 0 Star 0

Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
[Get started with GitHub Copilot](#)

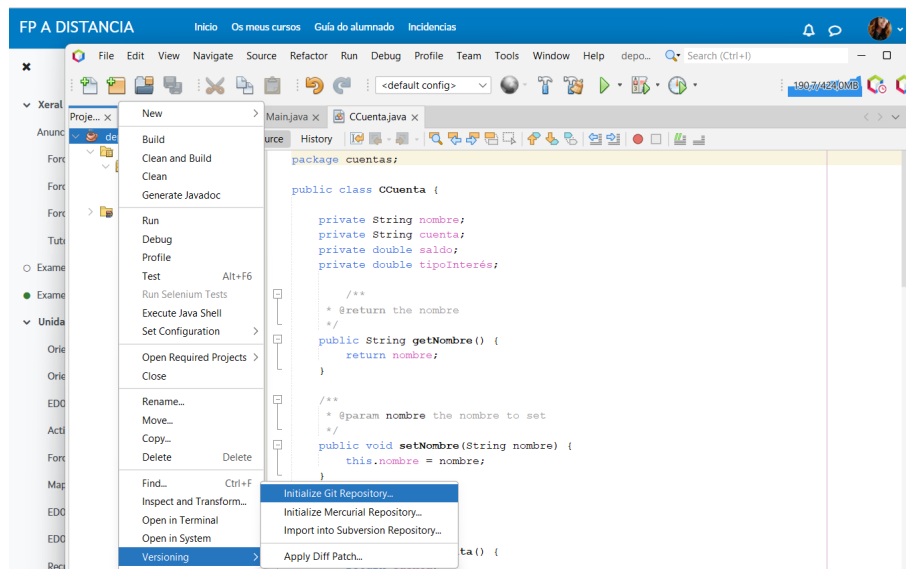
Add collaborators to this repository
Search for people using their GitHub username or email address.
[Invite collaborators](#)

Quick setup — if you've done this kind of thing before

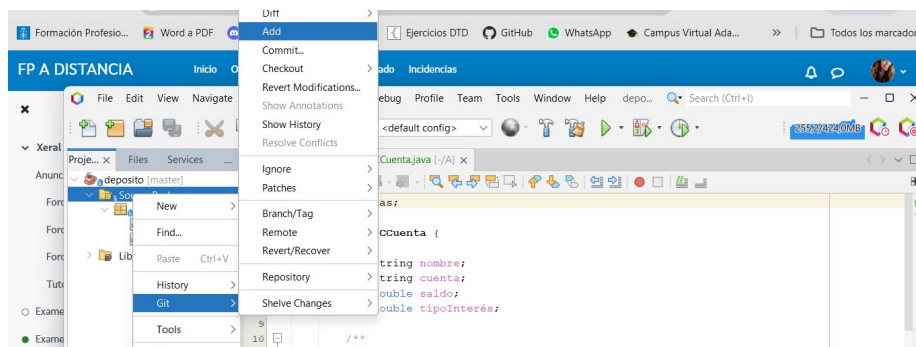
[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/MalditaCarlita/ED04TAREA.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

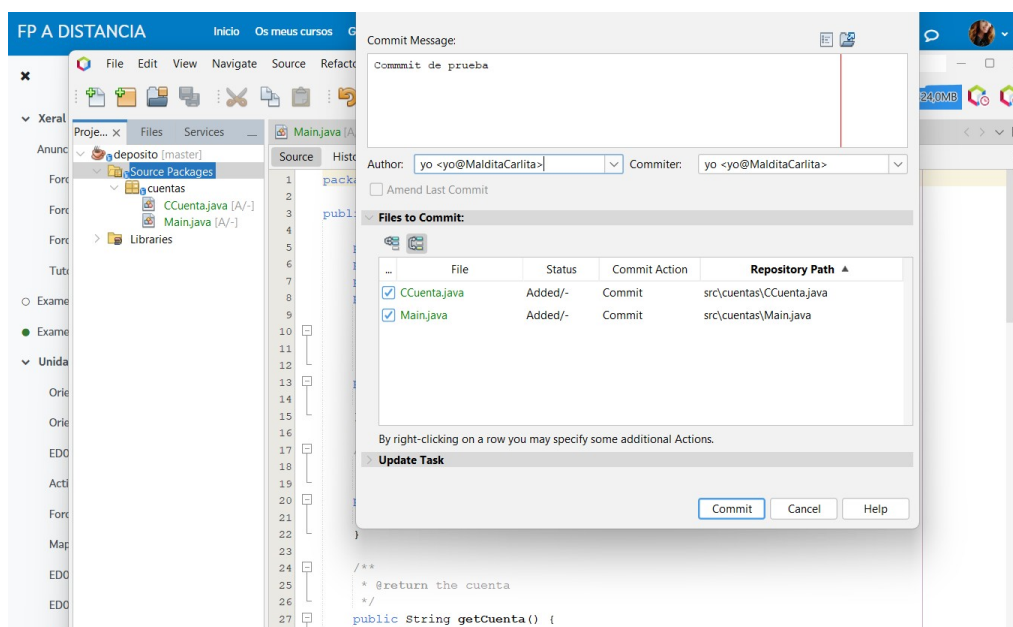
2. **Realizar, al menos, una operación commit. Comentando el resultado de la ejecución.** Primero seleccionamos el proyecto, clicamos con botón derecho en Versioning → Initialize Git Repository y seleccionamos el directorio.



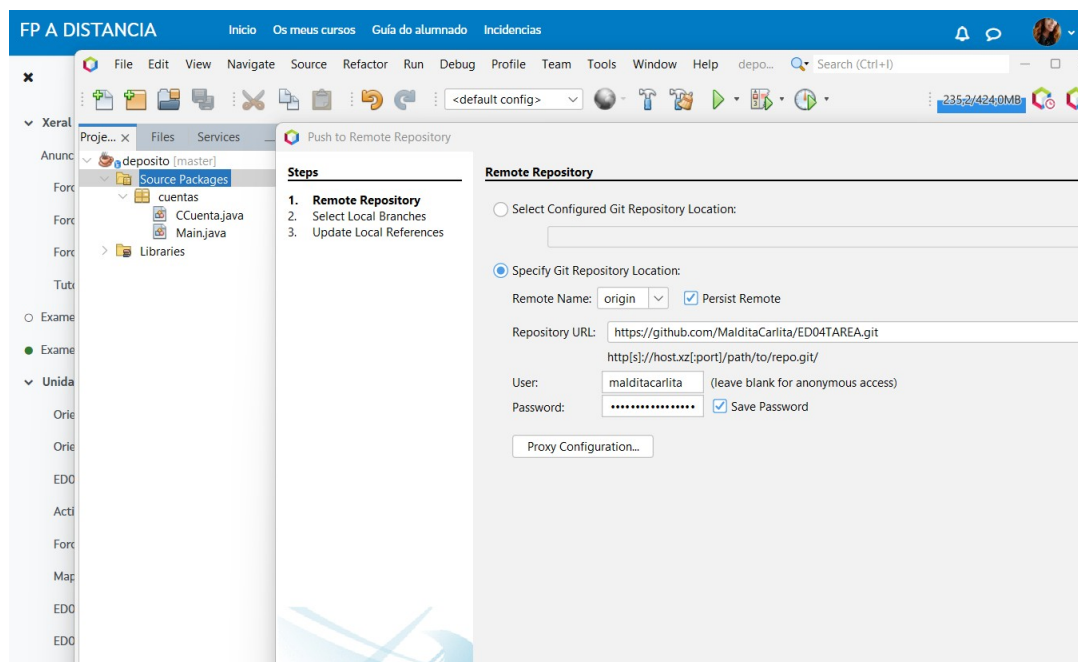
Las clases ahora se muestran de color verde ya que aún no han sido subidas al repositorio. Para añadirlas, seleccionamos la carpeta y con el botón derecho seleccionamos Git → Add. Esto las prepara para el próximo Commit (momento en el que se añadirá al repositorio local). Se pueden añadir tantos archivos como se quiera mediante Add.



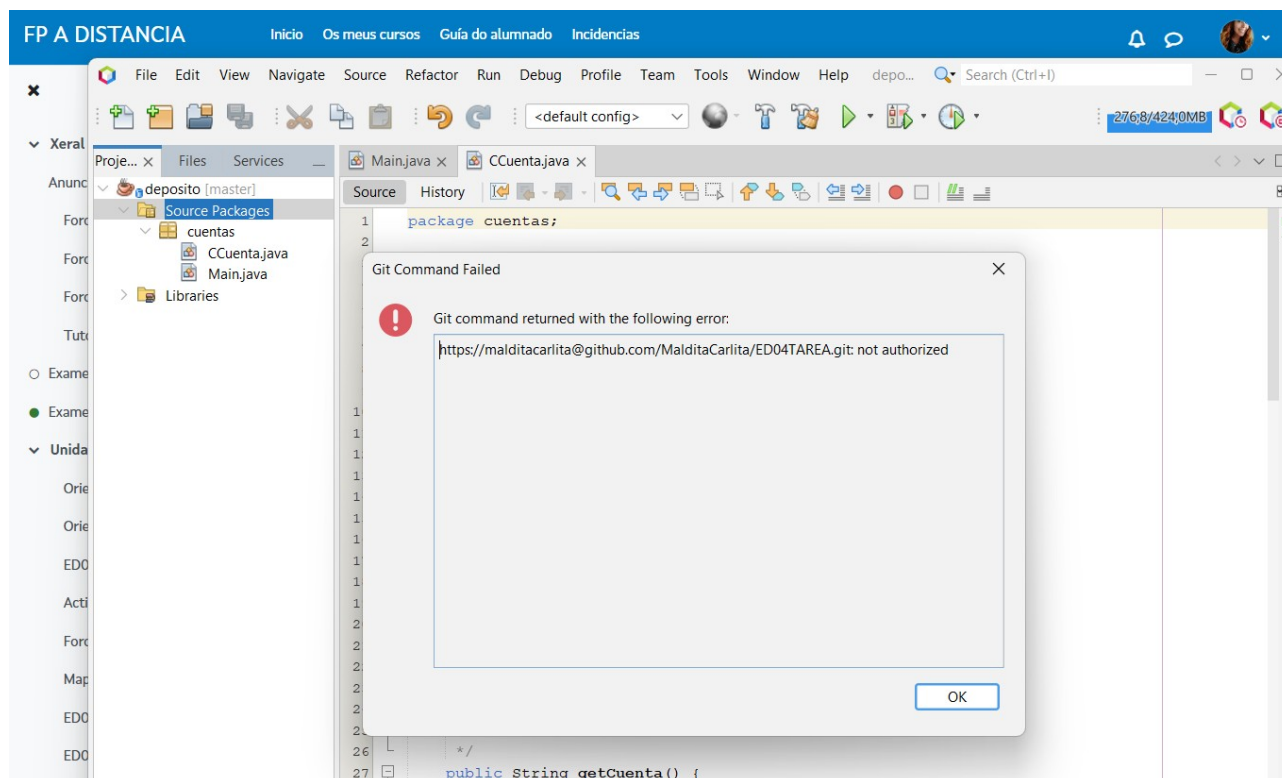
Una vez añadidas, seleccionamos la carpeta y con el botón derecho seleccionamos Git → Commit (Punto de versionamiento). Al hacer clic en Commit, el archivo se incluirá en el repositorio local y estará sujeto al control de versiones. Además habrá cambiado al color por defecto.



Una vez realizado el commit, seleccionamos la carpeta y con el botón derecho seleccionamos Git → Remote → Push y le indicamos el repositorio de GitHub al que queremos que suba el código fuente copiando el link del repositorio en nuestro perfil de GitHub.



Cuando intento realizar el push me informa de que no tengo autorización.



Para corregir este error, tengo que ir a mi perfil de GitHub → Settings → Developer Settings → Personal Access Tokens → Tokens Classic → Generate New Tokens para **generar un token**, que colocaremos como password cuando intentemos hacer el push.

Para generarlo, es necesario seleccionar todas las opciones que nos aparece y clicar en Generate Token.

FP A DISTANCIA Inicio Os meus cursos Guia do alumnado Incidencias

New Personal Access Token (Classic)

github.com/settings/tokens/new

GitHub Apps
OAuth Apps
Personal access tokens
Fine-grained tokens (Beta)
Tokens (classic)

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

Token Netbeans

What's this token for?

Expiration *

30 days The token will expire on Sat, Mar 30 2024

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows
<input checked="" type="checkbox"/> write:packages	Upload packages to GitHub Package Registry

FP A DISTANCIA Inicio Os meus cursos Guia do alumnado Incidencias

Personal Access Tokens (Classic)

github.com/settings/tokens

Push to Remote Repository

Steps

1. Remote Repository
2. Select Local Branches
3. Update Local References

Remote Repository

☐ Select Configured Git Repository Location:

origin:https://github.com/MalditaCarlita/ED04TAREA.git

☒ Specify Git Repository Location:

Remote Name: origin ☒ Persist Remote

Repository URL: https://github.com/MalditaCarlita/ED04TAREA.git

http[s]://host.xz[:port]/path/to/repo.git/

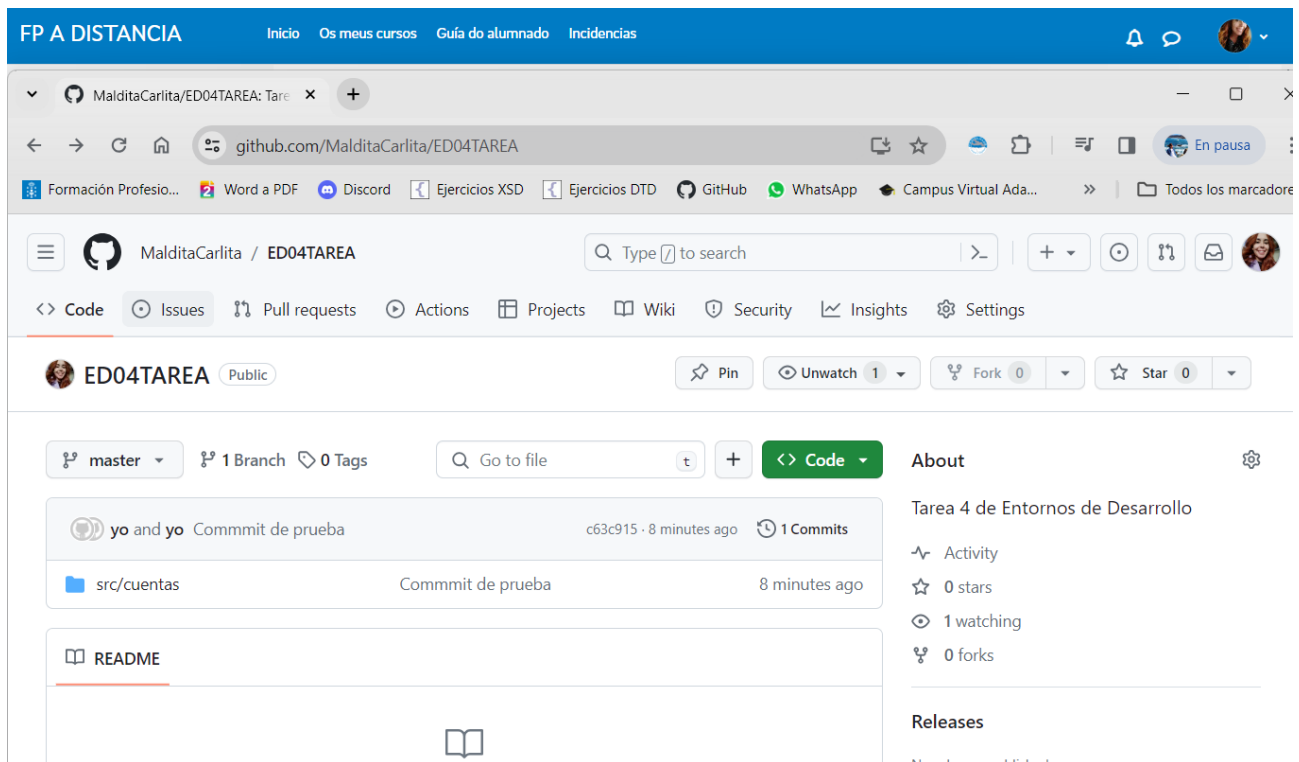
User: malditacarlita (leave blank for anonymous acce...)

Password: ***** ☒ Save Password

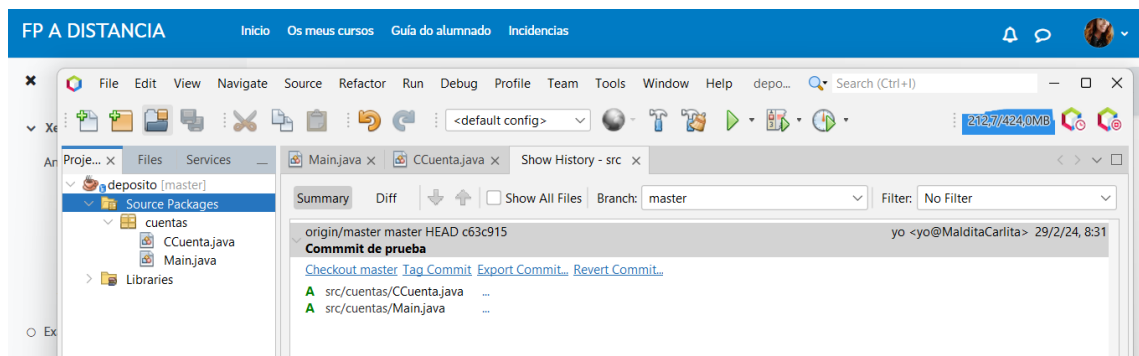
Proxy Configuratio...

< Back Next > Finish Cancel Help

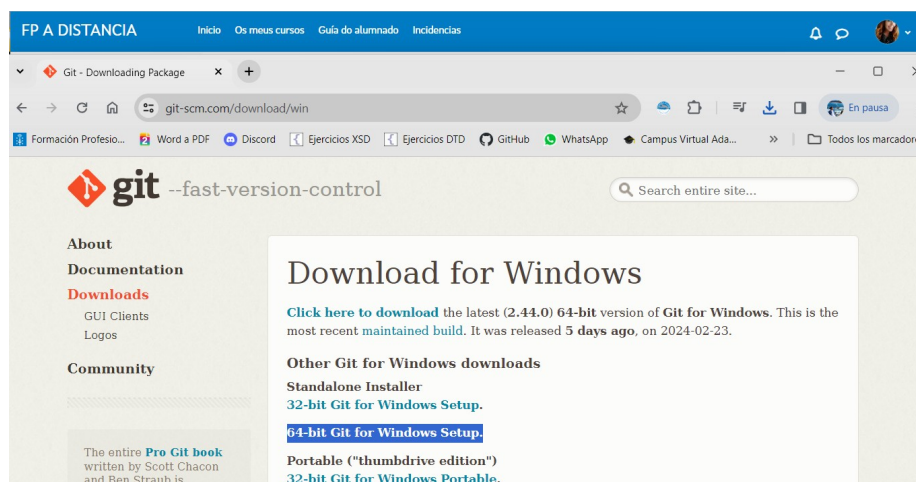
Sustituimos la contraseña por el token generado y ya nos permite hacer el push. Si ahora recargamos el repositorio en GitHub, observamos los cambios.

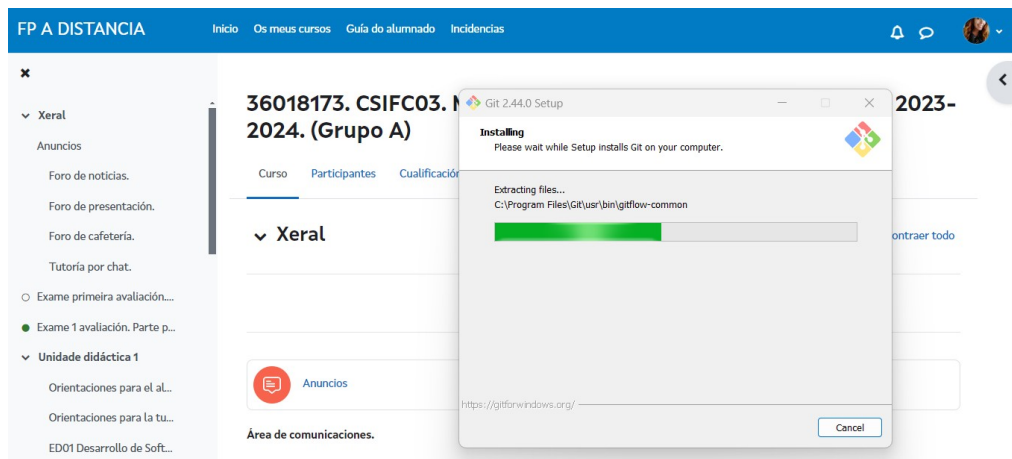


3. **Mostrar el historial de versiones para el proyecto mediante un comando desde consola.** Si seleccionamos la carpeta con botón derecho y seleccionamos Git → Show History nos mostrará el historial de versiones.

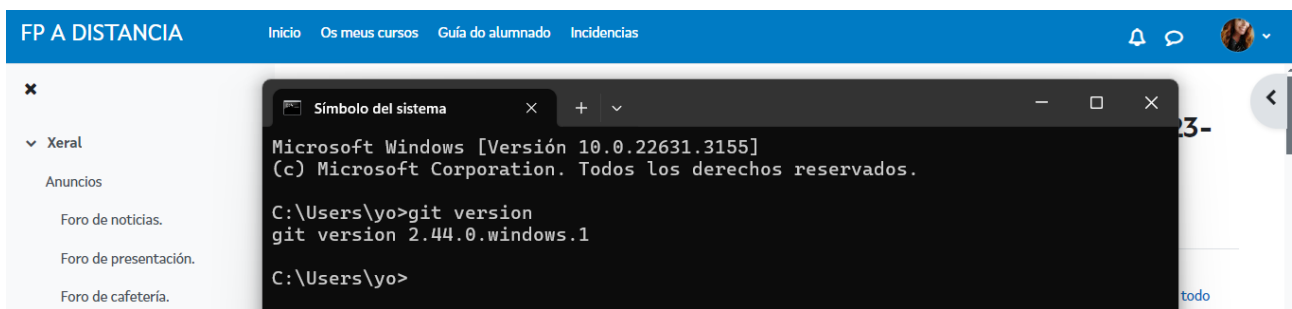


Para realizarlo desde línea de comandos, instalamos Git para nuestro sistema operativo (en mi caso para Windows 64 bit).

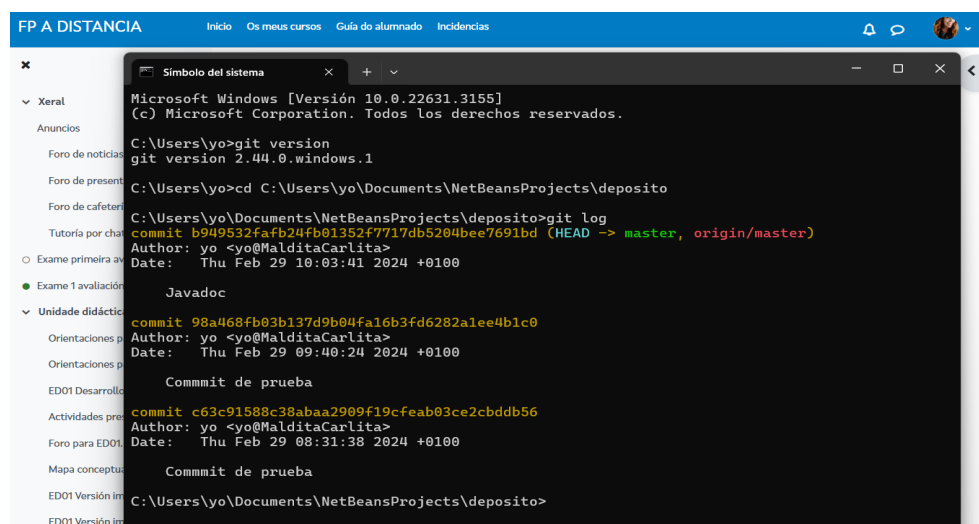




Una vez instalado, abrimos la terminal de comandos y comprobamos que este correctamente instalado mediante *git version*



Nos movemos al directorio en el que se encuentra nuestro proyecto mediante el comando *cd + ruta* y ejecutamos *git log* sobre ese proyecto. Por defecto, si no pasamos ningún parámetro, *git log* lista las confirmaciones hechas sobre ese repositorio en orden cronológico inverso. Es decir, las confirmaciones más recientes se muestran al principio. Como podemos ver, este comando lista cada confirmación con su suma de comprobación SHA-1, el nombre y dirección de correo del autor, la fecha y el mensaje de confirmación.

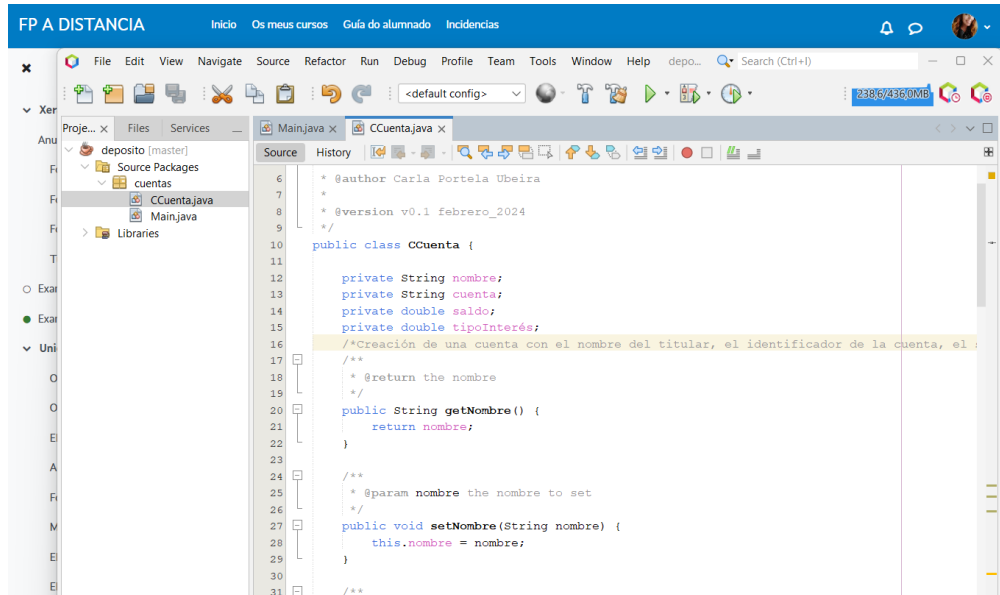


Y ya nos mostraria el historial de versiones.

JAVADOC

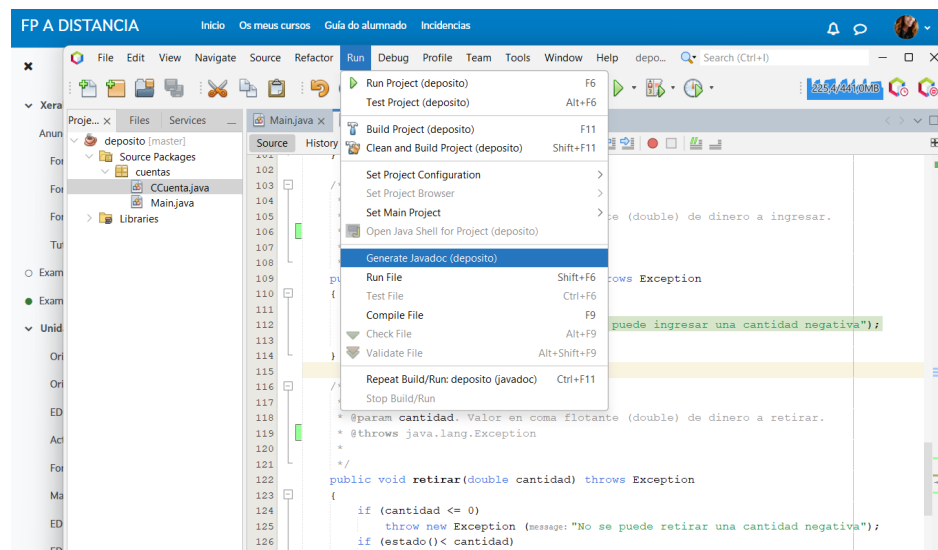
1. Insertar comentarios JavaDoc en la clase Ccuenta.

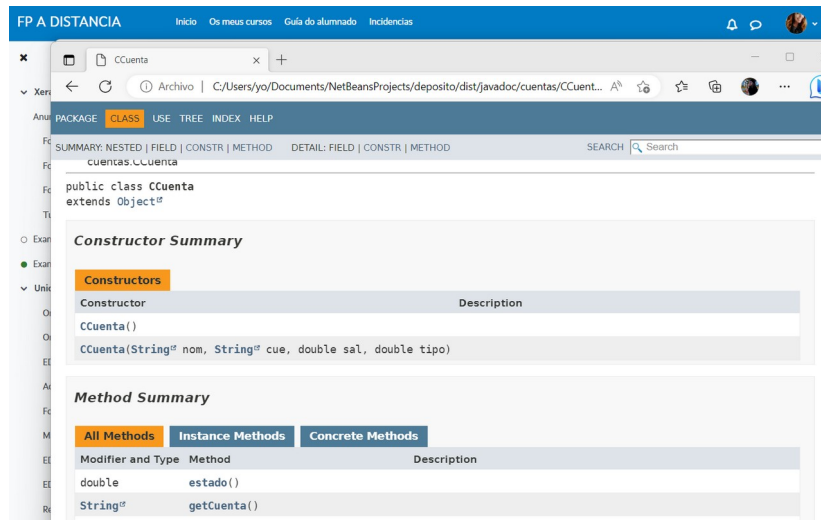
Insertamos los comentarios JavaDoc utilizando `/** */`, al principio de la clase, métodos y variables. Utilizamos etiquetas como `@autor` (nombre del autor), `@version` (número de versión), `@param` (descripción de los parámetros de los métodos) y `@return` (descripción del valor devuelto por el método).



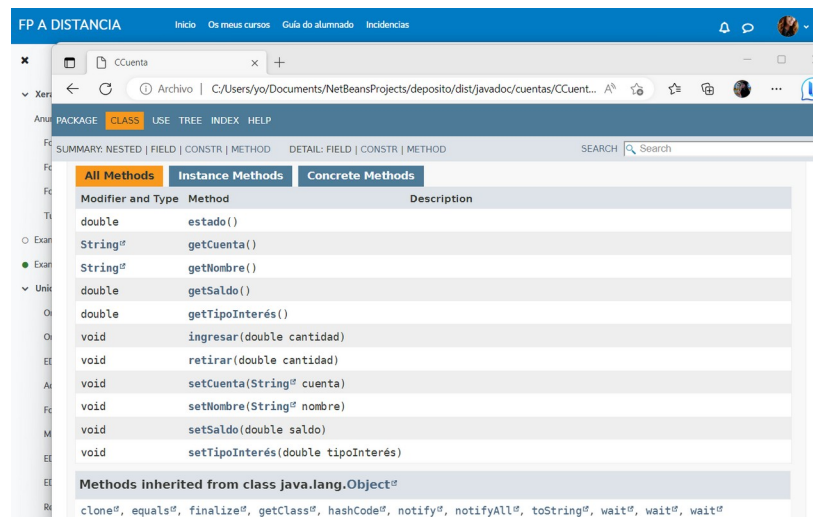
2. Generar documentación JavaDoc para todo el proyecto y comprueba que abarca todos los métodos y atributos de la clase Ccuenta.

Para ello seleccionamos en NetBeans, la pestaña Run → Generate Javadoc





Y nos lo muestra en HTML.



Mediante Control +P lo guardamos en formato pdf.

Añadimos los ficheros con cambios, realizamos un commit y los subimos haciendo un push al repositorio.

El enlace al repositorio es <https://github.com/MalditaCarlita/ED04TAREA.git>

