



ENTORNOS DE DESARROLLO

Optimización y Documentación

Descripción breve

Trabajo en el que refactorizamos un proyecto, lo subimos a GitHub y creamos un JavaDoc.

Eduardo Jimeno Pablo
info@hangar-gym.com

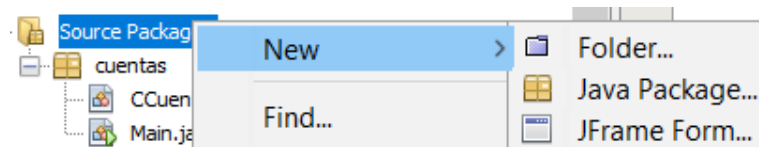
Tarea para ED04.

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.

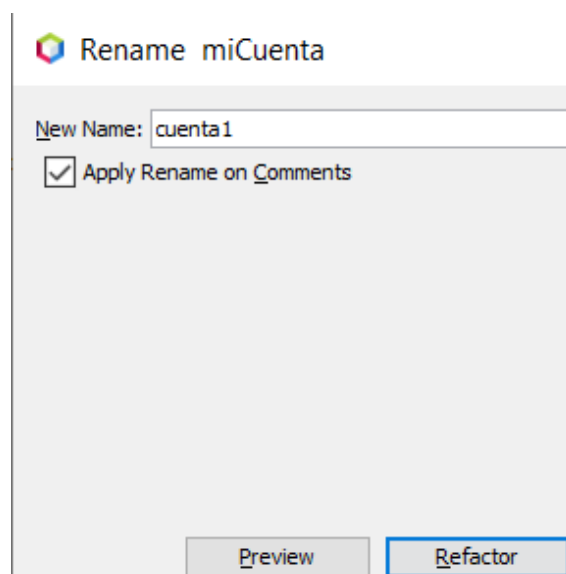
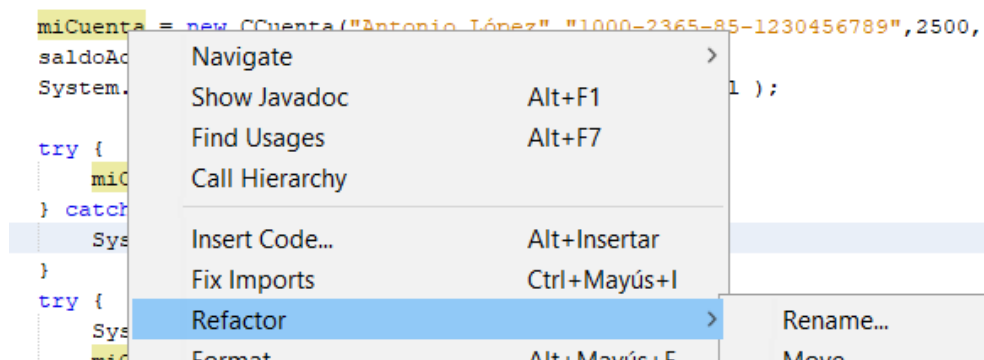
Basándonos en ese proyecto, vamos a realizar las siguientes actividades.

REFACTORIZACIÓN

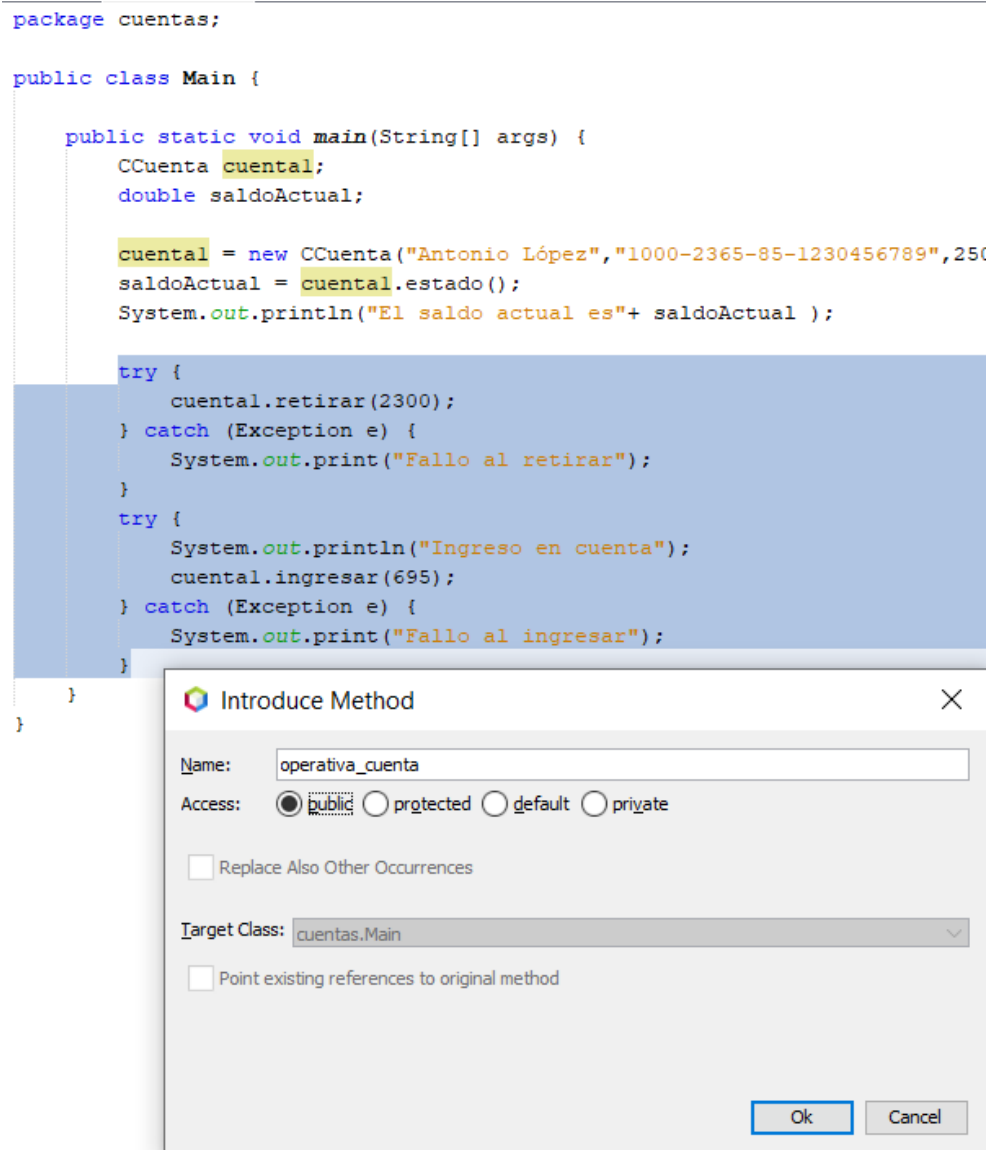
1. Las clases deberán formar parte del paquete cuentas.



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



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



```
package cuentas;

public class Main {

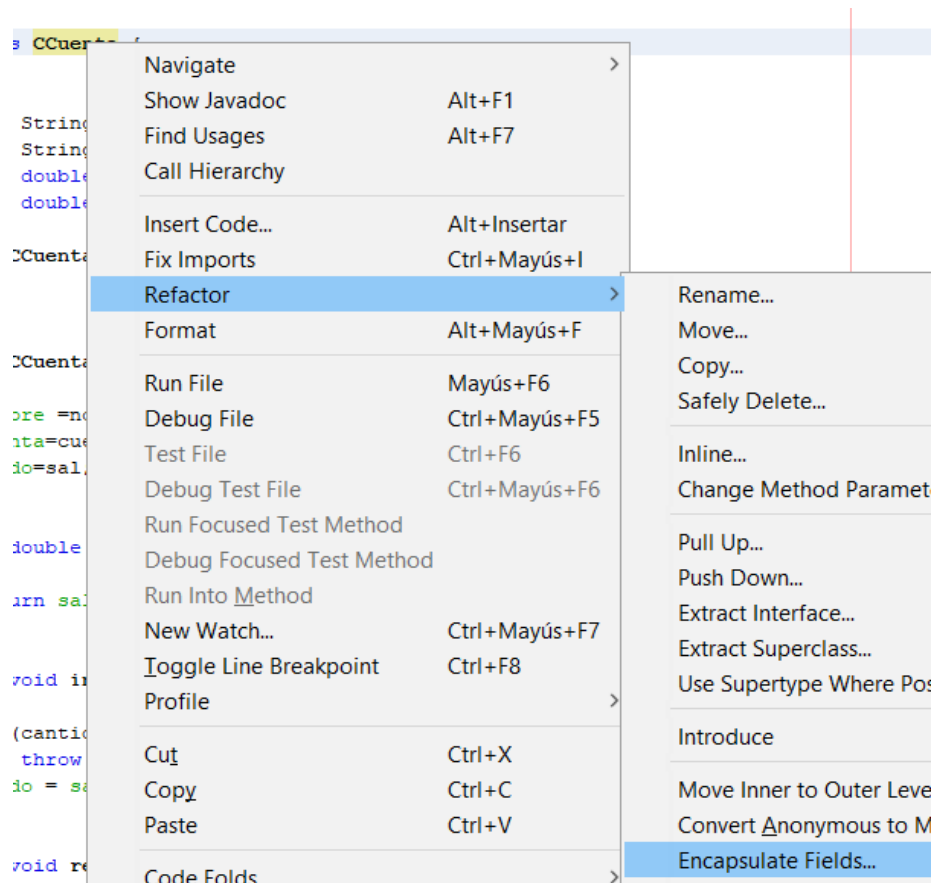
    public static void main(String[] args) {
        CCuenta cuental;
        double saldoActual;

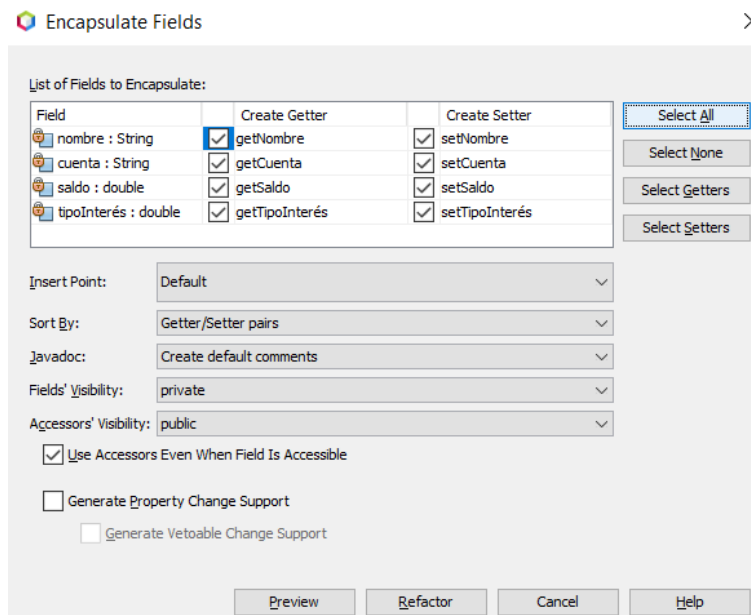
        cuental = new CCuenta("Antonio López","1000-2365-85-123045678";
        saldoActual = cuental.estado();
        System.out.println("El saldo actual es"+ saldoActual );

        operativa_cuenta(cuental);
    }

    public static void operativa_cuenta(CCuenta cuental) {
        try {
            cuental.retirar(2300);
        } catch (Exception e) {
            System.out.print("Fallo al retirar");
        }
        try {
            System.out.println("Ingreso en cuenta");
            cuental.ingresar(695);
        } catch (Exception e) {
            System.out.print("Fallo al ingresar");
        }
    }
}
```

4. Encapsular los atributos de la clase CCuenta.





```
+ /**...3 lines */
- public String getNombre() {
  return nombre;
}

+ /**...3 lines */
- public void setNombre(String nombre) {
  this.nombre = nombre;
}

+ /**...3 lines */
- public String getCuenta() {
  return cuenta;
}

+ /**...3 lines */
- public void setCuenta(String cuenta) {
  this.cuenta = cuenta;
}

+ /**...3 lines */
- public double getSaldo() {
  return saldo;
}

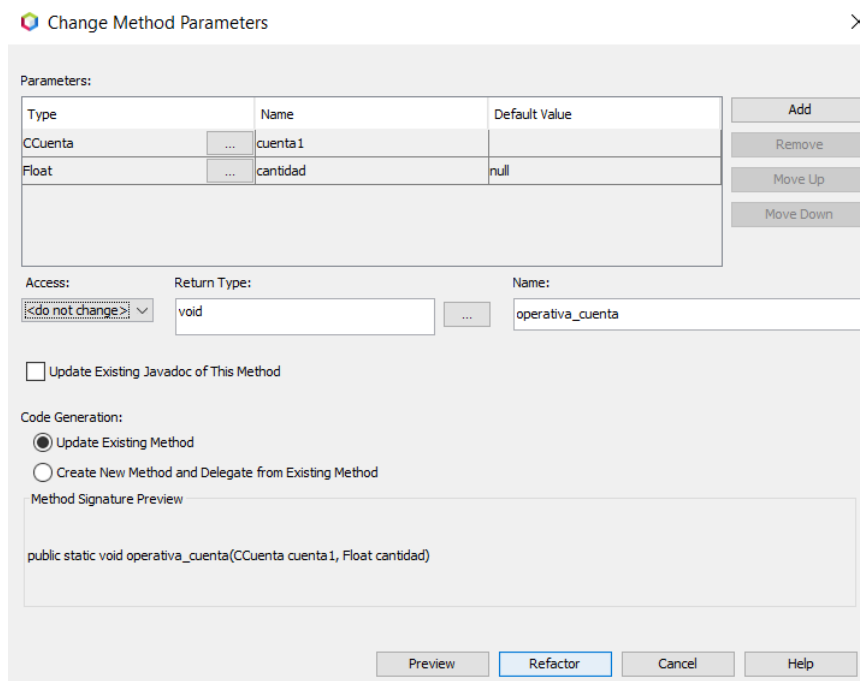
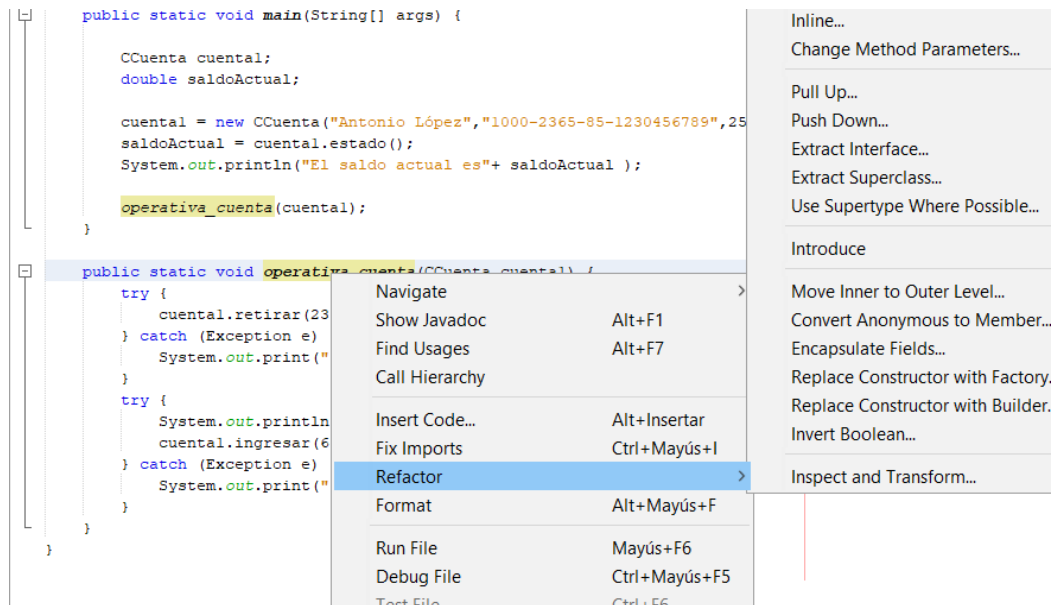
+ /**...3 lines */
- public void setSaldo(double saldo) {
  this.saldo = saldo;
}

+ /**...3 lines */
- public double getTipoInterés() {
  return tipoInterés;
}

+ /**...3 lines */
- public void setTipoInterés(double tipoInterés) {
  this.tipoInterés = tipoInterés;
}

}
```

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

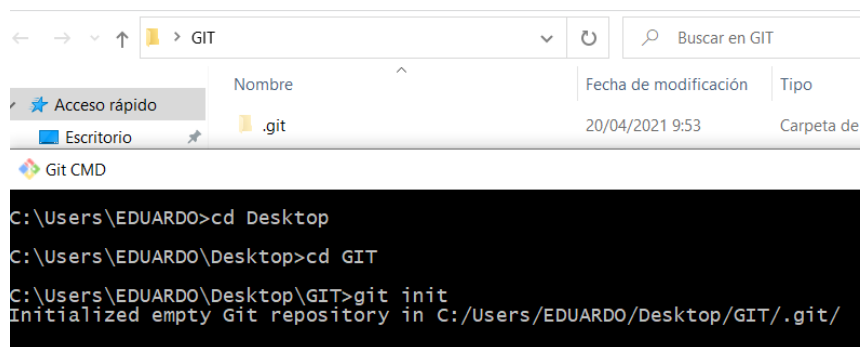


```
public static void operativa_cuenta(CCuenta cuental, Float cantidad) {  
    try {  
        cuental.retirar(2300);  
    } catch (Exception e) {  
        System.out.print("Fallo al retirar");  
    }  
    try {  
        System.out.println("Ingreso en cuenta");  
        cuental.ingresar(695);  
    } catch (Exception e) {  
        System.out.print("Fallo al ingresar");  
    }  
}
```

GIT

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

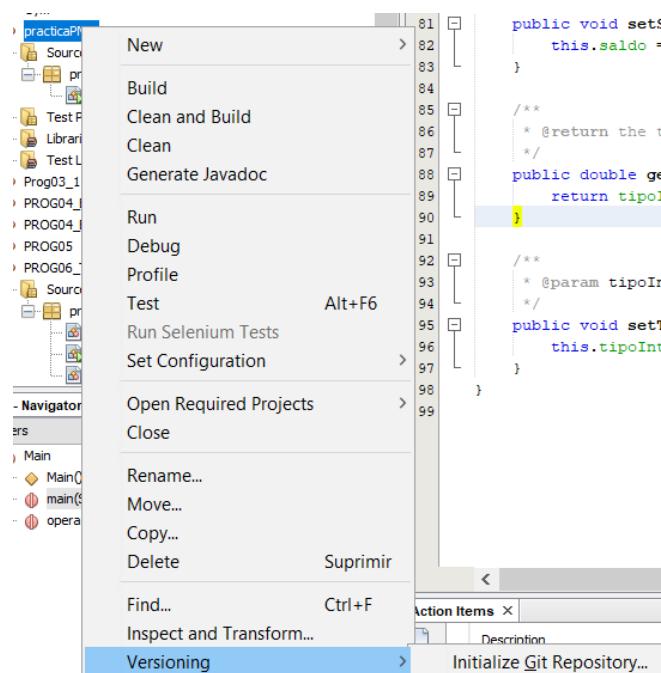
Desde consola podríamos crearlo...



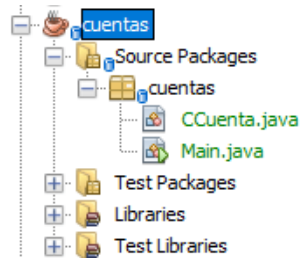
The image shows a file explorer window with the path C:\Users\EDUARDO\Desktop\GIT. It displays a .git folder. Below the explorer is a terminal window with the following commands and output:

```
C:\Users\EDUARDO>cd Desktop  
C:\Users\EDUARDO\Desktop>cd GIT  
C:\Users\EDUARDO\Desktop\GIT>git init  
Initialized empty Git repository in C:/Users/EDUARDO/Desktop/GIT/.git/
```

En NetBeans > Versioning > Initialize Git Repository...



Vemos como se ha creado el GIT



Nos iniciamos en GitHub con usuario y contraseña y creamos Repositorio

Create a new repository

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

Owner * wtorioja2 / Repository name * Tarea4Cuentas ✓

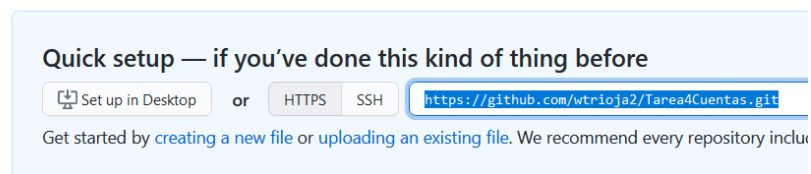
Great repository names Tarea4Cuentas is available. Need inspiration? How about...

Description (optional)

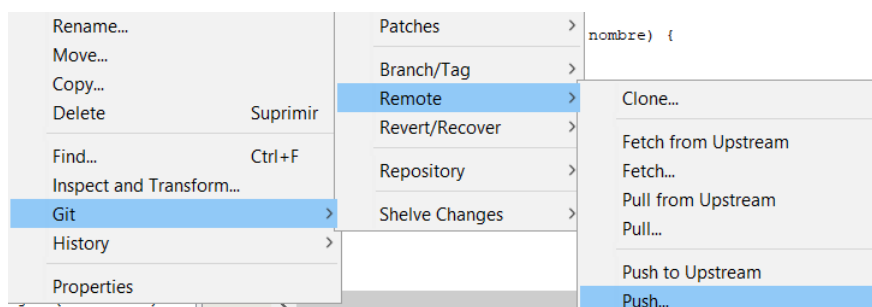
☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

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

Copiamos la URL que necesitaremos en Netbeans



En NetBeans > Git > Remote > Push...



Pegamos la URL e introducimos usuario y contraseña

Push to Remote Repository

Steps

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

Remote Repository

☐ Select Configured Git Repository Location:

☒ Specify Git Repository Location:

Remote Name: origin ☒ Persist Remote

Repository URL: https://github.com/wtrioja2/Tarea4Cuentas.git
http[s]://host.xz[:port]/path/to/repo.git/

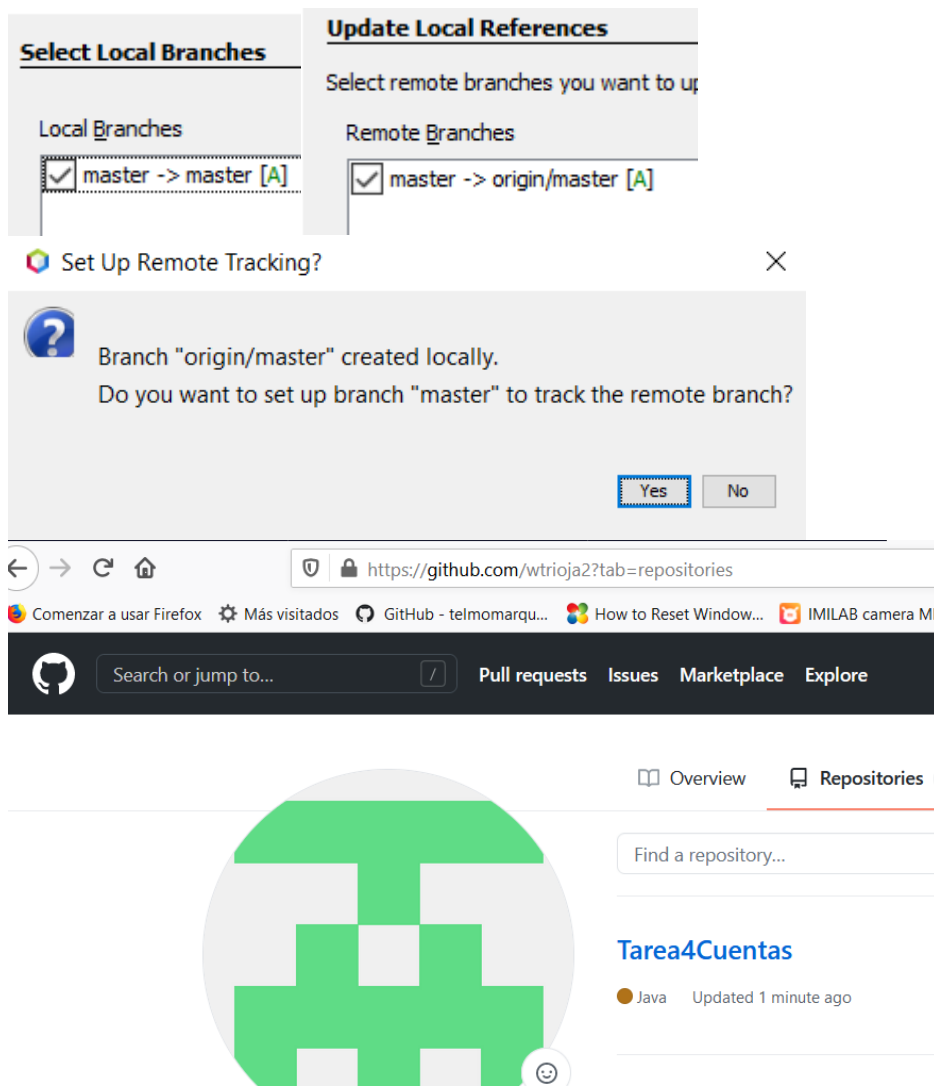
User: wtrioja2 (leave blank for anonymous access)

Password: ☒ Save Password

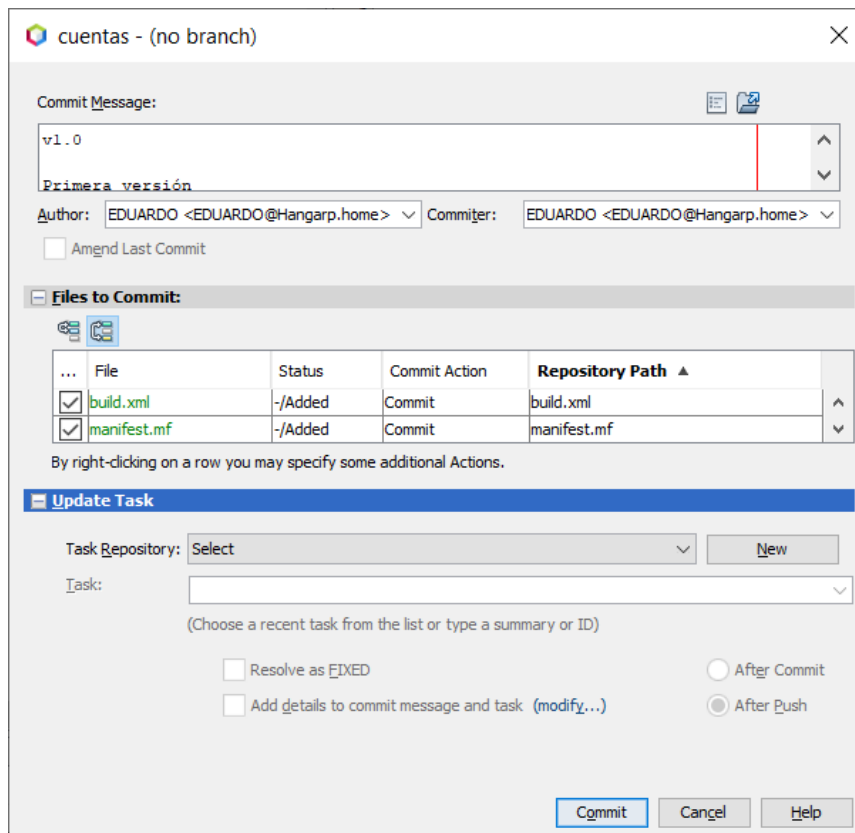
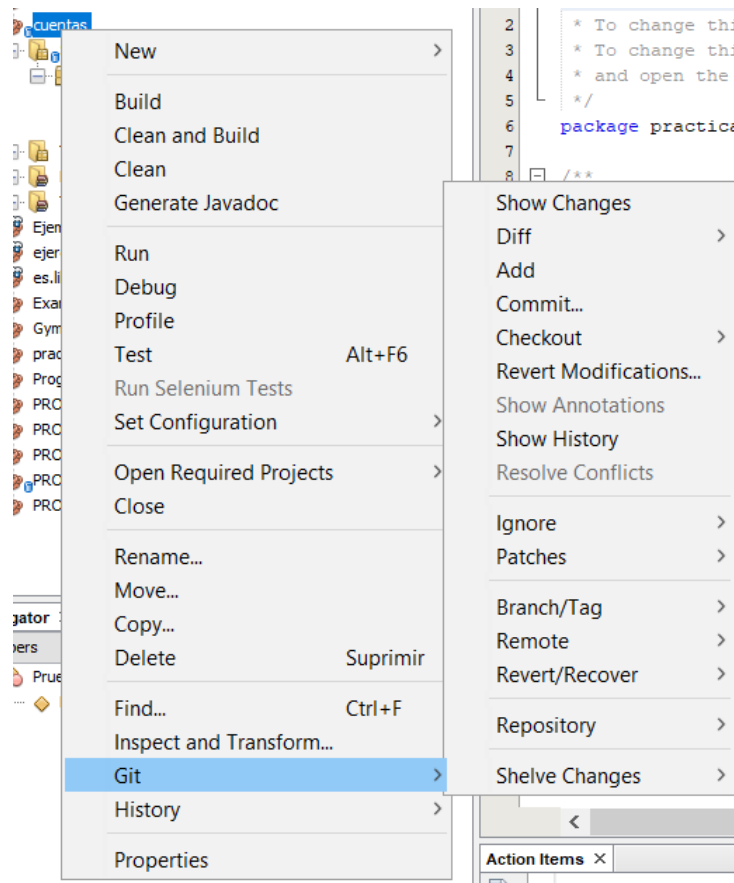
Proxy Configuration...

< Back Next > Finish Cancel Help

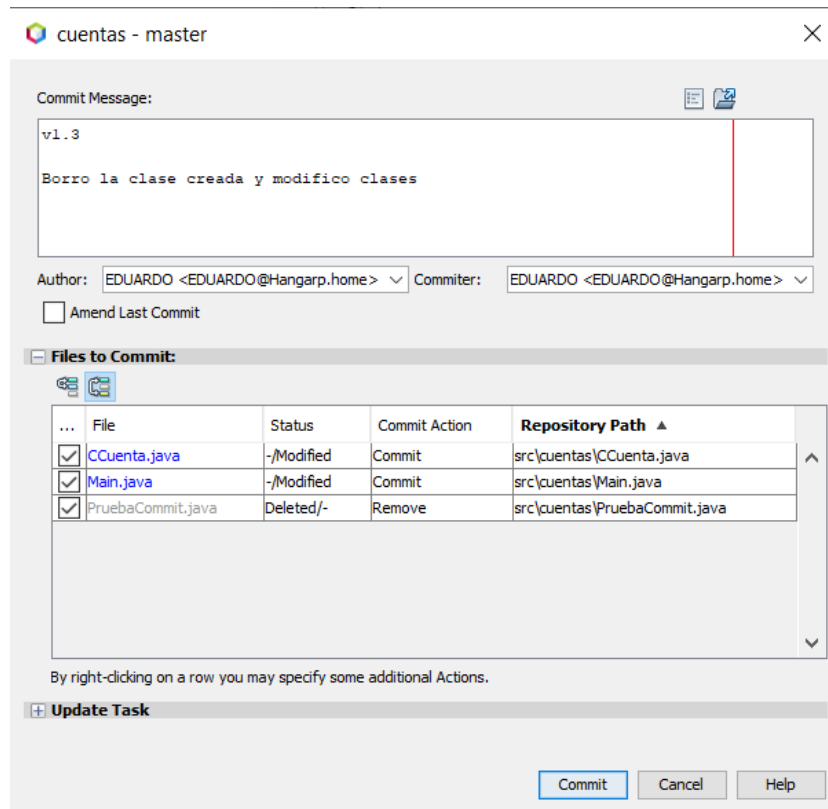
y



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



En un segundo commit simplemente creo una clase que borraré en el tercero commit...



Podemos ver como las nuevas creaciones aparecen en color verde (Added), los cambios en color azul (Modified) y cuando borramos aparece en color gris (Deleted).

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

```
C:\Users\EDUARDO\Documents\NetBeansProjects\cuentas>git log
commit 7fdb3fb4ff4365d825d0a80c9b7a7e57d83f8f1d (HEAD -> master)
Author: EDUARDO <EDUARDO@Hangarp.home>
Date: Tue Apr 20 11:31:15 2021 +0200

    v1.3

    Borro la clase creada y modifico clases

commit 7da7884d7cedf6cbdc1d7b400659edb744d8c46c
Author: EDUARDO <EDUARDO@Hangarp.home>
Date: Tue Apr 20 11:27:51 2021 +0200

    v1.1

    Añado una clase al proyecto

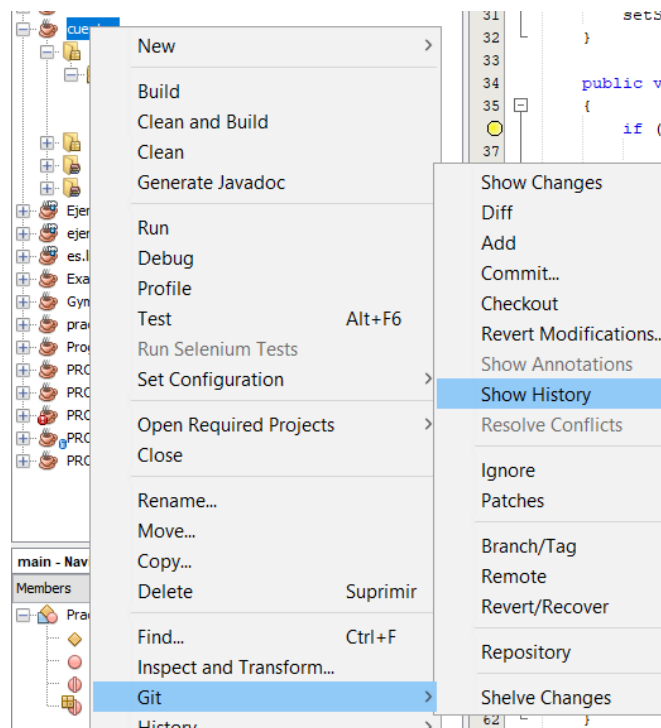
commit 2392b22612842dbc70d122fd02d1d58f1c2b4797
Author: EDUARDO <EDUARDO@Hangarp.home>
Date: Tue Apr 20 11:24:15 2021 +0200

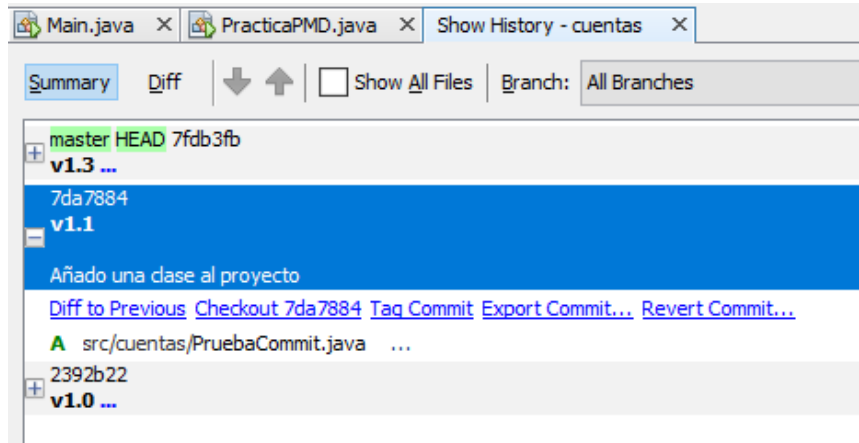
    v1.0

    Primera versión

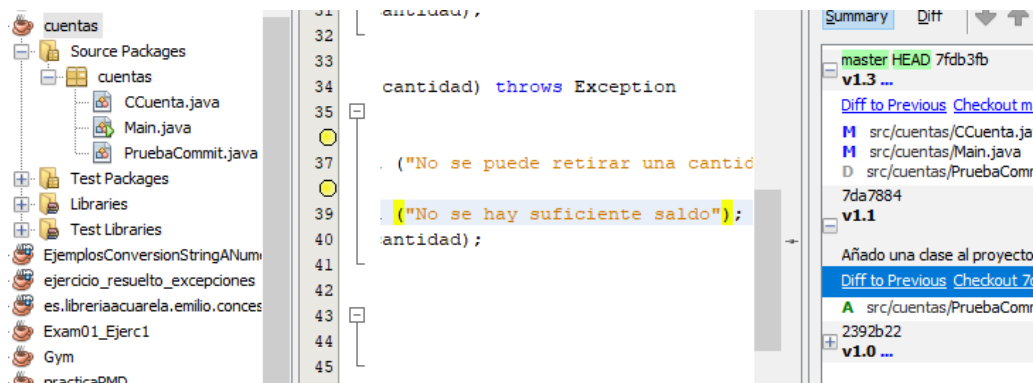
C:\Users\EDUARDO\Documents\NetBeansProjects\cuentas>
```

También podemos verlo en NetBeans en...

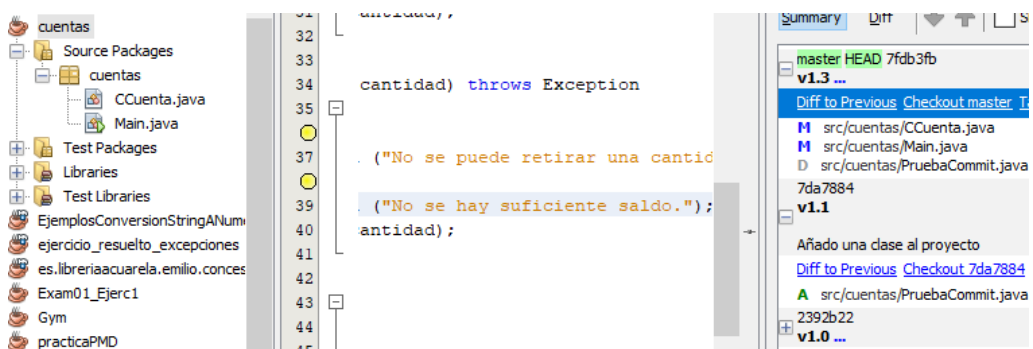




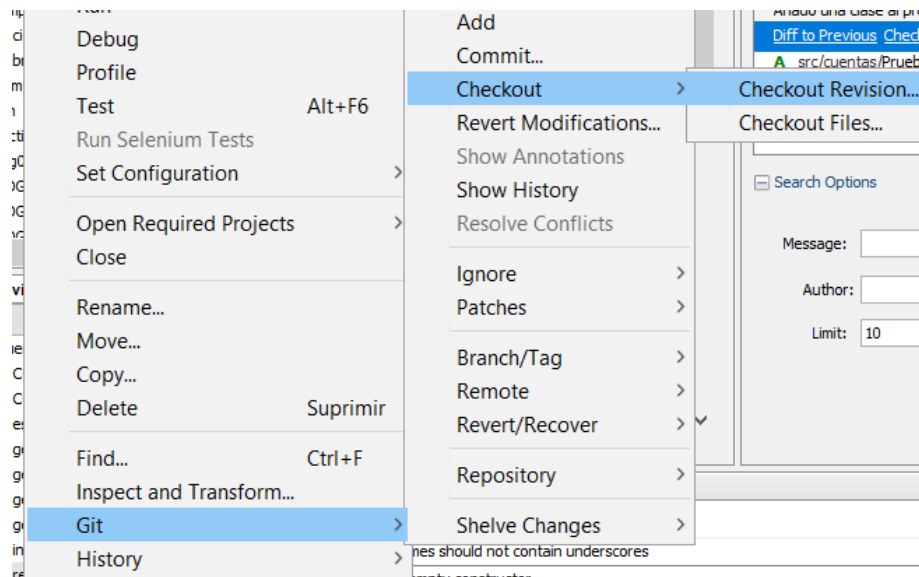
Pulsamos Checkout sobre la versión 1.1 y vemos como aparece PruebaCommit y desaparecen las modificaciones de la versión 1.3



Volvemos a la última versión con Chekout master



También podemos ver los cambios en las diferentes versión en:



JAVADOC

1. Insertar comentarios JavaDoc en la clase CCuenta.

```
CCuenta.java x
1  package cuentas;
2
3  /**
4   * Clase que gener las posibles acciones de una cuenta
5   *
6   * @author EDUARDO
7   * @version 1.3
8   *
9   */
10 public class CCuenta {
11
12     /**
13     * Declaración de las variables (nombre, cuenta, saldo y tipo de interés).
14     */
15     private String nombre;
16     private String cuenta;
17     private double saldo;
18     private double tipoInterés;
19
20     public CCuenta()
21     {
22     }
23
24     /**
25     * Constructor de la clase con parámetros (encima uno vacío).
26     * @param nom nombre de la cuenta
27     * @param cue cuenta
28     * @param sal saldo de la cuenta
29     * @param tipo tipo de interes
30     */
31     public CCuenta(String nom, String cue, double sal, double tipo)
32     {
33         nombre =nom;
34         cuenta=cue;
35         saldo=sal;
36     }
37
38     /**
39     * Método que nos muestra el saldo de la cuenta
40     * @return Saldo de la cuenta
41     */
42     public double estado()
43     {
44         return getSaldo();
45     }
46 }
```

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

The screenshot shows the NetBeans IDE interface. A context menu is open over a project, with the 'Generate Javadoc' option highlighted. Below the menu, the Javadoc documentation for the 'CCuenta' class is displayed. The documentation includes the class hierarchy, a description, a constructor summary, and a method summary table.

Class CCuenta

java.lang.Object
cuentas.CCuenta

public class **CCuenta**
extends java.lang.Object

Clase que genera las posibles acciones de una cuenta

Constructor Summary

Constructors

Constructor and Description

CCuenta ()

CCuenta(java.lang.String nom, java.lang.String cue, double sal, double tipo)
Constructor de la clase con parámetros (encima uno vacío).

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method and Description	
double	estado () Método que nos muestra el saldo de la cuenta	
java.lang.String	getCuenta ()	
java.lang.String	getNombre ()	
double	getSaldo ()	
double	getTipoInterés ()	