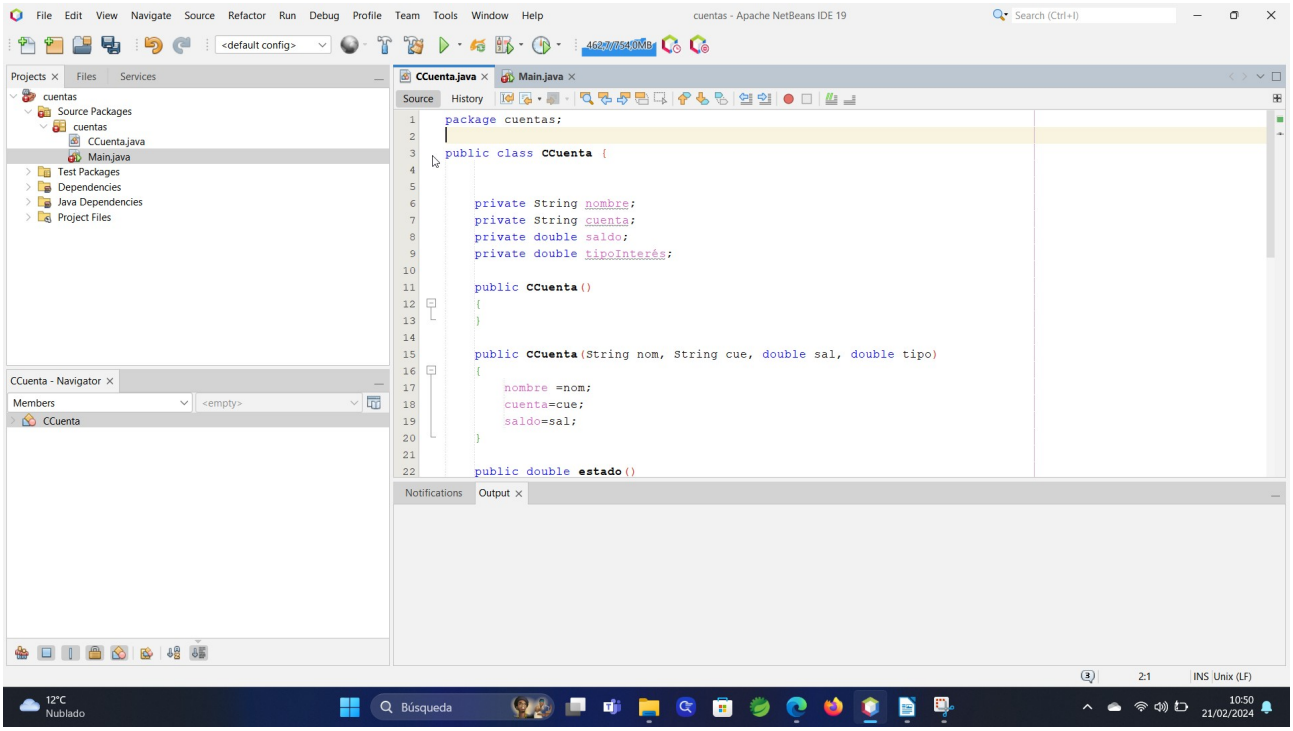
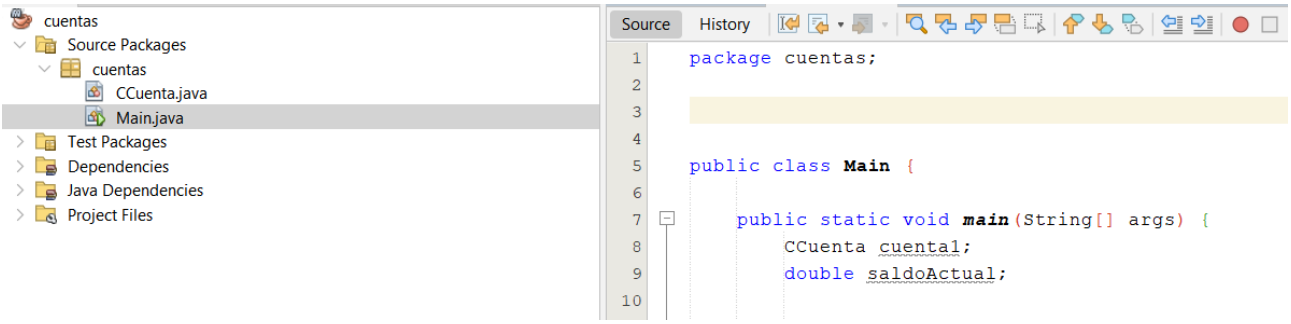
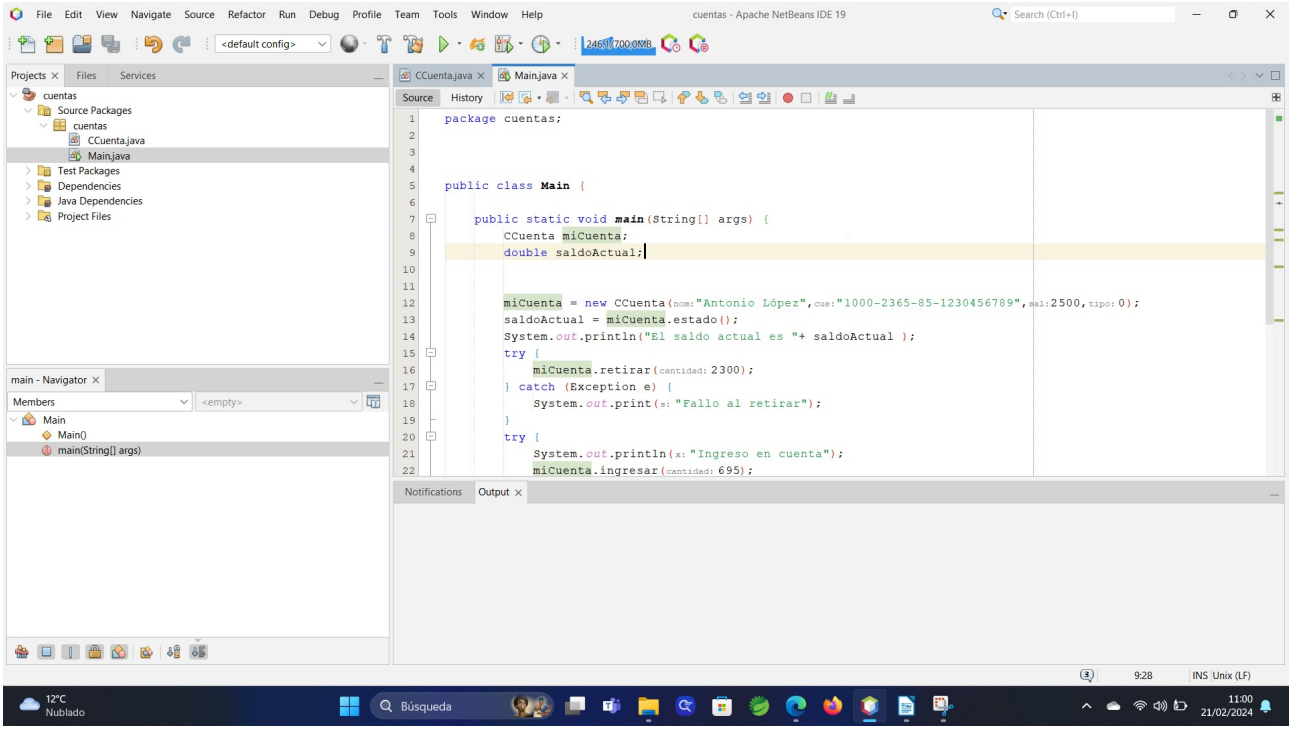
Tarefa EU04
Refactorización

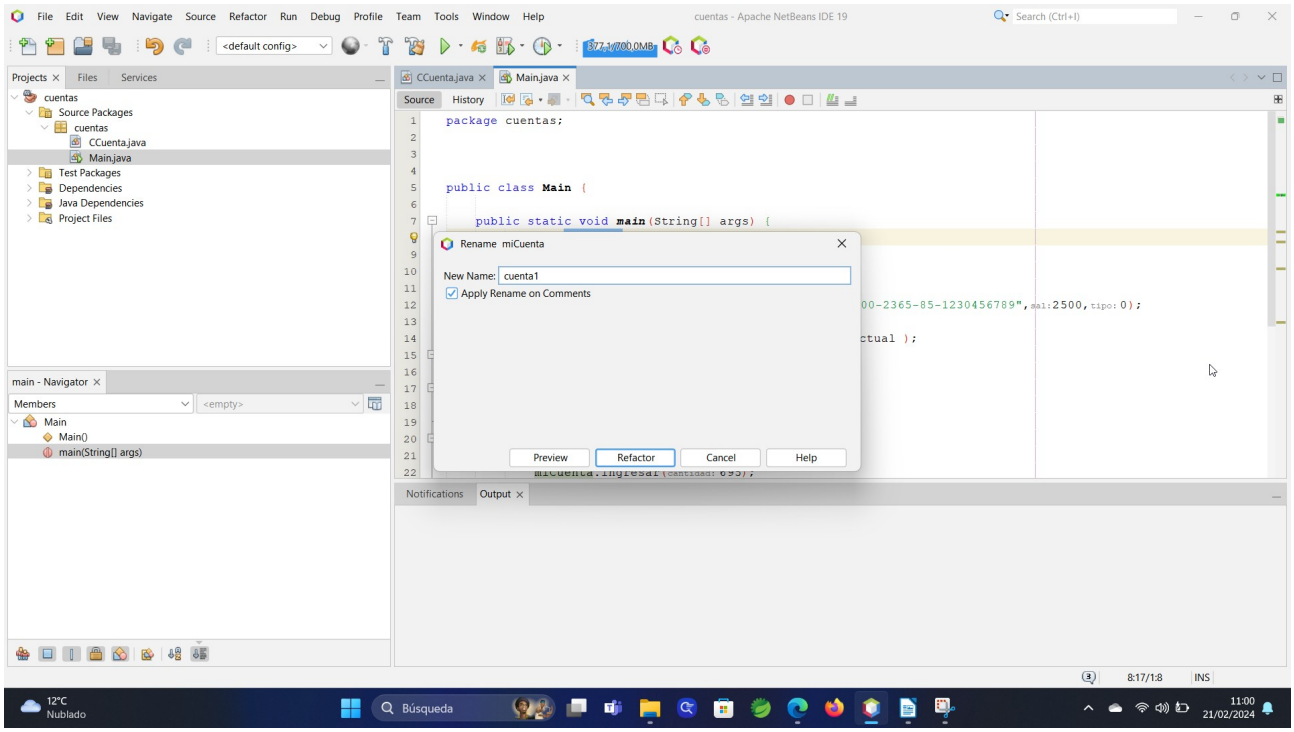
1.-Las clases deberán formar parte del paquete cuentas



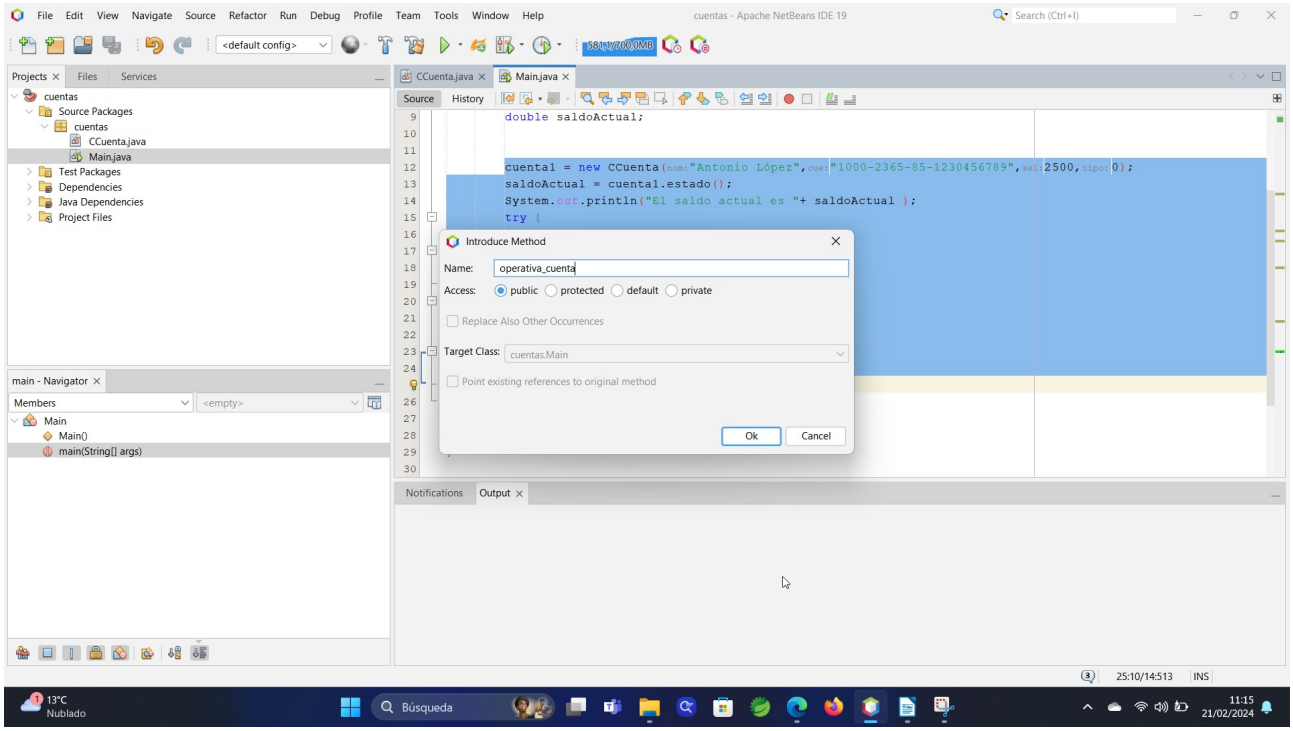


2-. Cambiar el nombre de la variable "miCuenta" por "cuenta1".

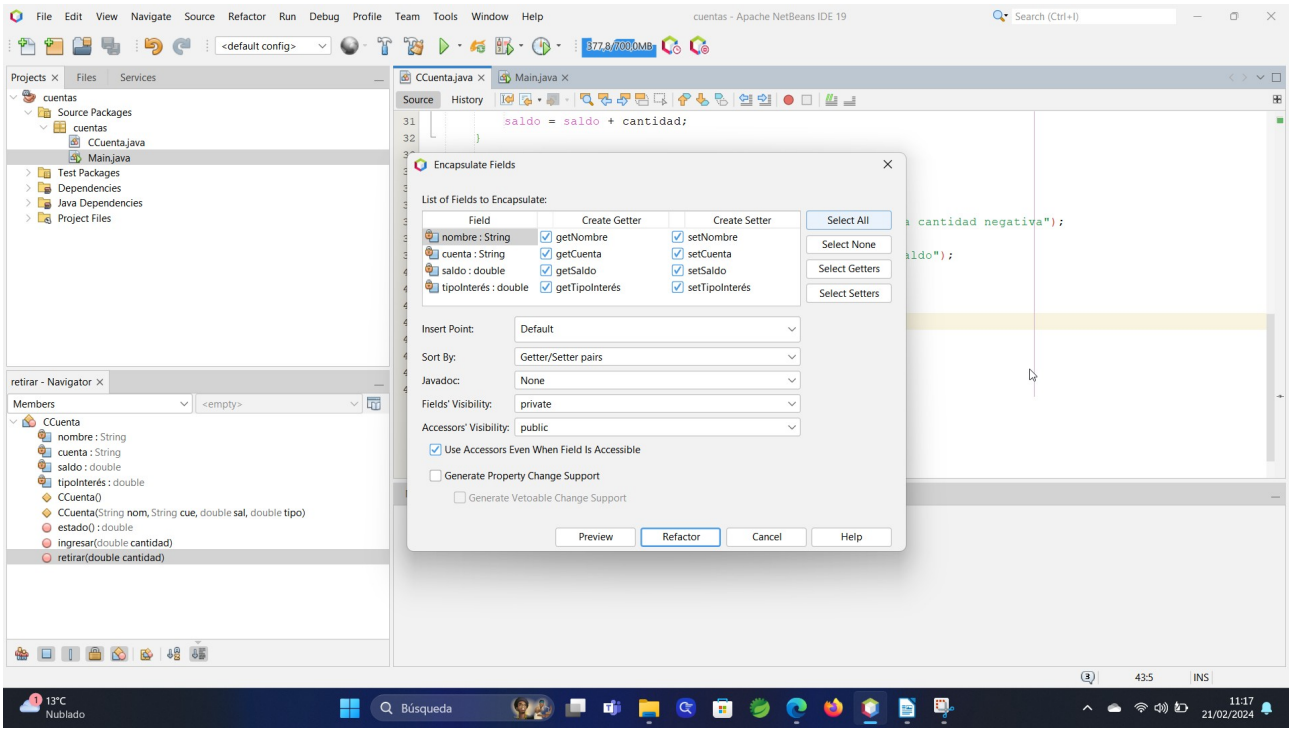




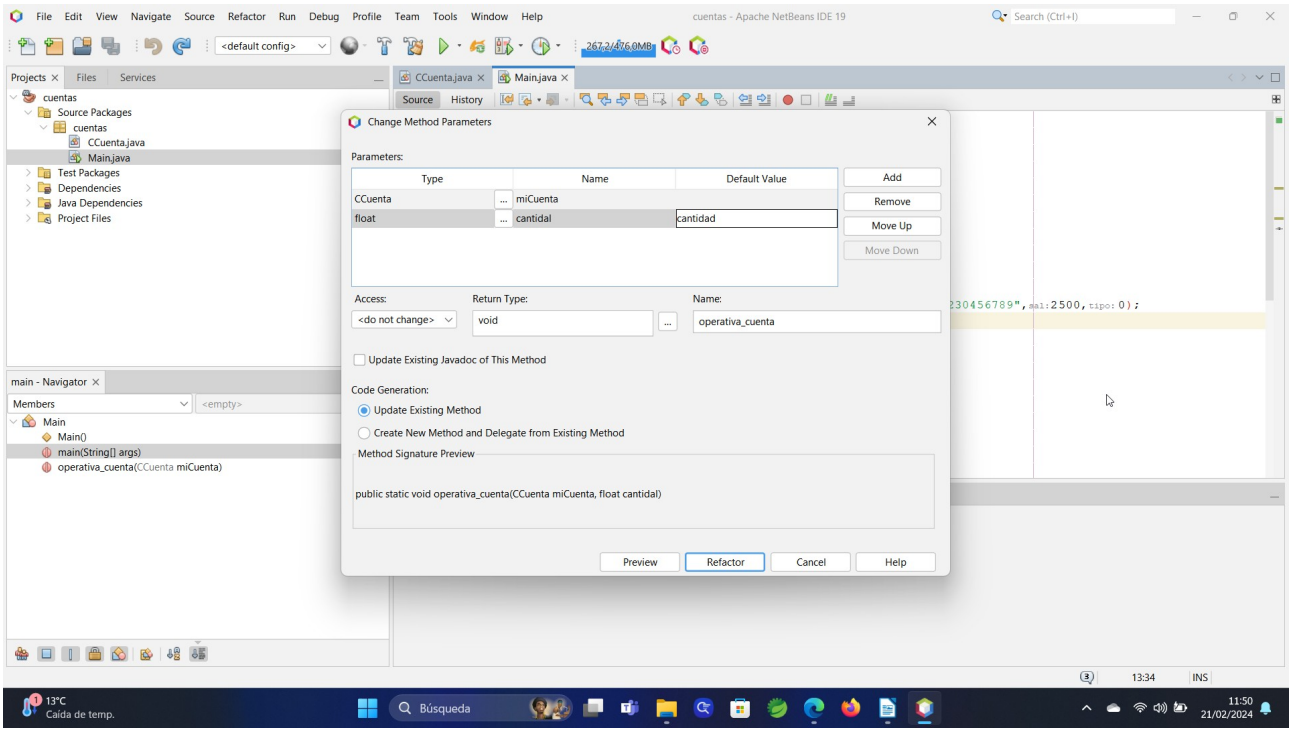
3.-. Introducir el método operativa_cuenta, que englobe las sentencias de la clase Main que operan con el objeto cuenta1.



4.-Encapsular los atributos de la clase CCuenta

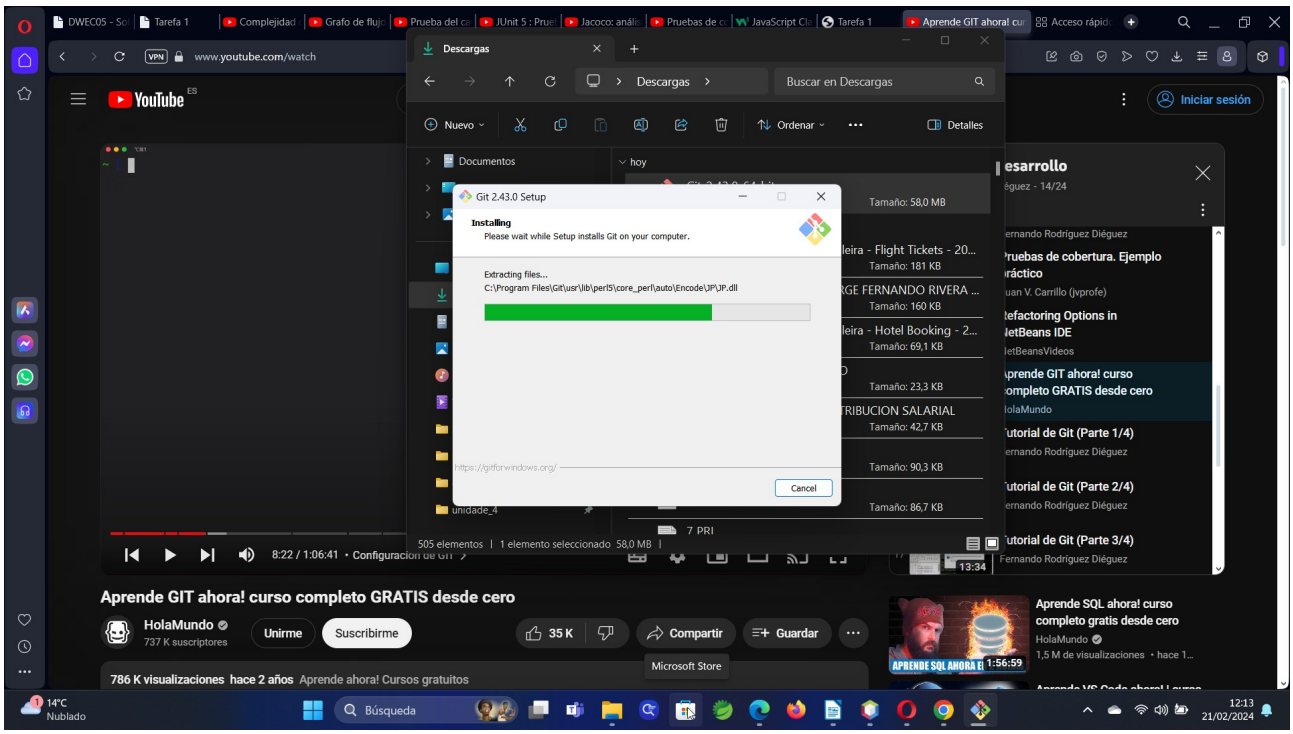


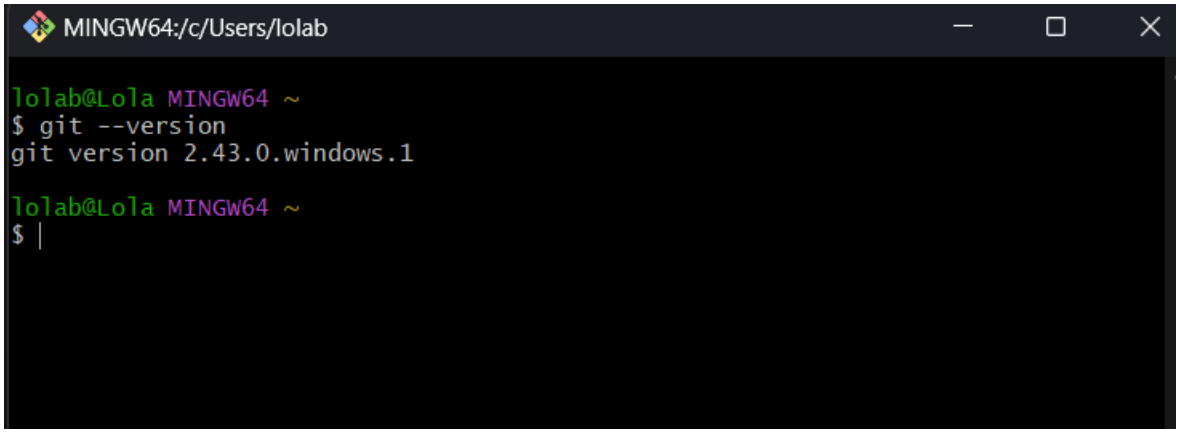
5.-Añadir un nuevo parámetro al método operativa_cuenta, de nombre cantidad y de tipo float

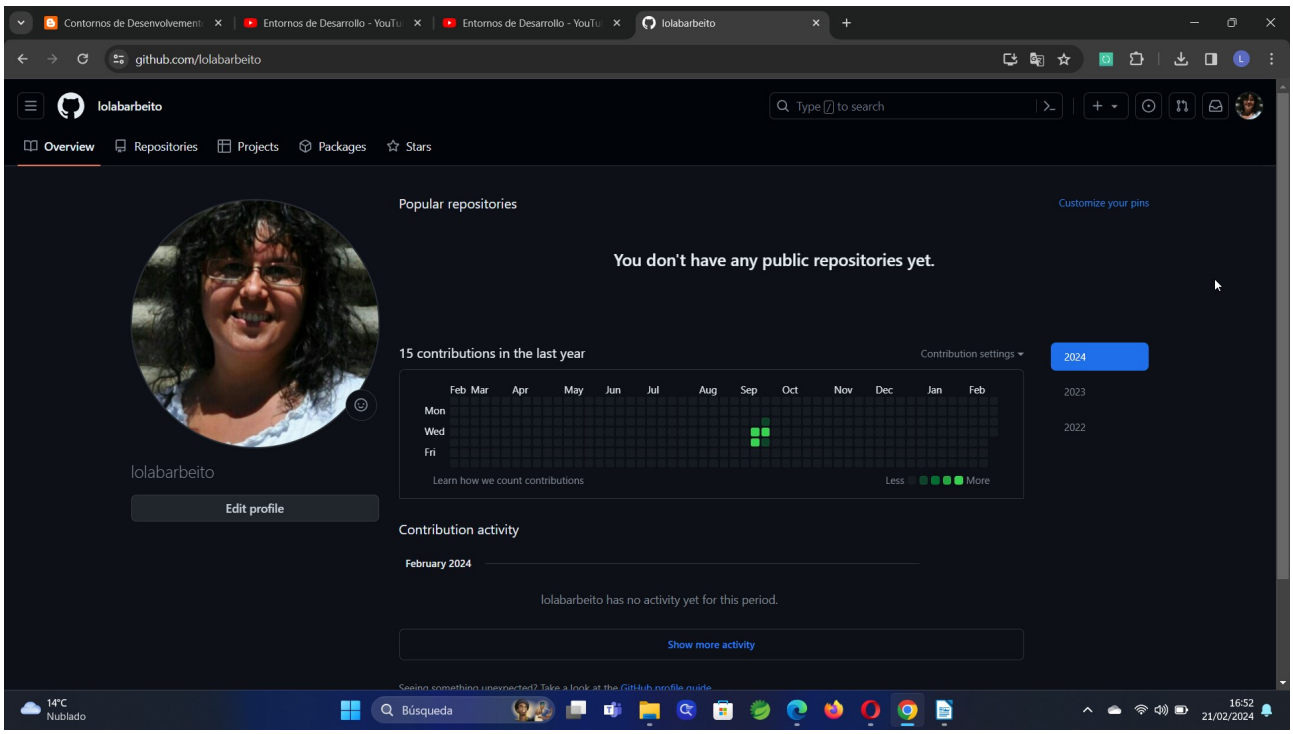


GIT

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


Descargas
Buscar en Descargas
Nuevo
Ordenar
Detalles
Documentos
hoy
Git 2.43.0 Setup
Installing
Please wait while Setup installs Git on your computer.
Extracting files...
C:\Program Files\Git\usr\lib\perl5\core_perl\auto\Encode\JP\JP.dll
Cancel
505 elementos | 1 elemento seleccionado 58,0 MB
YouTube
Iniciar sesión
Aprende GIT ahora! curso completo GRATIS desde cero
HolaMundo
737 K suscriptores
Unirme
Suscribirme
35 K
Compartir
Guardar
786 K visualizaciones hace 2 años Aprende ahora! Cursos gratuitos
Aprende SQL ahora! curso completo gratis desde cero
HolaMundo
1,5 M de visualizaciones
Microsoft Store
14°C
Nublado
Búsqueda
12:13
21/02/2024


MINGW64:/c/Users/lolab
lolab@Lola MINGW64 ~
$ git --version
git version 2.43.0.windows.1
lolab@Lola MINGW64 ~
$



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

Engadimos unha variable (private double variable) no documento Ccuenta.java, engadimos o cambio e logo facemos un commit e chamámolo “Engadir variable”.

```
lolab@Lola MINGW64 ~/OneDrive/Escritorio/cuentas/src/main/java/cuentas (master)
$ git add CCuenta.java

lolab@Lola MINGW64 ~/OneDrive/Escritorio/cuentas/src/main/java/cuentas (master)
$ git commit -m "Engadir variable"
[master (root-commit) 5d9d2ca] Engadir variable
 3 files changed, 122 insertions(+)
 create mode 100644 pom.xml
 create mode 100644 src/main/java/cuentas/CCuenta.java
 create mode 100644 src/main/java/cuentas/Main.java
```

```
public class CCuenta {

    private String nombre;
    private String cuenta;
    private double saldo;
    private double tipoInterés;
    private double variable;
```

```
lolab@Lola MINGW64 ~/OneDrive/Escritorio/cuentas/src/main/java/cuentas (master)
$ git commit -m "Borrar ăriable"
[master bbdb738] Borrar ăriable
 1 file changed, 1 insertion(+), 2 deletions(-)

lolab@Lola MINGW64 ~/OneDrive/Escritorio/cuentas/src/main/java/cuentas (master)
$
```

```
public class CCuenta {


    private String nombre;
    private String cuenta;
    private double saldo;
    private double tipoInterés;

    public CCuenta()
```

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

```
lolab@Lola MINGW64 ~/OneDrive/Escritorio/cuentas/src/main/java/cuentas (master)
$ git log
commit bbdb73861e4d2a52c54a39462a265b98d6ea89ff (HEAD -> master)
Author: Lola <lolabarbeito@gmail.com>
Date:   Fri Feb 23 11:56:40 2024 +0100

    Borrar ăriable

commit 513437ca881341b2fd3736db3e5c82d23d5db870
Author: Lola <lolabarbeito@gmail.com>
Date:   Fri Feb 23 11:49:24 2024 +0100

    Engadir variable

commit 5d9d2cae1c7bd2b567d49426aa18fde6e6d69ea5
Author: Lola <lolabarbeito@gmail.com>
Date:   Fri Feb 23 10:58:20 2024 +0100

    Engadir variable

lolab@Lola MINGW64 ~/OneDrive/Escritorio/cuentas/src/main/java/cuentas (master)
$
```

JAVADOC

1.-. Insertar comentarios JavaDoc en la clase CCuenta.

```
package cuentas;
/**
 * Clase CCuenta variables:
 * private String nombre;
 * private String cuenta;
 * private double saldo;
 * private double tipoInterés;
 * @author lolab
```

```
 *
 */
public class CCuenta {

    private String nombre;
    private String cuenta;
    private double saldo;
    private double tipoInterés;
    /**
     * Constructor de la clase CCuenta sin parámetros
     */
    public CCuenta()
    {
    }
    /**
     * Constructor de la clase CCuenta con parámatros.
     * @param nom
     * @param cue
     * @param sal
     * @param tipo
     */
    public CCuenta(String nom, String cue, double sal, double tipo)
    {
        nombre =nom;
        cuenta=cue;
        saldo=sal;
    }
    /**
     * Método estado retorna el estado
     * @return getSaldo()
     */
    public double estado()
    {
        return getSaldo();
    }
    /**
     * Método ingresar No se puede ingresar una cantidad negativa y nos suma al saldo la cantidad
     * que ingresemos.
     * @param cantidad
     * @throws Exception
     */
    public void ingresar(double cantidad) throws Exception
    {
        if (cantidad<0)
            throw new Exception("No se puede ingresar una cantidad negativa");
        setSaldo(getSaldo() + cantidad);
    }
    /**
     * Método retirar con dos excepciones, no se puede retirar una cantidad negativa
     * ni tampoco si no hay saldo suficiente,también nos resta del saldo la cantidad que retiremos.
     * @param cantidad
     * @throws Exception
```

```
 */
public void retirar(double cantidad) throws Exception
{
   if (cantidad <= 0)
      throw new Exception ("No se puede retirar una cantidad negativa");
   if (estado()< cantidad)
      throw new Exception ("No se hay suficiente saldo");
   setSaldo(getSaldo() - cantidad);
}
/**
 * Método getNombre retorna el nombre.
 * @return nombre
 */
public String getNombre() {
   return nombre;
}
/**
 * Método setNombre modifica el nombre.
 * @param nombre
 */
public void setNombre(String nombre) {
   this.nombre = nombre;
}
/**
 * Método getCuenta retorna cuenta.
 * @return cuenta
 */
public String getCuenta() {
   return cuenta;
}
/**
 * Método setCuenta modifica cuenta.
 * @param cuenta
 */
public void setCuenta(String cuenta) {
   this.cuenta = cuenta;
}
/**
 * Método getSaldo retorna el saldo.
 * @return saldo
 */
public double getSaldo() {
   return saldo;
}
/**
 * Método setSaldo modifica el saldo.
 * @param saldo
 */
public void setSaldo(double saldo) {
   this.saldo = saldo;
}
/**
```

```
     * Método getTipoInterés retorna el tipo de interés.
     * @return
     */
    public double getTipoInterés() {
        return tipoInterés;
    }
    /**
     * Método setTipoInterés modifica el tipo de interés.
     * @param tipoInterés
     */
    public void setTipoInterés(double tipoInterés) {
        this.tipoInterés = tipoInterés;
    }
}
```

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

cuentas (cuentas 1.0-SNAPSHOT API)