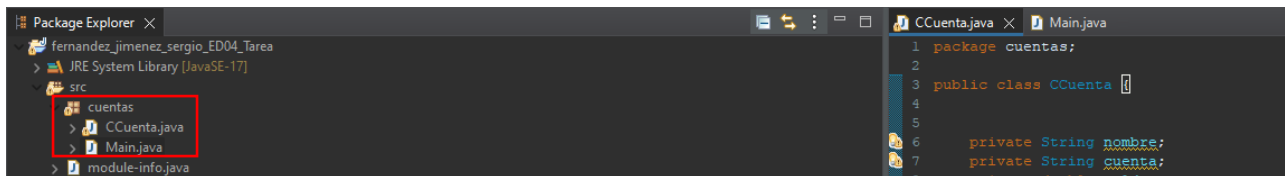


Tarea para ED04.

REFACTORIZACIÓN

1. Las clases deberán formar parte del paquete cuentas.



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

```
public class Main {

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

        cuental = new CCuenta("Antonio López","1000-2365-85-1230456789",2500,0);
        saldoActual = cuental.estado();
        System.out.println("El saldo actual es"+ saldoActual );

        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");
        }
    }

}
```

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

```
cuental = new CCuenta("Antonio López","1000-2365-85-1230456789",2500,0);
operativa_cuenta(cuental);

private static void operativa_cuenta(CCuenta cuental) {
    double saldoActual;
    saldoActual = cuental.estado();
    System.out.println("El saldo actual es: " + saldoActual);
}
```

4. Encapsular los atributos de la clase CCuenta.

```
void setNombre(String nombre) {  
    this.nombre = nombre;  
}  
  
String getNombre(){  
    return nombre;  
}  
  
void setCuenta(String cuenta) {  
    this.cuenta = cuenta;  
}  
  
String getCuenta(){  
    return cuenta;  
}  
  
void setSaldo(double saldo) {  
    this.saldo = saldo;  
}  
  
double getSaldo(){  
    return saldo;  
}  
  
void setTipoInteres(double tipoInteres) {  
    this.tipoInterés = tipoInteres;  
}  
  
double getTipoInteres(){  
    return tipoInterés;  
}  
}
```

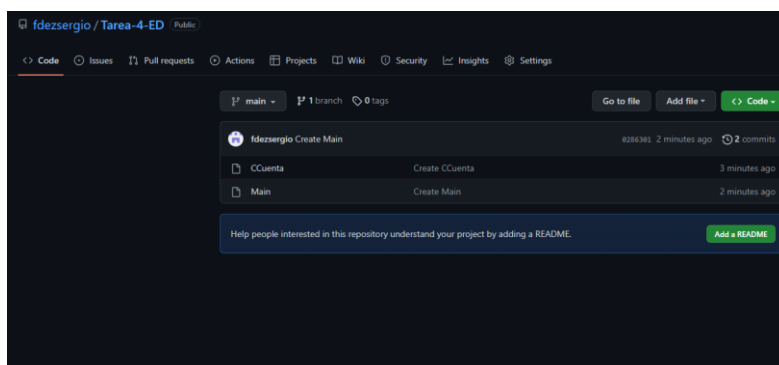
5. Añadir un nuevo parámetro al método operativa_cuenta, de nombre cantidad y de tipo float.

```
private static void operativa_cuenta(CCuenta cuental, float cantidad) {  
    double saldoActual;  
    saldoActual = cuental.estado();  
    System.out.println("El saldo actual es: " + saldoActual);  
}
```

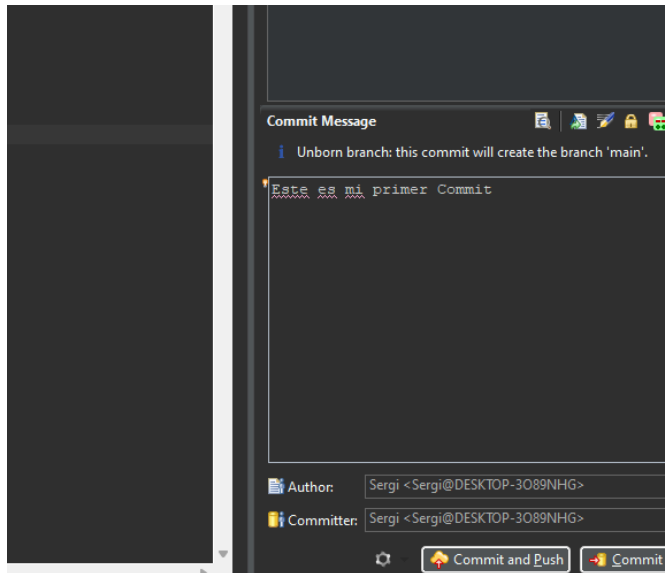
GIT

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

```
C:\Users\Sergi\eclipse-workspace\fernandez_jimenez_sergio_ED04_Tarea>git init  
Initialized empty Git repository in C:/Users/Sergi/eclipse-workspace/fernandez_jimenez_sergio_ED04_Tarea/.git/
```



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



JAVADOC

1. Insertar comentarios JavaDoc en la clase CCuenta.

```
/*
 * Author: Sergi Esta app simula el comportamiento de una cuenta bancaria
 */

package cuentas;

public class CCuenta {

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

    // CONSTRUTOR
    public CCuenta ()
    {
    }

    public CCuenta(String nom, String cue, double sal, double tipo)
    {
        nombre =nom;
        cuenta=cue;
        saldo=sal;
    }

    // METODOS
    public double estado()
    {
        return saldo;
    }
}
```

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

```
public void ingresar(double cantidad) throws Exception
{
    if (cantidad<0)
        throw new Exception("No se puede ingresar una cantidad negativa");
    saldo = saldo + cantidad;
}

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");
    saldo = saldo - cantidad;
}

// SETTERS Y GETTERS
void setNombre(String nombre) {
    this.nombre = nombre;
}

String getNombre(){
    return nombre;
}

void setCuenta(String cuenta) {
    this.cuenta = cuenta;
}
}
```