



IES Augusto González de Linares.

Tarea 4:

Optimización y documentación.

ENTORNOS DE DESARROLLO.

(DAM_ED)

Ciclo formativo

Desarrollo De Aplicaciones Multiplataforma

(DAM)

CURSO:

2021/2022

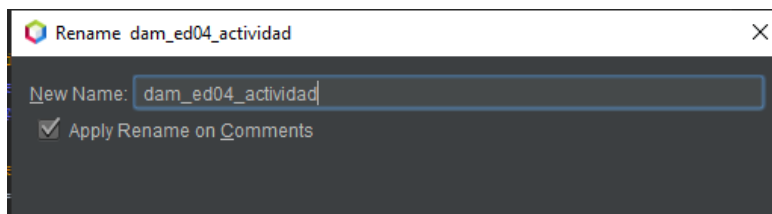
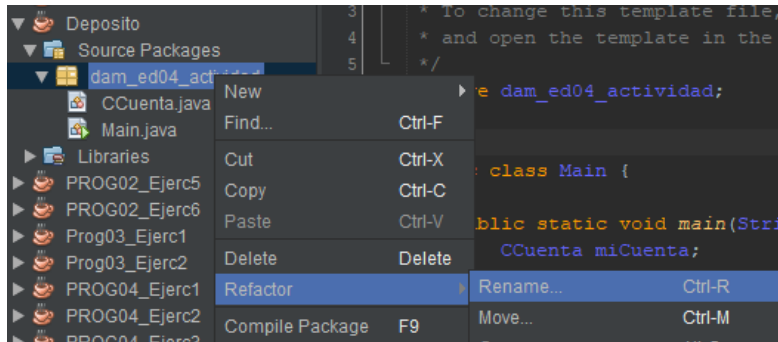
Francisco M. Colls Gutiérrez.

REFACTORIZACIÓN.

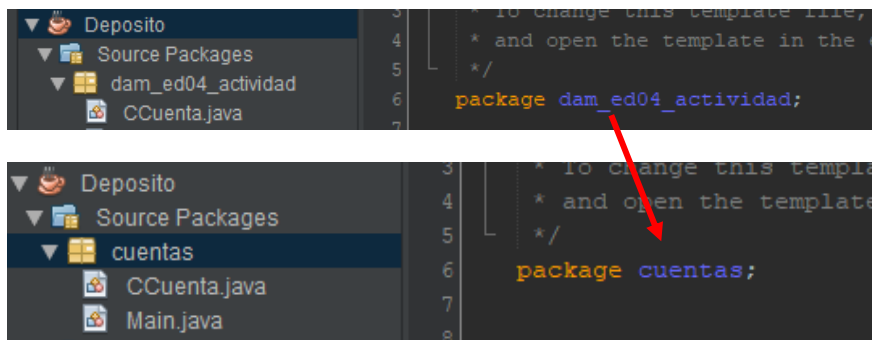
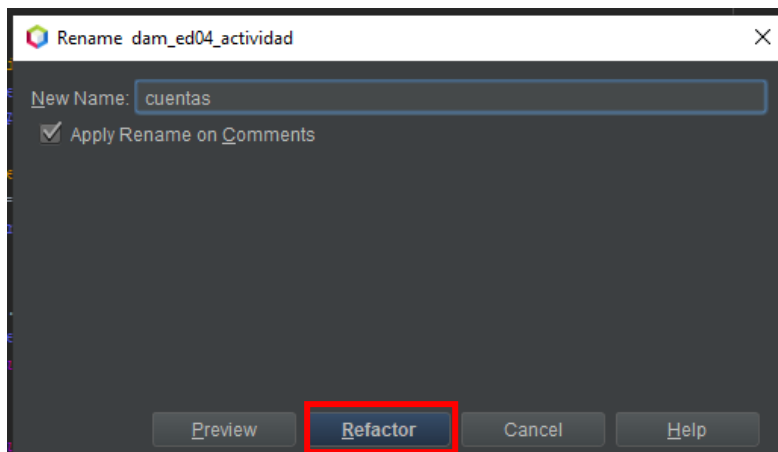
1. Las clases deberán formar parte del paquete cuentas.

Clic derecho encima del nombre del paquete.

Refactor > Rename. Y asignamos el nuevo nombre.



Finalmente hacemos clic en Refactor.



Francisco M. Colls Gutiérrez.

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

Para cambiar el nombre de una variable hacemos clic sobre esa variable y presionamos ctrl+r. Se seleccionan todas las copias de la variable, y lo que tenemos que hacer es borrar el nombre y escribir el nuevo nombre.

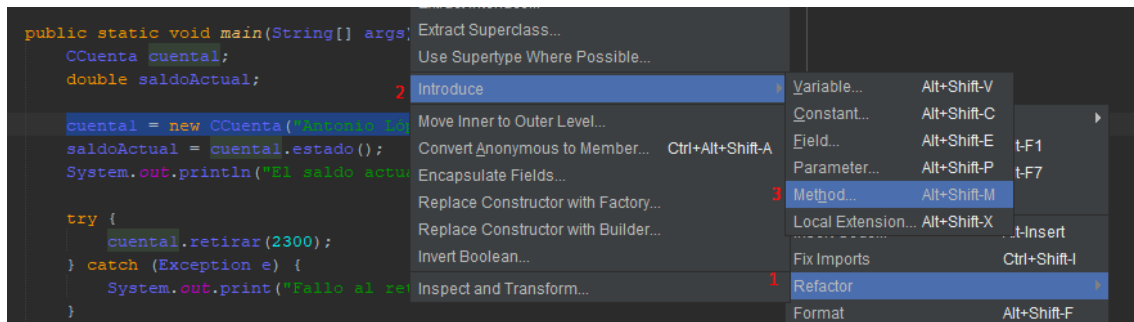
```
public class Main {  
  
    public static void main(String[] args) {  
        CCuenta miCuenta;  
        double saldoActual;  
  
        miCuenta = new CCuenta("Antonio López", "1000-2365-85-123456789", 2500, 0);  
        saldoActual = miCuenta.estado();  
        System.out.println("El saldo actual es"+ saldoActual);  
  
        try {  
            miCuenta.retirar(2300);  
        } catch (Exception e) {  
            System.out.print("Fallo al retirar");  
        }  
  
        try {  
            System.out.println("Ingreso en cuenta");  
            miCuenta.ingresar(695);  
        } catch (Exception e) {  
            System.out.print("Fallo al ingresar");  
        }  
    }  
}
```

```
public static void main(String[] args) {  
    CCuenta cuenta1;  
    double saldoActual;  
  
    cuenta1 = new CCuenta("Antonio López", "1000-2365-85-123456789", 2500, 0);  
    saldoActual = cuenta1.estado();  
    System.out.println("El saldo actual es"+ saldoActual);  
  
    try {  
        cuenta1.retirar(2300);  
    } catch (Exception e) {  
        System.out.print("Fallo al retirar");  
    }  
  
    try {  
        System.out.println("Ingreso en cuenta");  
        cuenta1.ingresar(695);  
    } catch (Exception e) {  
        System.out.print("Fallo al ingresar");  
    }  
}
```

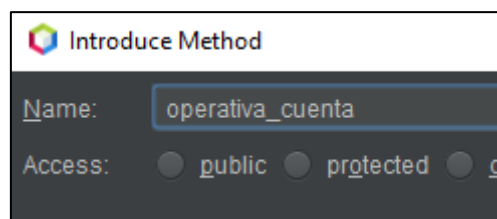
3. Introducir el método operativa_cuenta, que englobe las sentencias de la clase Main que operan con el objeto cuenta1.

Sombreamos la línea donde se encuentra el objeto cuenta1.

Clic derecho > Refactor > Introduce > Method > Nombre del método.



Nombre del método.



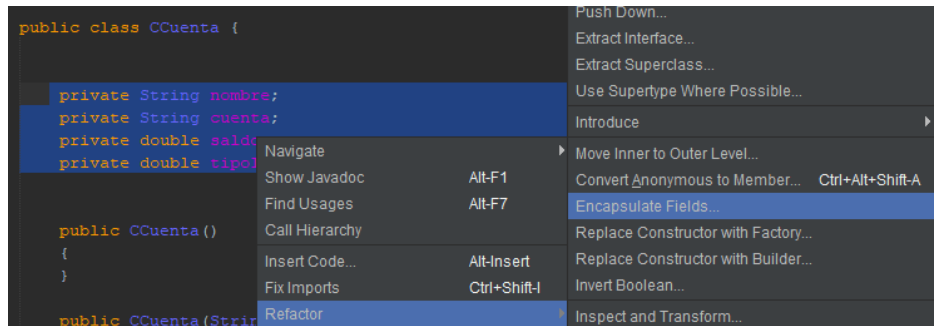
```
private static CCuenta operativa_cuenta() {  
    CCuenta cuenta1;  
    cuenta1 = new CCuenta("Antonio López", "1000-2365-85-123456789", 2500, 0);  
    return cuenta1;  
}
```

Francisco M. Colls Gutiérrez.

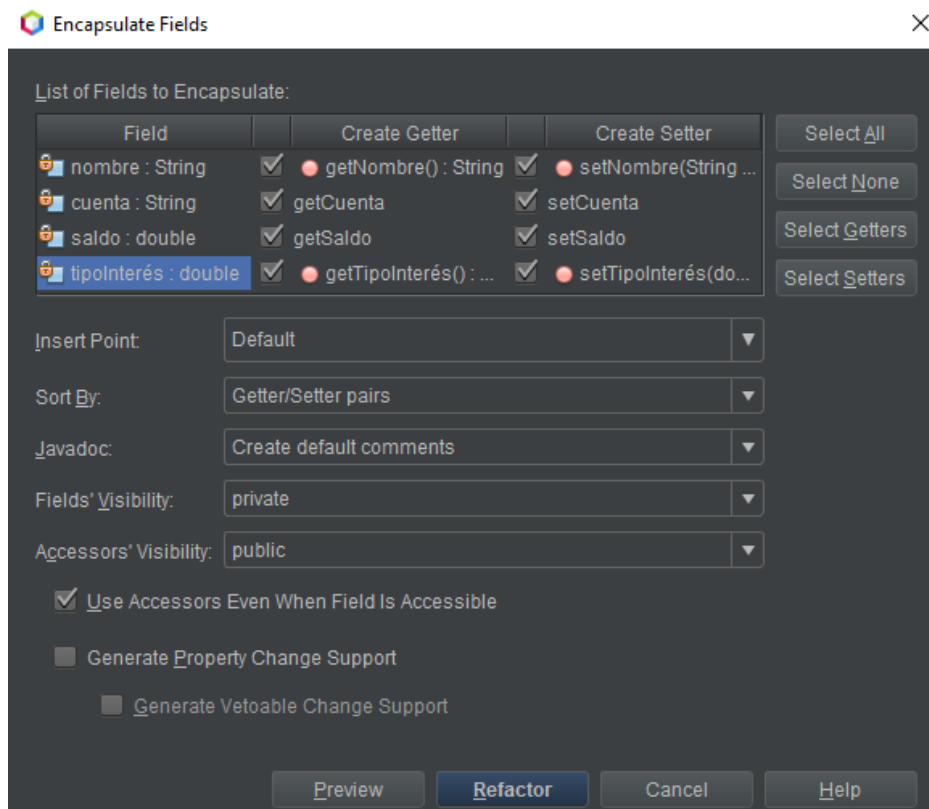
4. Encapsular los atributos de la clase CCuenta.

Seleccionamos los atributos que queremos encapsular.

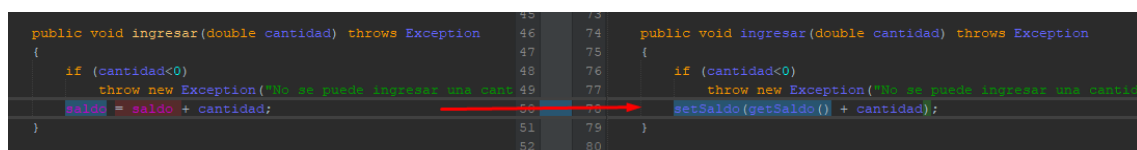
Clic derecho>Refactor> Encapsulate Fields



Nos muestra los métodos que serán creados. Podemos hacer clic en Preview para ver los cambios que serán realizados o hacer clic en Refactor directamente

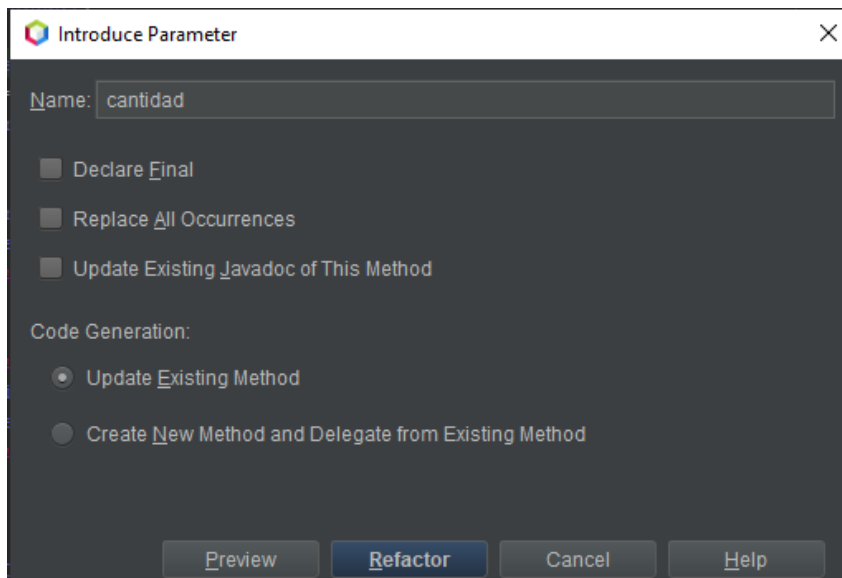
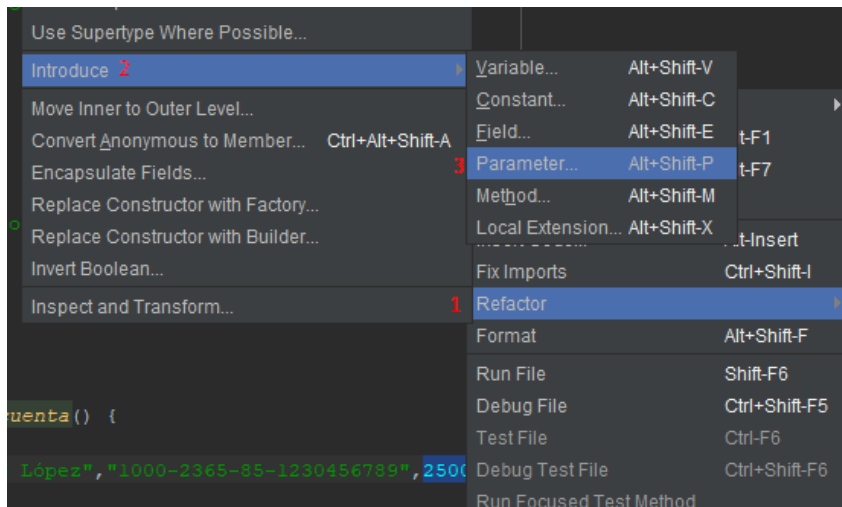


Previsualización de los cambios.



Francisco M. Colls Gutiérrez.

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



```
private static CCuenta operativa_cuenta(float cantidad) {  
    CCuenta cuental;  
    cuental = new CCuenta("Antonio López", "1000-2365-85-1230456789", cantidad, 0);  
    return cuental;  
}
```

Francisco M. Colls Gutiérrez.

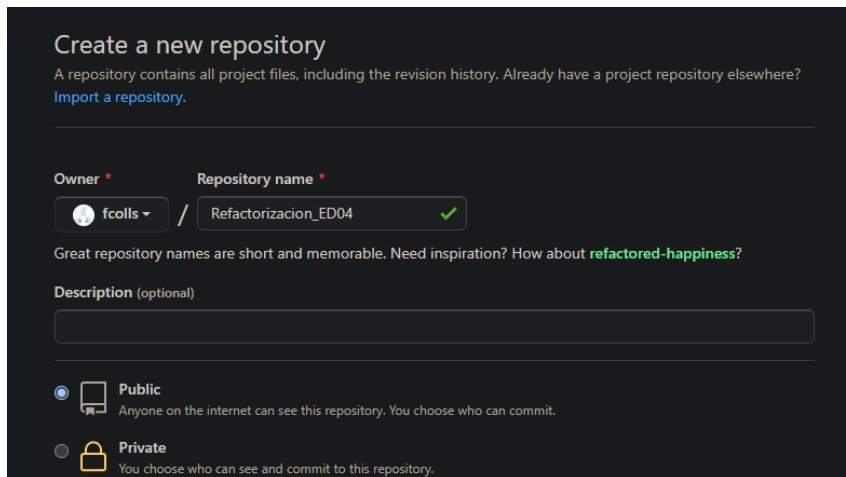
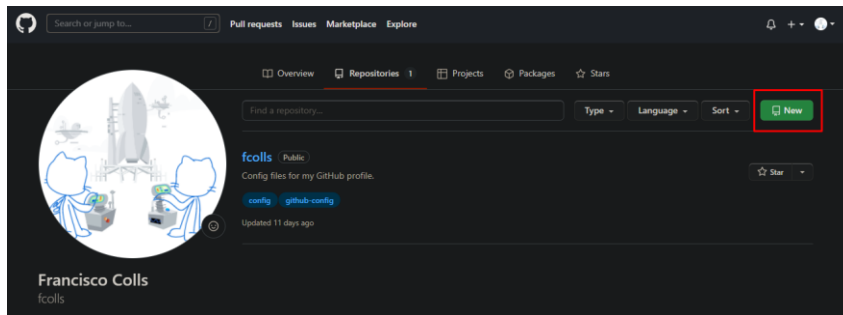
GIT.

Enlace al repositorio de github.

https://github.com/fcolls/Refactorizacion_ED04

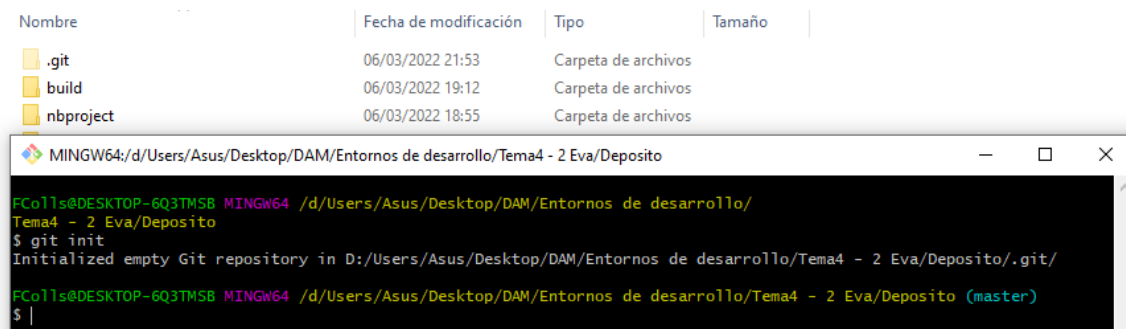
1. Configurar GIT para el proyecto. Crear un repositorio público en GitHub.

Creemos un nuevo repositorio publico en github llamado Refactorizacion_ED04.



En la carpeta del proyecto abrimos el terminal GitBash y ejecutamos el siguiente comando:

>git init



Francisco M. Colls Gutiérrez.

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

Para hacer un commit agregamos todos los archivos al Staging y luego hacemos un commit inicial con los siguientes comandos:

```
>git add .  
>git commit -m "Commit inicial"
```

```
FColls@DESKTOP-6Q3TMSB MINGW64 /d/Users/Asus/Desktop/DAM/Entornos de desarrollo/Tema4 - 2 Eva/Deposito (master)  
$ git add .  
warning: LF will be replaced by CRLF in src/cuentas/ccuenta.java.  
The file will have its original line endings in your working directory  
FColls@DESKTOP-6Q3TMSB MINGW64 /d/Users/Asus/Desktop/DAM/Entornos de desarrollo/Tema4 - 2 Eva/Deposito (master)  
$ git status  
On branch master  
  
No commits yet  
  
changes to be committed:  
  (use "git rm --cached <file>..." to unstage)  
    new file:   build.xml  
    new file:   build/classes/.netbeans_automatic_build  
    new file:   build/classes/.netbeans_update_resources  
    new file:   build/classes/cuentas/ccuenta.class  
    new file:   build/classes/cuentas/Main.class  
    new file:   manifest.mf  
    new file:   nbproject/build-impl.xml  
    new file:   nbproject/genfiles.properties  
    new file:   nbproject/private/private.properties
```

```
FColls@DESKTOP-6Q3TMSB MINGW64 /d/Users/Asus/Desktop/DAM/Entornos de desarrollo/Tema4 - 2 Eva/Deposito (master)  
$ git commit -m "Commit inicial"  
[master (root-commit) dd460fc] commit inicial  
14 files changed, 2129 insertions(+)  
create mode 100644 build.xml  
create mode 100644 build/classes/.netbeans_automatic_build  
create mode 100644 build/classes/.netbeans_update_resources  
create mode 100644 build/classes/cuentas/ccuenta.class  
create mode 100644 build/classes/cuentas/Main.class  
create mode 100644 manifest.mf  
create mode 100644 nbproject/build-impl.xml  
create mode 100644 nbproject/genfiles.properties  
create mode 100644 nbproject/private/private.properties  
create mode 100644 nbproject/private/private.xml
```

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

Para mostrar los commit usamos el siguiente comando:

```
>git log --oneline
```

```
FColls@DESKTOP-6Q3TMSB MINGW64 /d/Users/Asus/Desktop/DAM/Entornos de desarrollo/Tema4 - 2 Eva/Deposito (master)  
$ git log --oneline  
dd460fc (HEAD -> master) Commit inicial
```

Para subir los ficheros a github ejecutamos los siguientes comandos:

```
>git remote add origin https://github.com/fcolls/Refactorizacion_ED04.git  
>git branch -M main  
>git push -u origin main
```

```
FColls@DESKTOP-6Q3TMSB MINGW64 /d/Users/Asus/Desktop/DAM/Entornos de desarrollo/Tema4 - 2 Eva/Deposito (main)  
$ git push -u origin main  
Enumerating objects: 22, done.  
Counting objects: 100% (22/22), done.  
Delta compression using up to 4 threads  
Compressing objects: 100% (19/19), done.  
Writing objects: 100% (22/22), 19.01 KiB | 2.71 MiB/s, done.  
Total 22 (delta 0), reused 0 (delta 0), pack-reused 0  
To https://github.com/fcolls/Refactorizacion_ED04.git  
* [new branch]      main -> main  
branch 'main' set up to track 'origin/main'.
```

Francisco M. Colls Gutiérrez.

JAVADOC.

1. Insertar comentarios JavaDoc en la clase CCuenta.

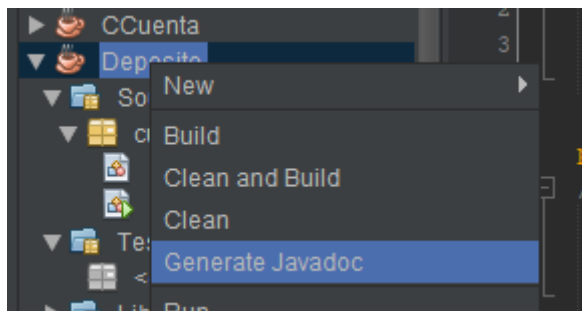
Se insertaron comentarios en los métodos de las clases del proyecto.

```
/**
 * Segundo constructor.
 * @param nom nombre del titular de la cuenta
 * @param cue número de cuenta.
 * @param sal saldo de la cuenta
 * @param tipo tipo de interés de la cuenta.
 */

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

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

Para generar el archivo JavaDoc, hacemos clic derecho sobre el proyecto y luego en Generate JavaDoc.



El archivo JavaDoc se guarda en la carpeta del proyecto dist/javadoc.

fcolls Agregando imagen historial commit e444cc8 11 hours ago 4 commits		
build/classes	Agregados comentarios JavaDoc	11 hours ago
dist/javadoc	Generado archivo JavaDoc	11 hours ago
nbproject	Agregados comentarios JavaDoc	11 hours ago
src/cuentas	Agregados comentarios JavaDoc	11 hours ago
.gitignore	Generado archivo JavaDoc	11 hours ago

Francisco M. Colls Gutiérrez.