

## Proyecto de Entorno y Desarrollo

José Ramon.

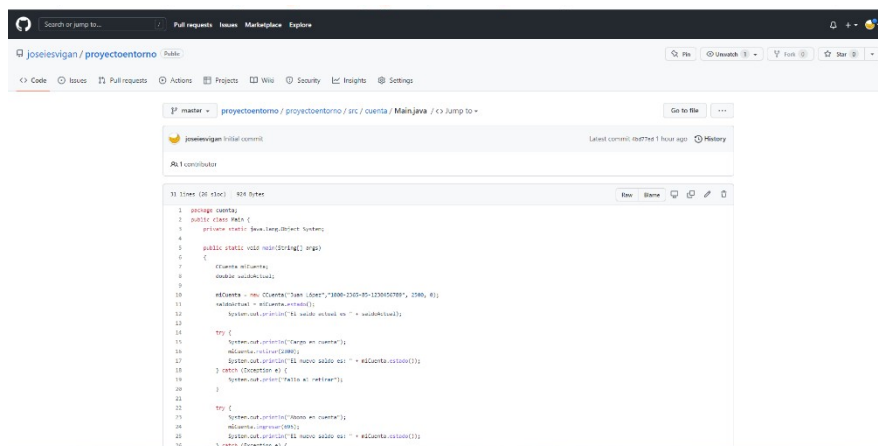
### Enlazar, refactorizar y mostrar código con la Ide IntelliJ y Git-Hub.

En este primer apartado mostrare paso a paso los puntos que se han establecido en la tarea. Comenzare por el apartado 2 ya que es esencial antes de meternos con el 1.

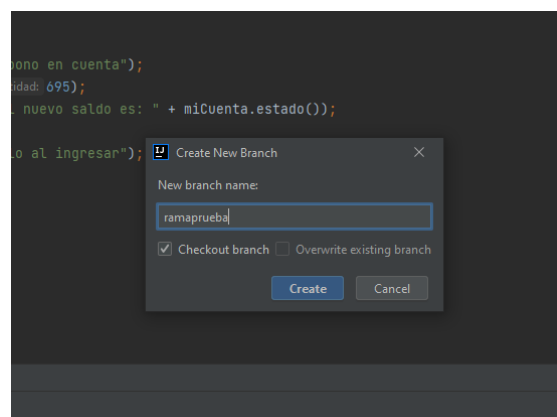
### GIT

Configurar Git para el proyecto, creando un repositorio público y lo llamaremos DAW\_ED04.

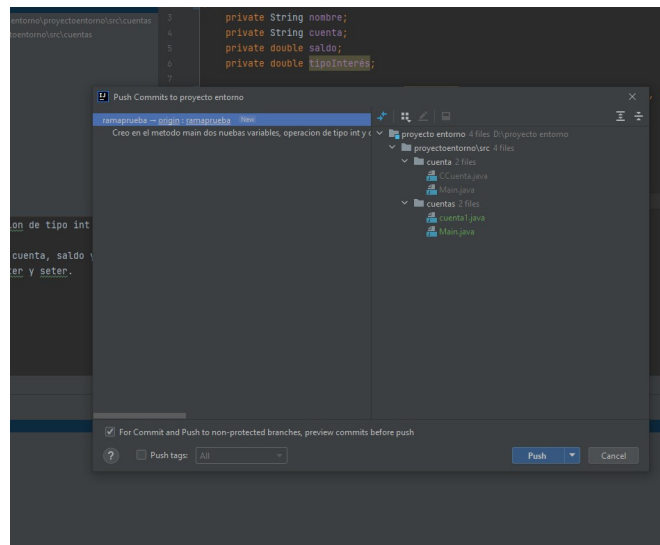
1-Una vez configurado el primer paso, subo el proyecto al Git.



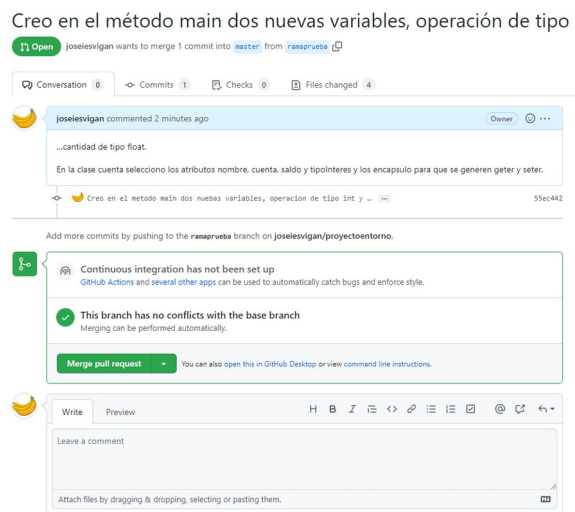
2- Cargado el proyecto en el repositorio de Git, creo una rama aparte la cual será la rama que modificare para poder avanzar en la prueba, la llamare “ramaprueba”.



3- Completado todos los puntos del apartado 1, realizo una operación ‘commit’ y ‘push’.



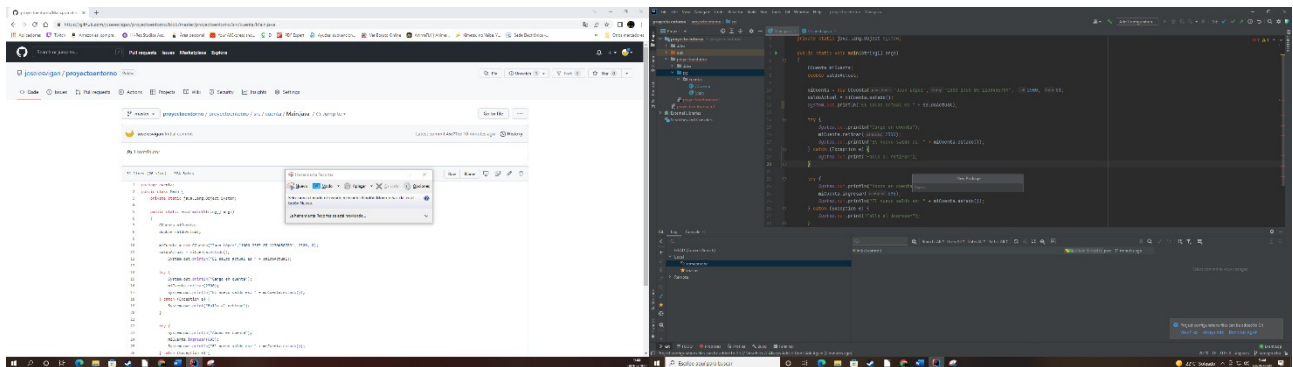
4-En este punto integro los cambios realizados en la rama ‘ramaprueba’ en la principal. Si todo sale correctamente no habrá problemas y te dejara fusionar la rama secundaria con la principal, confirmamos el pull y el merge.



Una vez completados los apartados del Git, utilizo la rama creada “ramaprueba” en donde completare todos los pasos del apartado 1.

## Refactorización

1-Creo un package y lo llamo cuentas, dentro de él guardare la clase Main y la clase CCuentas.



1-Cambio el nombre de la variable “mi Cuenta” por “cuenta1” como me pide el ejercicio, e introduzco el método operativa cuenta para englobar las sentencias de la clase Main.

2-Introducir el método operativa\_cuenta.

Creo en el método Main, la sentencia operativa\_cuenta que engloba las sentencias del objeto “cuenta1”.

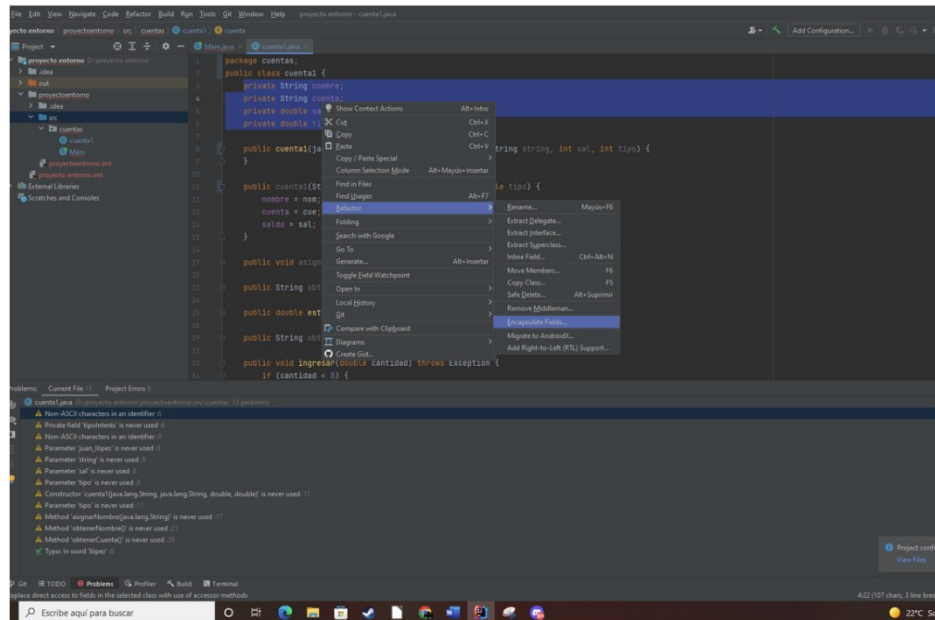
```
public static void main(String[] args) {
    operativa_cuenta();
}

public static void operativa_cuenta() {
    int operacion;
    float cantidad;
```

3-Añado los parámetros al método anteriormente creado, operación y cantidad.

```
int operacion;
float cantidad;
```

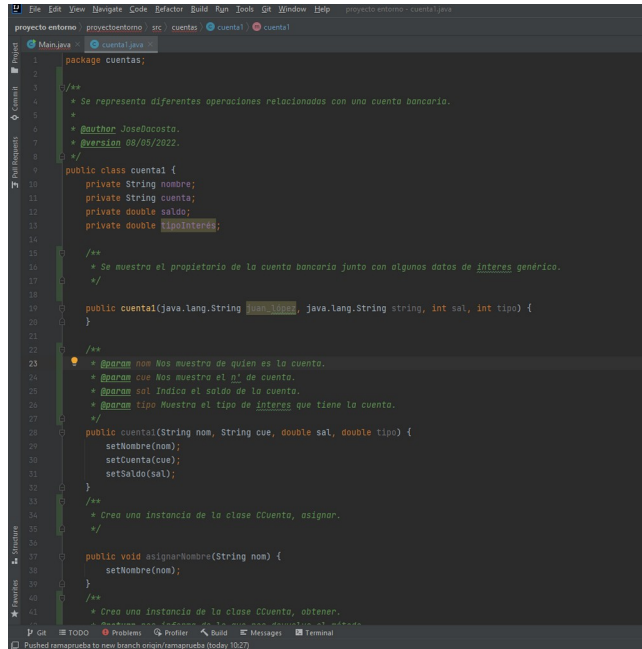
4-En este punto encapsulamos los atributos de la clase CCuenta, son 4, “nombre” “cuenta” “saldo” y “tipoInteres”. Los seleccionamos, botón derecho y buscamos la opción ‘refactor’ y ‘encapsulate fields’, esto genera a cada uno de los atributos un ‘get y set’.



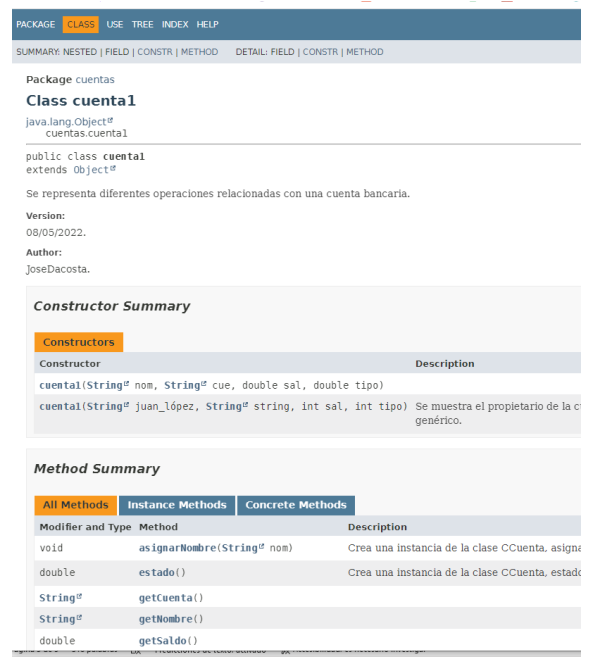
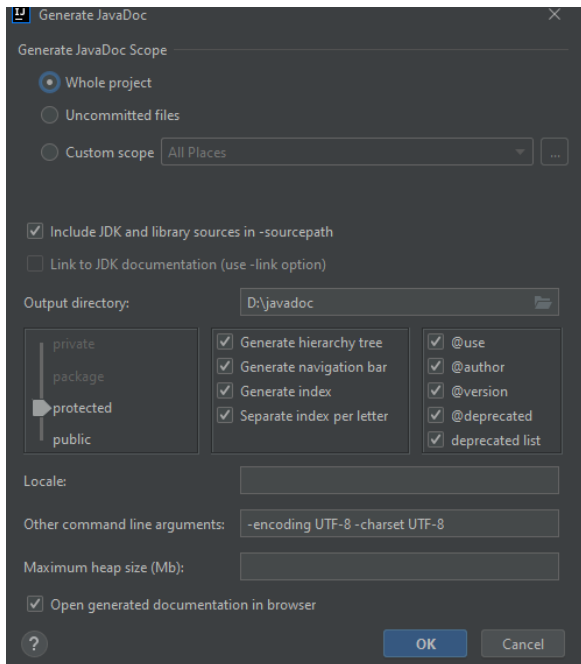
Con esto ya completaríamos el apartado 1.

## JAVADOC

1-Insertaremos todos los comentarios posibles en la clase CCuenta, renombrada 'cuenta1'.



## 2-Genero el Documento Java DOC.



Una vez generado el documento podremos visualizarlo a través de nuestro navegador.