

## **Git Colaborativo**

*Juan García Mejías  
Mauro Contador de Alfonso  
Asignatura - 1º DAW*

## Índice

<b>Índice.....</b>	<b>1</b>
<b>General.....</b>	<b>2</b>
Escenario del proyecto.....	2
<b>Etapa 1: Configuración Inicial.....</b>	<b>2</b>
Creado de la copia local.....	2
Creado de la rama.....	2
Creado de la carpeta src.....	2
Subida del proyecto base.....	3
Usuario B obtiene el mismo repositorio.....	3
<b>Etapa 2: Desarrollo Colaborativo.....</b>	<b>4</b>
Implementación de suma y resta por parte del Usuario A.....	4
Subida al repositorio remoto.....	4
El Usuario B añade las funcionalidades multiplicación y potencia sin actualizar previamente.....	5
Subida de los cambios realizados por el Usuario B.....	6
Aparición del conflicto.....	7
Subida del código fusionado.....	8
<b>Etapa 3: Rama de pruebas y versiones.....</b>	<b>8</b>
Creación de la rama testing a partir de develop.....	8
Fusionado de los cambios de develop en testing.....	8
Etiquetado de la v1.0 y registro de las funciones básicas.....	9
<b>Etapa 4: Nuevas versiones.....</b>	<b>9</b>
Versión 2.0.....	9
Versión 3.0.....	11
<b>Etapa 5: Pruebas con versiones anteriores.....</b>	<b>12</b>
Consulta de las versiones disponibles del proyecto mediante el sistema de control de versiones.....	12
Prueba de la ejecución del código de las versiones v1.0, v2.0 y v3.0 por separado.....	12
Comparación de los resultados obtenidos en cada versión y anotación de las diferencias observadas.....	14
Vuelta a la rama de desarrollo activa una vez finalizadas las pruebas.....	14
Registro de commits en formato gráfico.....	15

## General

### **Escenario del proyecto**

**Proyecto:** Calculadora colaborativa.

**Repositorio remoto:** [https://github.com/mconalf2108/1DAW\\_CalculadoraColaborativa.git](https://github.com/mconalf2108/1DAW_CalculadoraColaborativa.git)

**Lenguaje:** Java.

**Ramas:**

- main*: versión estable.
- develop*: Rama de desarrollo.
- testing*: rama de pruebas

**Participantes:**

- Usuario A:** Mauro Contador de Alfonso.
- Usuario B:** Juan García Mejías.

## Etapa 1: Configuración Inicial

### **Creado de la copia local**

El primer paso a realizar viene a ser la copia local del repositorio remoto e iniciar la rama *develop*, por parte del Usuario A;

```
1DAW-mconalf2108@A22-PC12 MINGW64 ~/Desktop
$ git clone https://github.com/mconalf2108/1DAW_CalculadoraColaborativa.git
Cloning into '1DAW_CalculadoraColaborativa'...
warning: You appear to have cloned an empty repository.
```

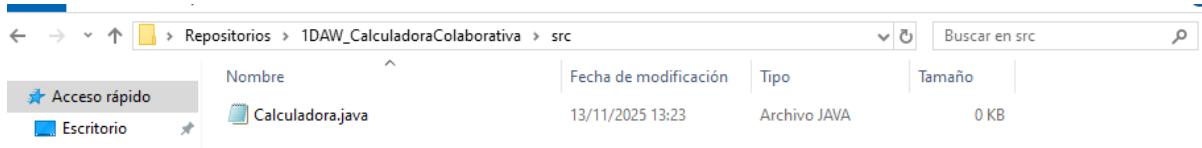
### **Creado de la rama**

Para crear la rama usaremos el comando *git checkout -b develop*;

```
1DAW-mconalf2108@A22-PC12 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa (main)
$ git checkout -b develop
Switched to a new branch 'develop'
```

### **Creado de la carpeta src**

Tras ello, el Usuario A, crea una carpeta llamada *src* y dentro de ella situamos el archivo *Calculadora.java* con un mensaje de inicio simple;



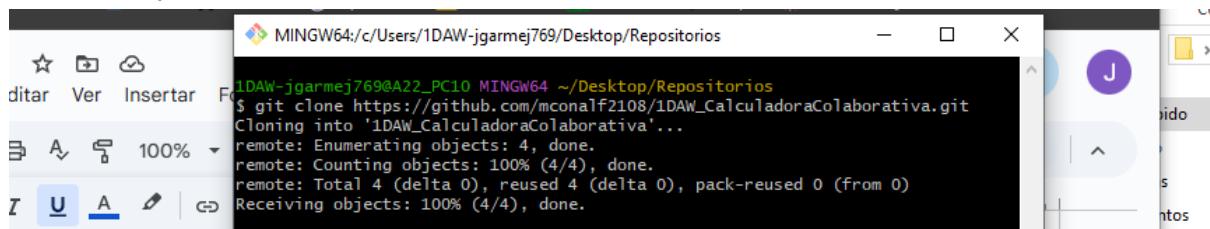
## Subida del proyecto base

Para este paso, debemos hacer *git add* y *git commit* para subir el archivo base a nuestro repositorio remoto;

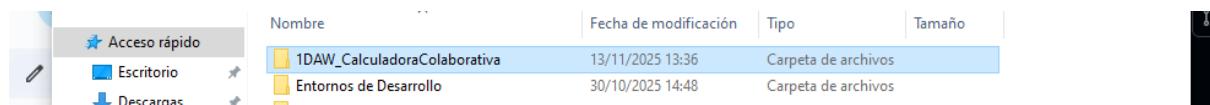
```
1DAW-mconalf2108@A22-PC12 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa (develop)
$ git push -u origin develop
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Writing objects: 100% (4/4), 284 bytes | 284.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/mconalf2108/1DAW_CalculadoraColaborativa.git
 * [new branch]      develop -> develop
branch 'develop' set up to track 'origin/develop'.
```

## Usuario B obtiene el mismo repositorio

En este paso, el Usuario B, deberá hacer un *git clone*, para obtener el repositorio remoto de forma local y sincronizarlo;



Tras ello, abrimos el Explorador de Archivos y comprobamos que el repositorio se ha bajado y sincronizado;



## Etapa 2: Desarrollo Colaborativo

### **Implementación de suma y resta por parte del Usuario A**

En este apartado, el Usuario A, comenzará a programar la calculadora. Accederá al archivo Calculadora.java y creará las funciones *suma* y *resta*, además de probarlas en el main, quedando de la siguiente forma;

```
package Calculadora;

public class Calculadora {

    // Método principal para probar las funciones
    public static void main(String[] args) {
        System.out.println("Calculadora colaborativa - v1.0 (Funciones A)");

        // Pruebas de Usuario A
        int resultadoSuma = suma(10, 5);
        System.out.println("Resultado Suma (10+5): " + resultadoSuma);

        int resultadoResta = resta(20, 8);
        System.out.println("Resultado Resta (20-8): " + resultadoResta);
    }

    // --- Funciones de Usuario A ---

    public static int suma(int a, int b) {
        return a + b;
    }

    public static int resta(int a, int b) {
        return a - b;
    }
}
```

### **Subida al repositorio remoto**

Con los cambios ya guardados, el Usuario A subirá mediante *git add*, *git commit* y *git push*, los cambios realizados sobre *Calculadora.java* al repositorio remoto;

```
1DAW-mconalf2108@A22-PC12 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa (develop)
$ git commit -m +"subida calculadora"
[develop e185d91] +subida calculadora
Committer: unknown <1DAW-mconalf2108@IESMARTINEZM.ES>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit
```

Una vez que el *git commit* se ha realizado con éxito, el Usuario A, realiza el *git push* y sube los cambios al repositorio remoto;

```
1DAW-mconalf2108@A22-PC12 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa (develop)
$ git push -u origin develop
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 618 bytes | 618.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/mconalf2108/1DAW_CalculadoraColaborativa.git
  7c04006..e185d91  develop -> develop
branch 'develop' set up to track 'origin/develop'.
```

### **El Usuario B añade las funcionalidades multiplicación y potencia sin actualizar previamente**

En esta ocasión, el Usuario B, implementará las funciones *multiplicación* y *división*, probando el en main si funcionan correctamente. Cabe destacar que todo esto debe hacerse sin actualizar el archivo *Calculadora.java* del repositorio remoto;

```
/*
public class Calculadora {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }
    // --- Funciones de Usuario B ---

    public static int multiplicacion(int a, int b) {
        return a * b;
    }

    public static double division(int a, int b) {
        return (double)a / b;
    }
}
```

## **Subida de los cambios realizados por el Usuario B**

A continuación, el Usuario B, guardará los cambios realizados y los subirá al repositorio remoto mediante *git add*, *git commit* y *git push*;

```
1DAW-jgarmej769@A22_PC10 MINGW64 ~/Desktop/Repositorios/1DAW_CalculadoraColaborativa (develop)
$ git commit -m"Subida de los cambios realizados por el Usuario B"
[develop 90f856a] Subida de los cambios realizados por el Usuario B
 1 file changed, 29 insertions(+)

1DAW-jgarmej769@A22_PC10 MINGW64 ~/Desktop/Repositorios/1DAW_CalculadoraColaborativa (develop)
$ git push
To https://github.com/mconalf2108/1DAW_CalculadoraColaborativa.git
 ! [rejected]      develop -> develop (fetch first)
error: failed to push some refs to 'https://github.com/mconalf2108/1DAW_CalculadoraColaborativa.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

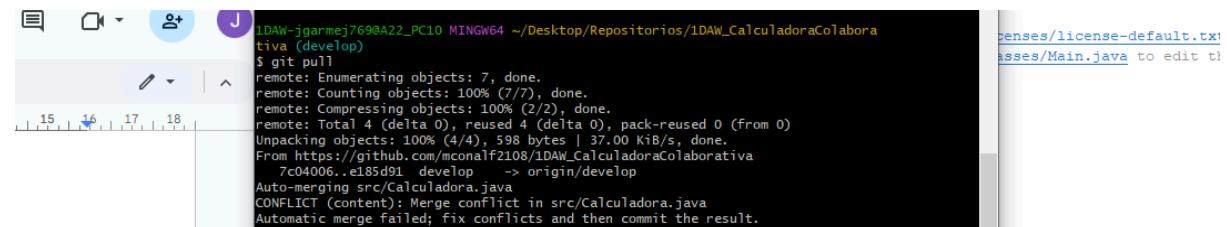
1DAW-jgarmej769@A22_PC10 MINGW64 ~/Desktop/Repositorios/1DAW_CalculadoraColaborativa (develop)
$
```

El *commit* se realiza con total normalidad, pero al realizar el *push*, nos salta el mensaje de error de la imagen superior.

Esto quiere decir que el Usuario B está intentando subir el archivo *Calculadora.java* pero ya hay un archivo *Calculadora.java* en GitHub más reciente. Es decir, nos está indicando que estamos intentando subir el archivo mientras que el Usuario A, ya ha subido un archivo con cambios que nosotros no hemos bajado mediante un *git pull*.

## Aparición del conflicto

Una vez leído el error anterior, hacemos el *git pull* que se nos solicita, y se crea el conflicto buscado;



```
1DAW-jgarmej769@A22_PCl0 MINGW64 ~/Desktop/Repositorios/1DAW_CalculadoraColaborativa
tiva (develop)
$ git pull
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (4/4), 598 bytes | 37.00 KiB/s, done.
From https://github.com/mconalfz108/1DAW_CalculadoraColaborativa
 7c04006..e185d91 develop -> origin/develop
Auto-merging src/Calculadora.java
CONFLICT (content): Merge conflict in src/Calculadora.java
Automatic merge failed; fix conflicts and then commit the result.
```

Git ha detectado que el archivo que me estoy bajando está escrito y combina ambos para que seamos nosotros, quienes pongamos solución a dicho error, el archivo se quedaría de la siguiente forma;

```
<<<<< HEAD
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package gitcolaborativo;
/**
 * @author 1DAW-jgarmej769
 */
public class Calculadora {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }
    // --- Funciones de Usuario B ---

    public static int multiplicacion(int a, int b) {
        return a * b;
    }

    public static double division(int a, int b) {
        return (double)a / b;
    }
}
=====
package Calculadora;

public class Calculadora {

    // Método principal para probar las funciones
    public static void main(String[] args) {
        System.out.println("Calculadora colaborativa - v1.0 (Funciones A)");

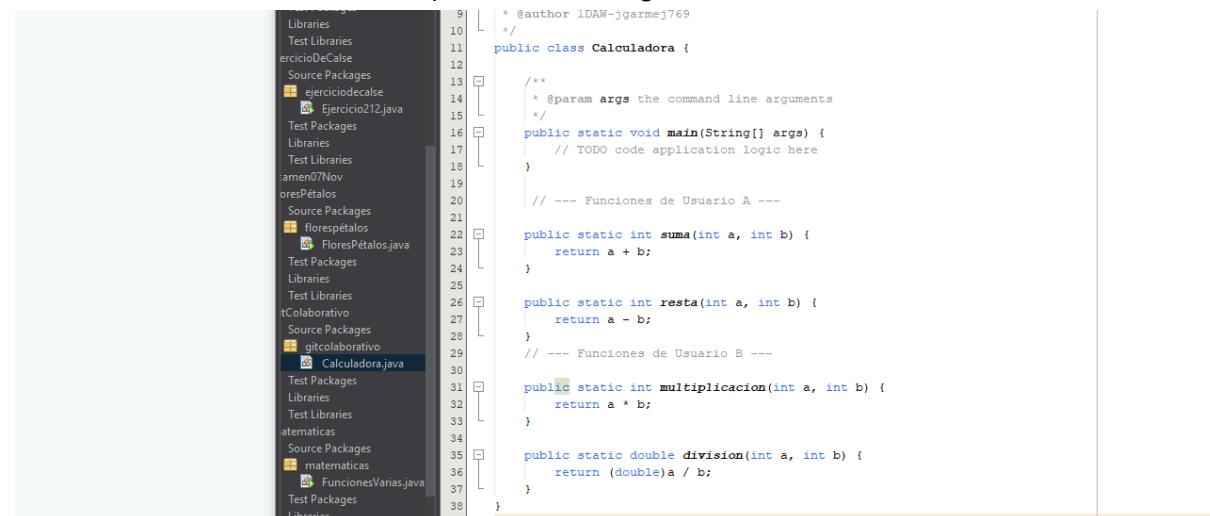
        // Pruebas de Usuario A
        int resultadoSuma = suma(10, 5);
        System.out.println("Resultado Suma (10+5): " + resultadoSuma);

        int resultadoResta = resta(20, 8);
        System.out.println("Resultado Resta (20-8): " + resultadoResta);
    }
    // --- Funciones de Usuario A ---

    public static int suma(int a, int b) {
        return a + b;
    }

    public static int resta(int a, int b) {
        return a - b;
    }
}
>>>> e185d9190147b61528a274537b6f21586da7c568
```

Aquí podemos observar como Git ha fusionado los códigos, sin sobreescribir ni machacar ninguno, y somos nosotros los que debemos arreglar el error. En nuestro entorno de desarrollo, el error solucionado quedaría de la siguiente forma;



```
* @author 1DAW-jgarmej769
public class Calculadora {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

    // --- Funciones de Usuario A ---

    public static int suma(int a, int b) {
        return a + b;
    }

    public static int resta(int a, int b) {
        return a - b;
    }
    // --- Funciones de Usuario B ---

    public static int multiplicacion(int a, int b) {
        return a * b;
    }

    public static double division(int a, int b) {
        return (double)a / b;
    }
}
```

## Subida del código fusionado

Una vez el Usuario B ha fusionado correctamente el código, el propio Usuario B, subirá el archivo al repositorio remoto.

```
MINGW64:/c/Users/1DAW-jgarmej769/Desktop/Repositorios/1DAW_CalculadoraColaborativa - □ :  
Changes to be committed:  
  modified:   src/Calculadora.java  
  
[DAW-jgarmej769@A22_PC10 MINGW64 ~/Desktop/Repositorios/1DAW_CalculadoraColaborativa (develop)  
MERGING)  
$ git commit -m"Archivo con el código ya fusionado"  
[develop 2f2aa90] Archivo con el código ya fusionado  
  
[DAW-jgarmej769@A22_PC10 MINGW64 ~/Desktop/Repositorios/1DAW_CalculadoraColaborativa (develop)  
]  
$ git push  
Enumerating objects: 14, done.  
Counting objects: 100% (14/14), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (4/4), done.  
Writing objects: 100% (8/8), 1016 bytes | 1016.00 KiB/s, done.  
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (1/1), done.  
To https://github.com/mconalf2108/1DAW_CalculadoraColaborativa.git  
 e185d91..2f2aa90  develop -> develop  
  
[DAW-jgarmej769@A22_PC10 MINGW64 ~/Desktop/Repositorios/1DAW_CalculadoraColaborativa (develop)  
]  
$
```

Tras ello, el Usuario A, se actualizará su repositorio, obteniendo la versión fusionada.

## Etapa 3: Rama de pruebas y versiones

### **Creación de la rama testing a partir de develop**

A continuación, el Usuario A, deberá crear la rama *testing* a partir de la rama *develop*.

```
[DAW-mconalf2108@A22-PC12 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa (develop)  
]  
$ git checkout -b testing  
Switched to a new branch 'testing'  
  
[DAW-mconalf2108@A22-PC12 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa (testing)  
]
```

### **Fusionado de los cambios de develop en testing**

Para preparar la primera versión de prueba, debemos fusionar los cambios de *develop* en la rama recién creada *testing*. Esto viene a comprobar y asegurar que la rama *testing* está perfectamente sincronizada con *develop*, lo que se comprueba con *git merge develop*;

```
[DAW-mconalf2108@A22-PC12 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa (testing)  
]  
$ git checkout testing  
Already on 'testing'  
Your branch is up to date with 'origin/testing'.  
  
[DAW-mconalf2108@A22-PC12 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa (testing)  
]  
$ git merge develop  
Already up to date.
```

## ***Etiquetado de la v1.0 y registro de las funciones básicas***

Una vez comprobada la fusión entre las ramas *develop* y *testing*, etiquetamos esta versión como v1.0 y dejamos registrado que contiene las funciones básicas (suma, resta, multiplicación y división).

```
1DAW-mconalf2108@A22-PC12 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa (testing)
$ git tag -a v1.0 -m"Versión 1.0 : Contiene las funciones básicas(suma,resta,multiplicacion y division)"
```

## **Etapa 4: Nuevas versiones**

### **Versión 2.0**

A continuación, el Usuario B, creará una nueva funcionalidad; *potencia*. Actualizará el método principal y fusionará los cambios en la rama *testing*, etiquetando como *v2.0*. El código nuevo quedará de la siguiente forma;

```
4102 38
      39     public static int potencia(int a, int b) {
      40         int potencia = 1;
      41         for (int i = 0; i < b; i++) {
      42             potencia = potencia * a;
      43         }
      44     return potencia;
      45 }
```

A continuación lo actualizamos y subimos al repositorio;

```
1DAW-jgarmej769@A22_PC10 MINGW64 ~/Desktop/Repositorios/1DAW_CalculadoraColaborativa (develop)
$ git add .
1DAW-jgarmej769@A22_PC10 MINGW64 ~/Desktop/Repositorios/1DAW_CalculadoraColaborativa (develop)
$ git commit -m "Subida del archivo con la nueva funcionalidad potencia"
[develop 570fe05] Subida del archivo con la nueva funcionalidad potencia
 1 file changed, 9 insertions(+)

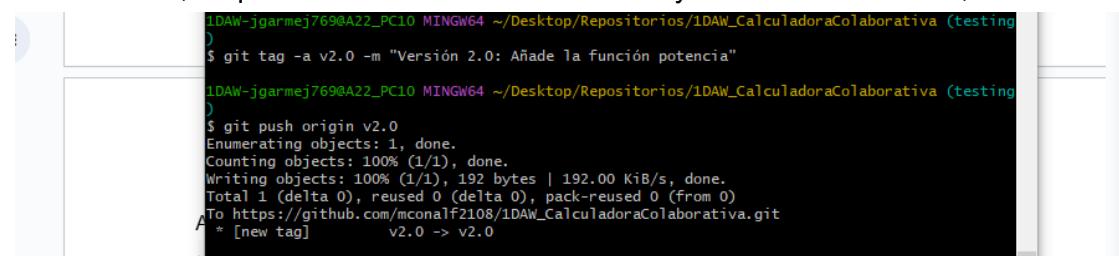
1DAW-jgarmej769@A22_PC10 MINGW64 ~/Desktop/Repositorios/1DAW_CalculadoraColaborativa (develop)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 457 bytes | 457.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/mconalf2108/1DAW_CalculadoraColaborativa.git
 2f2aa90..570fe05  develop -> develop
```

A continuación llevamos los nuevos cambios de *develop* a la rama *testing*:



```
Already up-to-date.  
1DAW-jgarmej769@A22_PC10 MINGW64 ~/Desktop/Repositorios/1DAW_CalculadoraColaborativa (testing)  
$ git merge develop  
Updating 2f2aa90..570fe05  
Fast-forward  
 src/Calculadora.java | 9 ++++++++  
 1 file changed, 9 insertions(+)  
  
1DAW-jgarmej769@A22_PC10 MINGW64 ~/Desktop/Repositorios/1DAW_CalculadoraColaborativa (testing)  
$ git push origin testing  
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
To https://github.com/mconalf2108/1DAW_CalculadoraColaborativa.git  
 2f2aa90..570fe05 testing -> testing
```

Para finalizar, etiquetamos esta versión como *v2.0* y lo subimos a GitHub:



```
1DAW-jgarmej769@A22_PC10 MINGW64 ~/Desktop/Repositorios/1DAW_CalculadoraColaborativa (testing)  
$ git tag -a v2.0 -m "Versión 2.0: Añade la función potencia"  
  
1DAW-jgarmej769@A22_PC10 MINGW64 ~/Desktop/Repositorios/1DAW_CalculadoraColaborativa (testing)  
$ git push origin v2.0  
Enumerating objects: 1, done.  
Counting objects: 100% (1/1), done.  
Writing objects: 100% (1/1), 192 bytes | 192.00 KiB/s, done.  
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
To https://github.com/mconalf2108/1DAW_CalculadoraColaborativa.git  
 * [new tag]      v2.0 -> v2.0
```

## Versión 3.0

En esta versión, el Usuario A, modificará la función de división para que controle el caso de división por cero y lance una excepción o muestre un mensaje de error;

```
// --- Función Modificada (v3.0 - Básica) ---
public static double division(int a, int b) {
    if (b == 0) {
        System.out.println("Error: No se puede dividir por cero.");
        return 0.0;
    }
    return (double)a / b;
}
```

A continuación, subimos los cambios al repositorio remoto;

```
1DAW-mconalf2108@A22-PC12 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa (testing)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 632 bytes | 632.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/mconalf2108/1DAW_CalculadoraColaborativa.git
  570fe05..4564509  testing -> testing
```

A continuación procedemos a etiquetar la versión v3.0.

```
1DAW-mconalf2108@A22-PC12 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa (testing)
$ git merge develop
Already up to date.

1DAW-mconalf2108@A22-PC12 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa (testing)
$ git tag -a v3.0 -m"contiene el manejo de ceros"

1DAW-mconalf2108@A22-PC12 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa (testing)
$ git push origin v3.0
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 191 bytes | 191.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/mconalf2108/1DAW_CalculadoraColaborativa.git
 * [new tag]      v3.0 -> v3.0
```

## Etapa 5: Pruebas con versiones anteriores

**Consulta de las versiones disponibles del proyecto mediante el sistema de control de versiones.**

Accedemos a Git e introducimos el comando `git tag`, con ello, el bash nos devuelve que versiones tenemos disponibles en el mismo; `v1.0`, `v2.0` y `v3.0`.

```
1DAW-mconalf2108@A22-PC12 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa ((v1.0)
)
$ git tag
v1.0
v2.0
v3.0
```

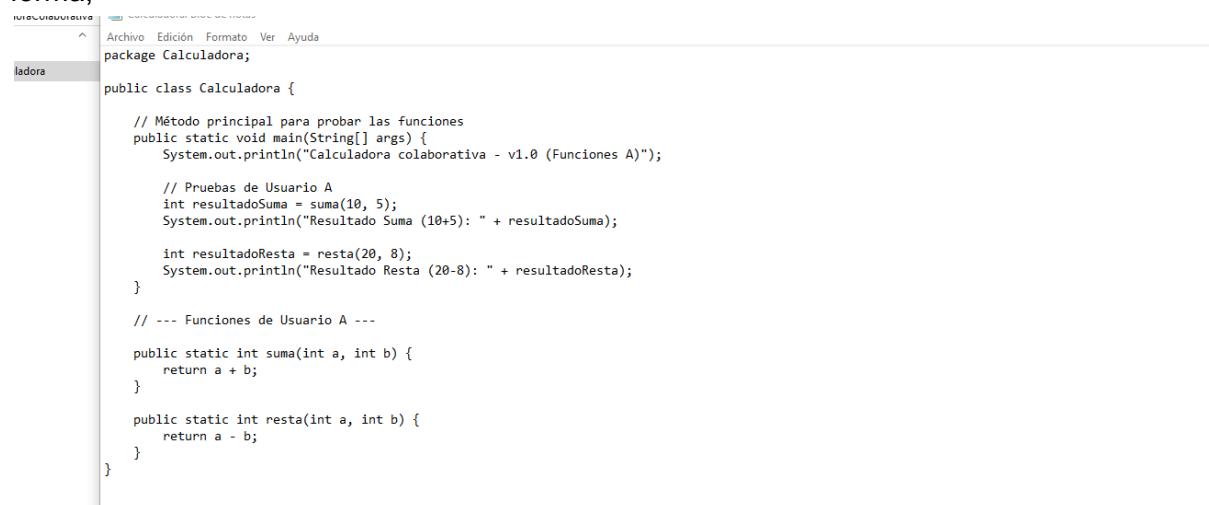
**Prueba de la ejecución del código de las versiones v1.0, v2.0 y v3.0 por separado.**

Para ello, en nuestro bash, introducimos el comando `git switch --detach v1.0`,

```
MINGW64:/c/Users/1DAW-jgarmej769/Desktop/1DAW_CalculadoraColaborativa... — □

1DAW-jgarmej769@A22-PC16 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa/src
.0))
$ git switch --detach v1.0
Previous HEAD position was 570fe05 Subida del archivo con la nueva funcional
potencia
HEAD is now at e185d91 +subida calculadora
```

con ello, actualizamos el archivo que tenemos en nuestra carpeta, y se ve de la siguiente forma;



```
Calculadora.java - Untitled Project - Eclipse IDE for Java Developers
^ Archivo Edición Formato Ver Ayuda
Calculadora;
public class Calculadora {

    // Método principal para probar las funciones
    public static void main(String[] args) {
        System.out.println("Calculadora colaborativa - v1.0 (Funciones A)");

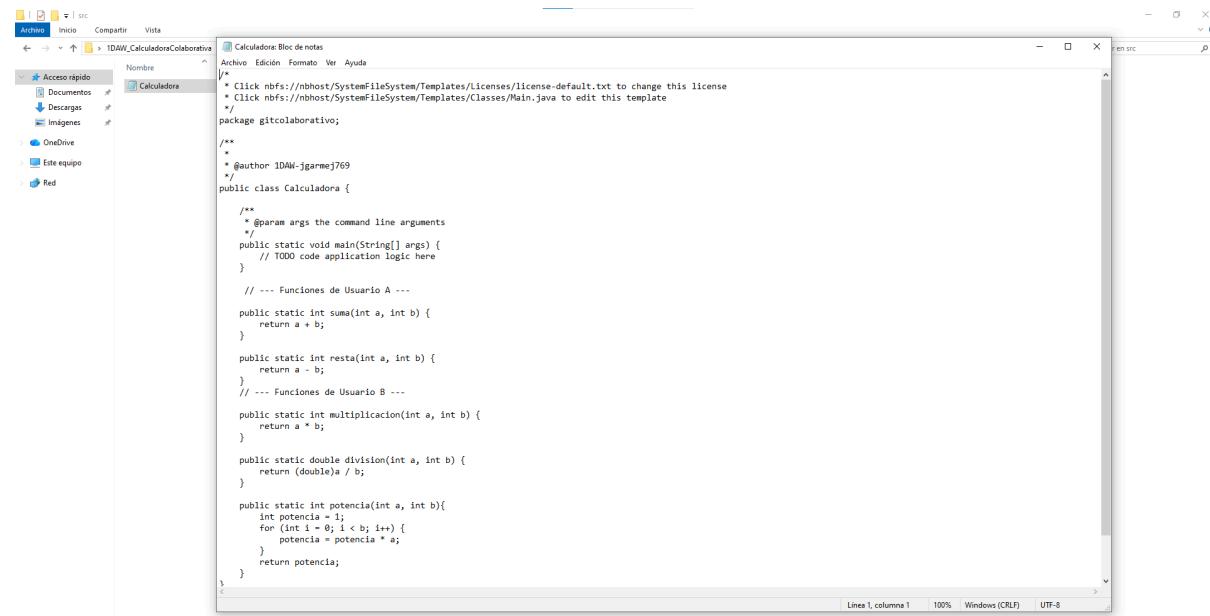
        // Pruebas de Usuario A
        int resultadoSuma = suma(10, 5);
        System.out.println("Resultado Suma (10+5): " + resultadoSuma);

        int resultadoResta = resta(20, 8);
        System.out.println("Resultado Resta (20-8): " + resultadoResta);
    }

    // --- Funciones de Usuario A ---
    public static int suma(int a, int b) {
        return a + b;
    }

    public static int resta(int a, int b) {
        return a - b;
    }
}
```

Tras ello, lo hacemos con la v2.0; con el comando `git switch -detach v2.0;`



```
Archivos Inicio Compartir Vista
Acceso rápido Calculadora Bloc de notas
Nombre
Calculadora
Calculadora: Bloc de notas
Archivo Edición Formato Ver Ayuda
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package gitcolaborativo;

/**
 * @param args the command line arguments
 */
public class Calculadora {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

    // --- Funciones de Usuario A ---

    public static int suma(int a, int b) {
        return a + b;
    }

    public static int resta(int a, int b) {
        return a - b;
    }
    // --- Funciones de Usuario B ---

    public static int multiplicacion(int a, int b) {
        return a * b;
    }

    public static double division(int a, int b) {
        return (double)a / b;
    }

    public static int potencia(int a, int b){
        int potencia = 1;
        for (int i = 0; i < b; i++) {
            potencia = potencia * a;
        }
        return potencia;
    }
}
```

Para finalizar, introducimos `git switch -detach v3.0`, y obtenemos la última versión;



```
Archivo Edición Formato Ver Ayuda
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package gitcolaborativo;

/**
 * @param args the command line arguments
 */
public class Calculadora {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

    // --- Funciones de Usuario A ---

    public static int suma(int a, int b) {
        return a + b;
    }

    public static int resta(int a, int b) {
        return a - b;
    }
    // --- Funciones de Usuario B ---

    public static int multiplicacion(int a, int b) {
        return a * b;
    }

    public static double division(int a, int b) {
        if (b==0){
            System.out.println("No se puede dividir por cero");
            return 1;
        }else{
            return (double)a / b;
        }
    }

    public static int potencia(int a, int b){
        int potencia = 1;
        for (int i = 0; i < b; i++) {
            potencia = potencia * a;
        }
        return potencia;
    }
}
```

## **Comparación de los resultados obtenidos en cada versión y anotación de las diferencias observadas.**

En este apartado, vamos a comentar las diferencias que se observan entre las diversas versiones;

La versión 2.0 es más completa que la anterior:

- v1.0: El programa solo sabía hacer cuentas básicas (sumar, restar, etc.).
- v2.0: Ahora también calcula potencias. Al usar el programa, verás que aparece el resultado de la potencia en la pantalla (por ejemplo,  $2^3 = 8.0$ ). Conclusión: El hecho de que aparezca este cálculo confirma que la actualización con las nuevas funciones se hizo correctamente.

Con respecto a los cambios que se observan de la v2.0 a la v3.0, podemos destacar que no son funcionales (no añade operaciones matemáticas nuevas), sino de robustez y calidad de software;

-v2.0: Si se fuerza una división por cero (ej.  $10 / 0$ ), el programa no tiene capacidad de respuesta controlada. Dependiendo de la implementación, el programa entrega un resultado matemático incorrecto (Infinity) o se rompe mostrando un "stack trace" de error de Java.

-v3.0: Al ejecutar la misma operación de división por cero, el sistema captura la excepción. En lugar de un error de sistema, el usuario ve un mensaje de advertencia personalizado ("No se puede dividir por cero").

## **Vuelta a la rama de desarrollo activa una vez finalizadas las pruebas.**

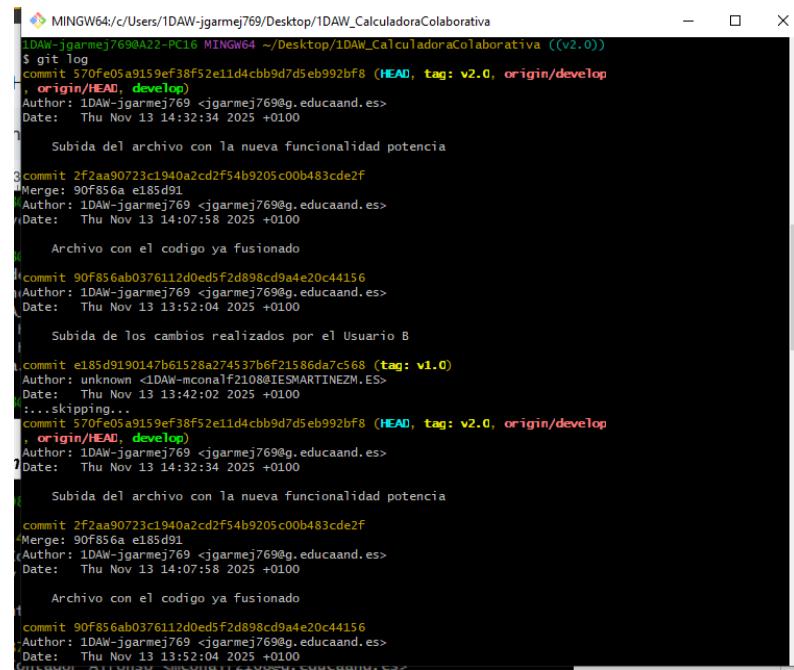
Finalmente, se nos pide que volvamos a la rama de desarrollo, para ello, simplemente introducimos el comando `git checkout develop`

```
1DAW-jgarmej769@A22-PC16 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa ((v2.0))
$ git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.

1DAW-jgarmej769@A22-PC16 MINGW64 ~/Desktop/1DAW_CalculadoraColaborativa (develop)
$
```

## Registro de commits en formato gráfico

Finalmente, utilizamos el comando `git log` para revisar el registro de commits;



```
MINGW64:/c/Users/1DAW-jgarmej769/Desktop/1DAW_CalculadoraColaborativa
$ git log
commit 570fe05a159ef38f52e11d4ccb9d7d5eb992bf8 (HEAD, tag: v2.0, origin/develop
origin/HEAD, develop)
Author: 1DAW-jgarmej769 <jgarmej769@g.educaand.es>
Date:   Thu Nov 13 14:32:34 2025 +0100

    Subida del archivo con la nueva funcionalidad potencia

commit 2f2aa90723c1940a2cd2f54b9205c00b483cde2f
Merge: 90f856a e185d91
Author: 1DAW-jgarmej769 <jgarmej769@g.educaand.es>
Date:   Thu Nov 13 14:07:58 2025 +0100

    Archivo con el codigo ya fusionado

commit 90f856ab0376112d0ed5f2d898cd9a4e20c44156
Author: 1DAW-jgarmej769 <jgarmej769@g.educaand.es>
Date:   Thu Nov 13 13:52:04 2025 +0100

    Subida de los cambios realizados por el Usuario B

commit e185d9190147b61528a274537b6f21586da7c568 (tag: v1.0)
Author: unknown <1DAW-mconalf2108@IESMARTINEZM.ES>
Date:   Thu Nov 13 13:42:02 2025 +0100
...skipping...
commit 570fe05a159ef38f52e11d4ccb9d7d5eb992bf8 (HEAD, tag: v2.0, origin/develop
origin/HEAD, develop)
Author: 1DAW-jgarmej769 <jgarmej769@g.educaand.es>
Date:   Thu Nov 13 14:32:34 2025 +0100

    Subida del archivo con la nueva funcionalidad potencia

commit 2f2aa90723c1940a2cd2f54b9205c00b483cde2f
Merge: 90f856a e185d91
Author: 1DAW-jgarmej769 <jgarmej769@g.educaand.es>
Date:   Thu Nov 13 14:07:58 2025 +0100

    Archivo con el codigo ya fusionado

commit 90f856ab0376112d0ed5f2d898cd9a4e20c44156
Author: 1DAW-jgarmej769 <jgarmej769@g.educaand.es>
Date:   Thu Nov 13 13:52:04 2025 +0100
```

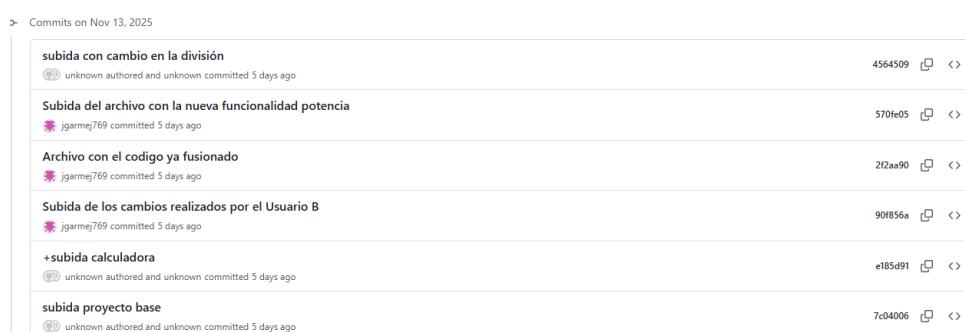
Para ver el repositorio de forma gráfica, debemos irnos a GitHub y desde ahí podemos obtenerlo. Con respecto a la rama `develop`, podemos observar;



Commits on Nov 13, 2025

- Subida del archivo con la nueva funcionalidad potencia
- Archivo con el codigo ya fusionado
- Subida de los cambios realizados por el Usuario B
- +subida calculadora
- subida proyecto base

Y, con respecto a la rama `testing`, observamos lo siguiente;



Commits on Nov 13, 2025

- subida con cambio en la división
- Subida del archivo con la nueva funcionalidad potencia
- Archivo con el codigo ya fusionado
- Subida de los cambios realizados por el Usuario B
- subida proyecto base