

PARTE 1: Implementación en java:Partiendo del diagrama de casos de uso creado en el proyecto durante el primer trimestre, elige un caso de uso cualquiera e implementa una clase que desarrolle total o parcialmente el citado caso de uso.

En la implementación de la clase, hay que especificar TODOS los atributos de la misma. Pero solo se IMPLEMENTARÁ UN MÉTODO de todos los propuestos en la misma. Podéis elegir el que queráis (incluso el más sencillo de todos. NO SE VALORARÁ LA COMPLEJIDAD DEL MISMO ya que no es el tema principal de esta materia)

Es imprescindible que el método que se desarrolle tenga uno o más parámetros de entrada y que devuelva algún valor de salida.

Aquí mostramos el metodoCalculadora que hemos implementado a la Calculadora:

The screenshot shows an IDE with two tabs: 'Calculadora.java' and 'metodoCalculadora.java'. The 'metodoCalculadora.java' tab is active, displaying the following Java code:

```

6
7  /**
8   *
9   * @author rober
10  */
11  public class metodoCalculadora {
12
13      public static double dividir(double numero1, double numero2) {
14          if (numero2 == 0) {
15              throw new ArithmeticException("No se puede dividir entre cero");
16          }
17          return numero1 / numero2;
18      }
19
20      public static void main(String[] args) {
21          double resultado = dividir(numero1:20, numero2:2);
22          System.out.println("El resultado de la división es: " + resultado);
23      }
24  }
25

```

Below the code editor, the 'Output - Run (metodoCalculadora)' window shows the execution results:

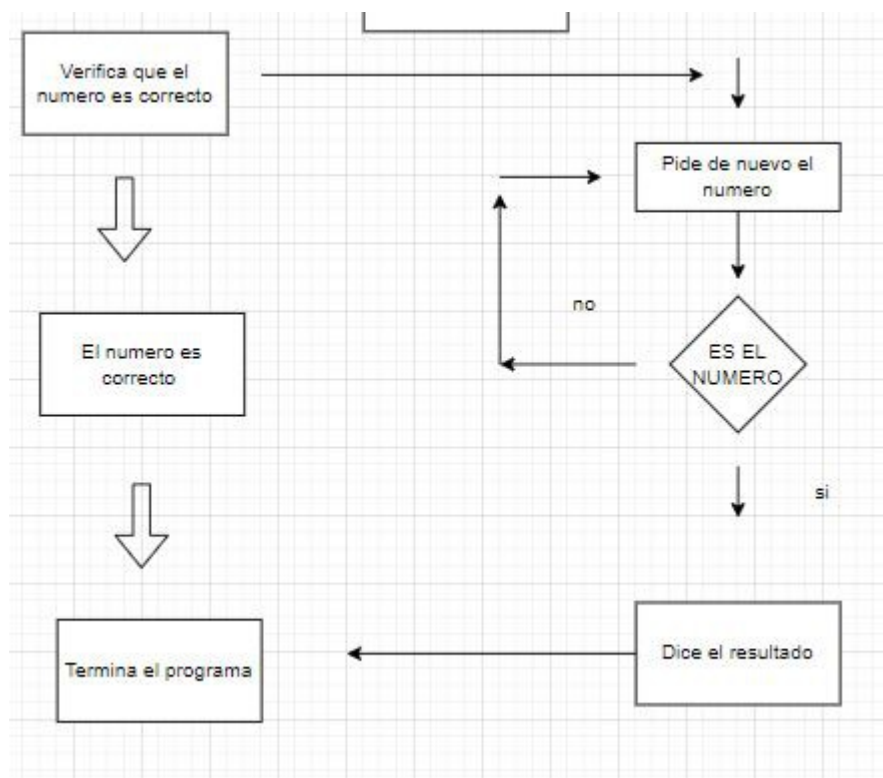
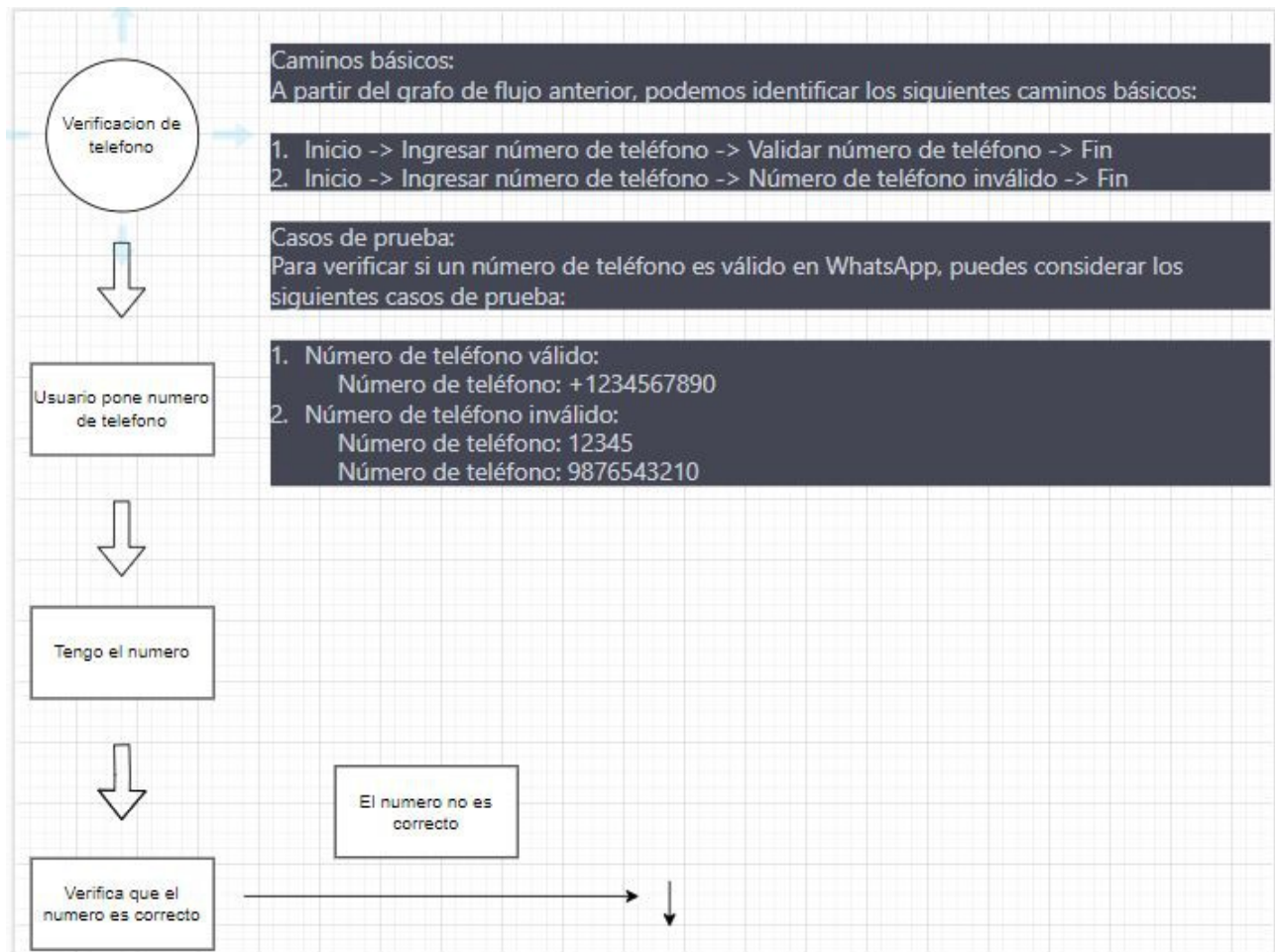
```

-----< com.mycompany:calculadora >-----
Building calculadora 1.0-SNAPSHOT
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ calculadora ---
El resultado de la división es: 10.0
BUILD SUCCESS
Total time: 13.276 s
Finished at: 2023-05-21T19:22:42+02:00

```

PARTE 2: Pruebas:

1. Dibujar el **grafo de flujo** del algoritmo (en DIA)
2. Calcular los **caminos básicos** de dicho grafo.
3. Representar los **casos de prueba** indicando el valor que hay que dar a los parámetros.



PARTE 3: SCV

Creamos un nuevo repositorio:

Crear un nuevo repositorio

Un repositorio contiene todos los archivos del proyecto, incluido el historial de revisión. ¿Ya tienes un repositorio de proyectos en otro lugar? [Importar un repositorio](#).

Dueño *

 rober17777777 ▾

nombre del repositorio *

/ ProyectoEntornos

✓ ProyectoEntornos está disponible.

Los grandes nombres de repositorios son cortos y fáciles de recordar. ¿Necesitas inspiración? ¿Qué hay de [percebe musical](#) ?

Descripción (opcional)

Parte 3 del trabajo: SVC



Público

Cualquier persona en Internet puede ver este repositorio. Tú eliges quién puede comprometerse.



Privado

Tú eliges quién puede ver y comprometerse con este repositorio.

Inicialice este repositorio con:



Agregar un archivo **LÉAME**

Aquí es donde puede escribir una descripción larga para su proyecto. [Obtenga más información sobre los archivos README](#).

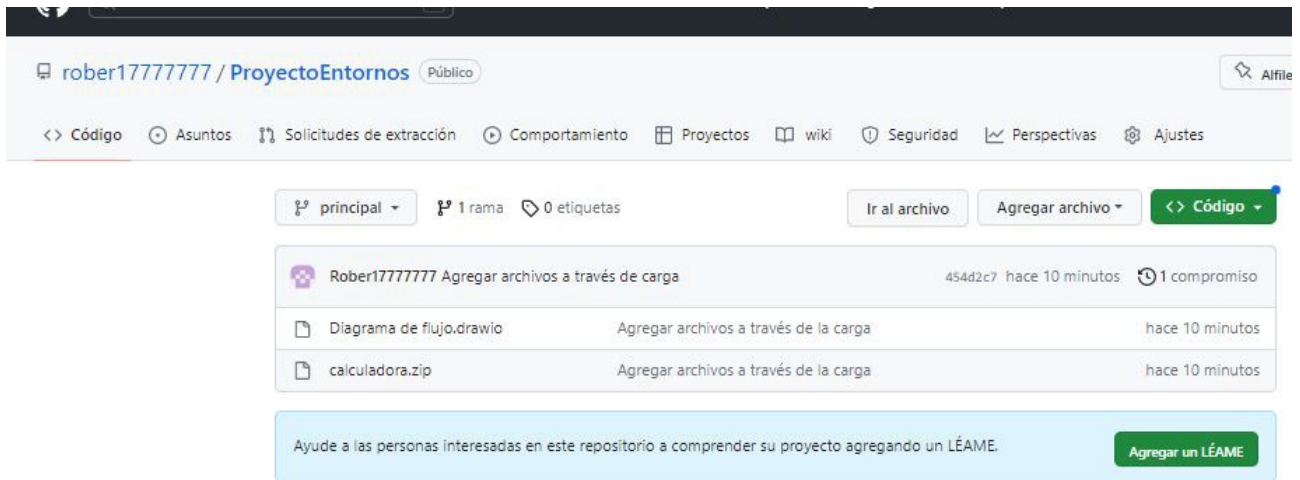
Agregar **.gitignore**

Plantilla **.gitignore** : ninguna ▾

Elija qué archivos no rastrear de una lista de plantillas. [Obtén más información sobre cómo ignorar archivos](#).

Elija una licencia

Y agregamos los archivos para este trabajo.



Finalmente añadimos también este pdf.