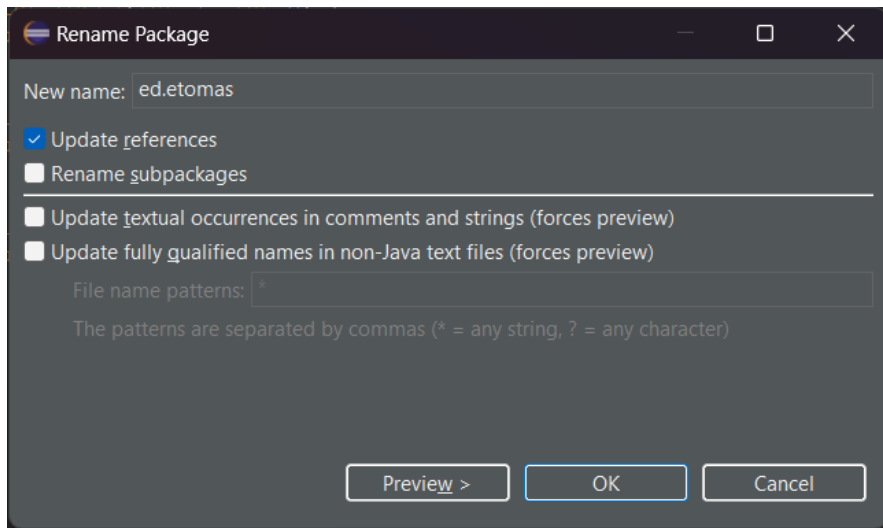


## 1.Modifica la clase Calculadora.java

Renombra el paquete ed.camilo a ed.<tu\_nombre>



## 2. Documenta el código de la clase Calculadora usando comentarios JavaDoc:

Para la clase:

Debe tener: una descripción corta que explique claramente para qué sirve esta clase. Añade en otro párrafo una descripción más larga.

Añade los tags JavaDoc necesarios para documentar:

El autor (pon tu nombre completo)

La versión actual de la clase (3.7)

Que está disponible desde la versión 2.1, junio de 2022.

```
*Calculadora.java x
1 package ed.etomas;
2
3 /**
4  * Realiza operaciones simples matematicas
5  * Realiza sumas, restas, divisiones y multiplicaciones.
6  * @author etomas
7  * @version 3.7
8  * @since 2.1, jun 2022
9  */
10
11
12
13
14 public class Calculadora {
15
```

Para cada uno de los métodos, documenta:

Descripción explicando que hace el método.

Parámetros con la descripción más adecuada.

Valor que devuelve con una descripción adecuada.

Si el método genera una excepción (error de ejecución), pon el tipo de excepción y la descripción de la misma (por ejemplo, el caso de la división por cero)

```
public class Calculadora {

    /**
     * Suma de @param a y @param b
     * Realiza la suma de 2 numeros
     * @param a numero entero que actua como sumando
     * @param b numero entero que actua como sumando
     * @return valor int resultado de sumar a y b
     */
    public int suma(int a, int b) {
        return a + b;
    }

    /**
     * Resta de @param a y @param b
     * Realiza la resta de 2 numeros
     * @param a numero entero que actua como minuendo
     * @param b numero entero que actua como sustraendo
     * @return diferencia valor int resultado de restar a y b
     */
    public int resta(int a, int b) {
        return a - b;
    }

    /**
     * Multiplicacion de @param a y @param b
     * Realiza la Multiplicacion de 2 numeros
     * @param a numero entero que actua como multiplicando
     * @param b numero entero que actua como multiplicador
     * @return valor int resultado de multiplicar a y b
     */
    public int multiplica(int a, int b){
        return a*b;
    }

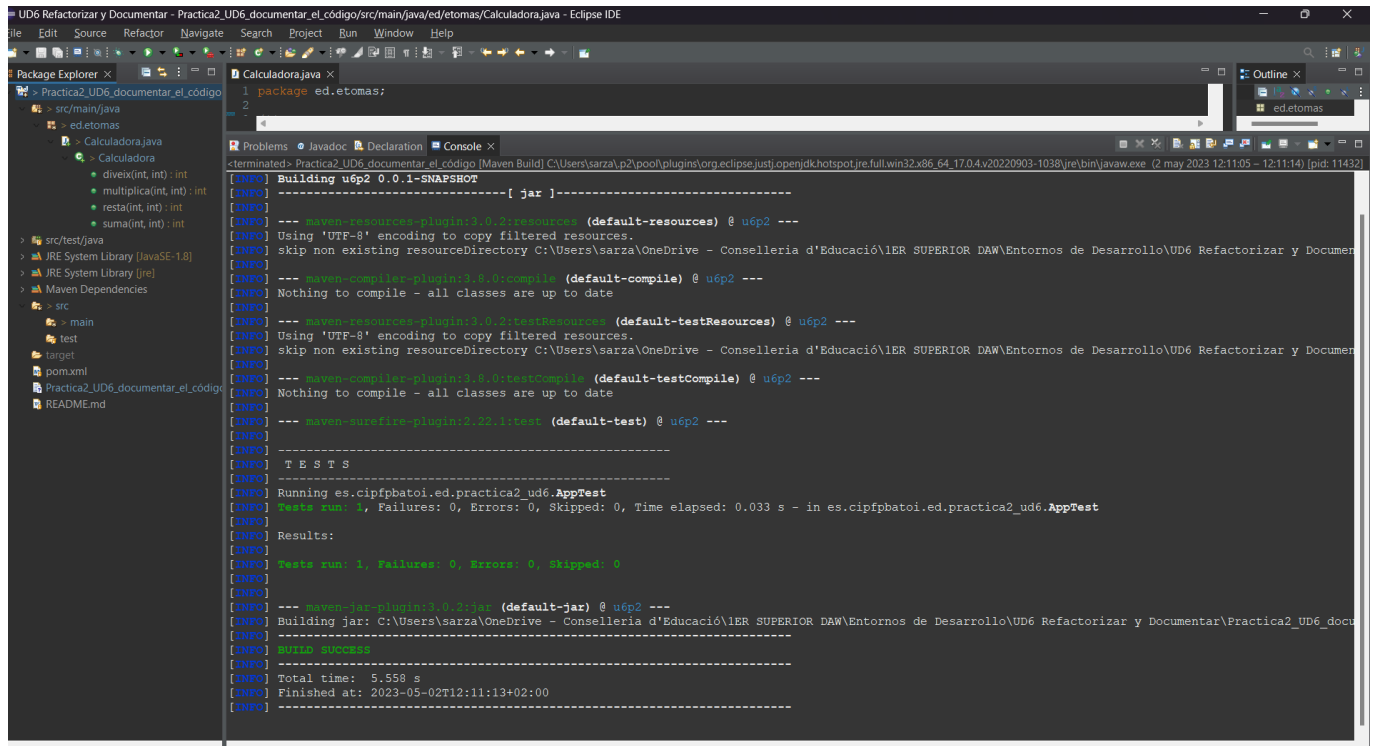
    /**
     * Division de @param a y @param b
     * Realiza la division de 2 numeros
     * @param a numero entero que actua como dividendo
     * @param b numero entero que actua como divisor
     * @return valor int resultado de dividir a y b
     * @exception division por 0 ArithmeticException
     */
}
```

3. Genera en eclipse la documentación Javadoc del proyecto.

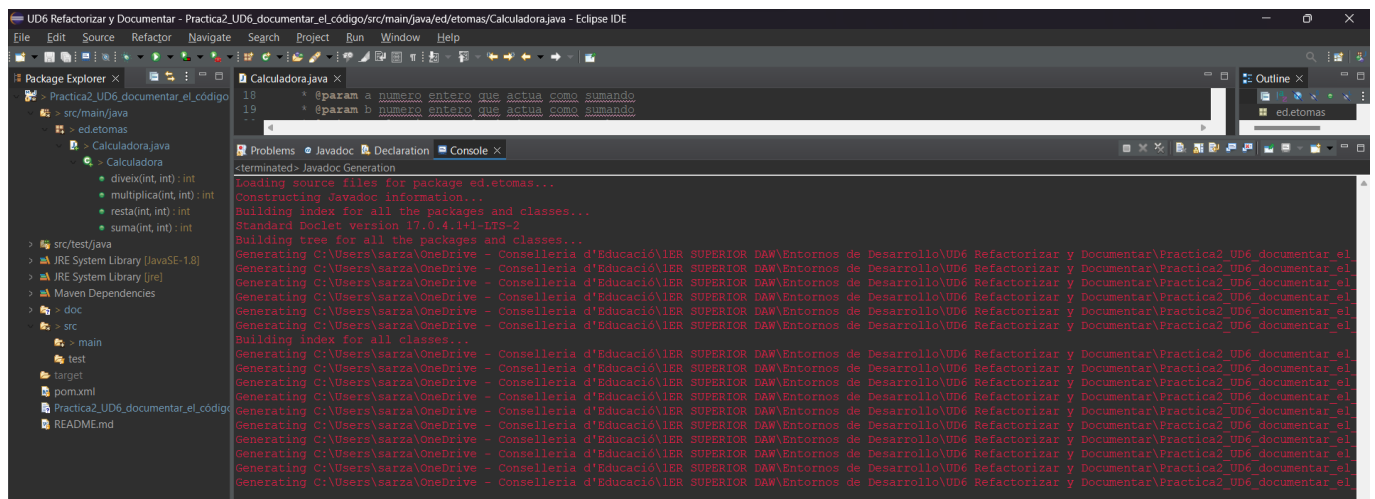
Arregla los posibles errors/warnings que aparezcan. Fijate en la descripción del error y vuelve a generar la documentación hasta que funcione correctamente.

Haz una captura de la Consola: debe verse todo el proceso de generación completo.

Haz una captura en la vista Package Explorer del directorio doc resultante.



The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays the project structure: 'Practica2\_UD6\_documentar\_el\_codigo' with sub-packages 'src/main/java' and 'src/test/java'. The 'src/main/java' package contains 'ed.etomas' and 'Calculadora.java'. The 'Calculadora.java' file is open in the editor, showing a simple class with methods 'divide(int, int)', 'multiplica(int, int)', 'resta(int, int)', and 'suma(int, int)'. The Console view on the right shows the output of a Maven build. The build is successful, with messages indicating that resources were copied, classes were compiled, and tests were run successfully. The build time is 5.558 s, and it finished at 2023-05-02T12:11:13+02:00.



The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays the project structure. The 'Calculadora.java' file is open in the editor, showing a simple class with methods 'divide(int, int)', 'multiplica(int, int)', 'resta(int, int)', and 'suma(int, int)'. The Console view on the right shows the output of Javadoc generation. The process starts with 'Loading source files for package ed.etomas...' and 'Constructing Javadoc information...'. It then builds the index for all packages and classes, and finally generates the Javadoc files. The console output shows the path to the generated Javadoc files: 'C:\Users\sarza\OneDrive - Conselleria d'Educació\1ER SUPERIOR DAW\Entornos de Desarrollo\UD6 Refactorizar y Documentar\Practica2\_UD6\_documentar\_el\_codigo\doc'.

Revisa si la documentación se ha añadido correctamente a git, en caso contrario, revisa el fichero .gitignore.

```

PS C:\Users\sarza\OneDrive - Conselleria d'Educació\1ER SUPERIOR DAW\Entornos de Desarrollo\UD6 Refactorizar y Documentar\Practica2_UD6_document
ar_el_código> git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   "Practica2_UD6_documentar_el_c\303\263digo-Tomas,Ethan.docx"
    new file:   doc/allclasses-index.html
    new file:   doc/allpackages-index.html
    new file:   doc/ed/etomas/Calculadora.html
    new file:   doc/ed/etomas/class-use/Calculadora.html
    new file:   doc/ed/etomas/package-summary.html
    new file:   doc/ed/etomas/package-tree.html
    new file:   doc/ed/etomas/package-use.html
    new file:   doc/element-list
    new file:   doc/es/cipfpbatoi/ed/practica2_ud6/AppTest.html
    new file:   doc/es/cipfpbatoi/ed/practica2_ud6/class-use/AppTest.html
    new file:   doc/es/cipfpbatoi/ed/practica2_ud6/package-summary.html
    new file:   doc/es/cipfpbatoi/ed/practica2_ud6/package-tree.html
    new file:   doc/es/cipfpbatoi/ed/practica2_ud6/package-use.html
    new file:   doc/help-doc.html
    new file:   doc/index-files/index-1.html
    new file:   doc/index-files/index-2.html
    new file:   doc/index-files/index-3.html
    new file:   doc/index-files/index-4.html
    new file:   doc/index-files/index-5.html
    new file:   doc/index-files/index-6.html
    new file:   doc/index-files/index-7.html
    new file:   doc/index.html
    new file:   doc/jquery-ui.overrides.css
    new file:   doc/legal/COPYRIGHT
    new file:   doc/legal/LICENSE
    new file:   doc/legal/jquery.md

```

4. Haz una captura del código fuente documentado resultante de la clase Calculadora

(En este caso, es posible que no sea necesario un commit)

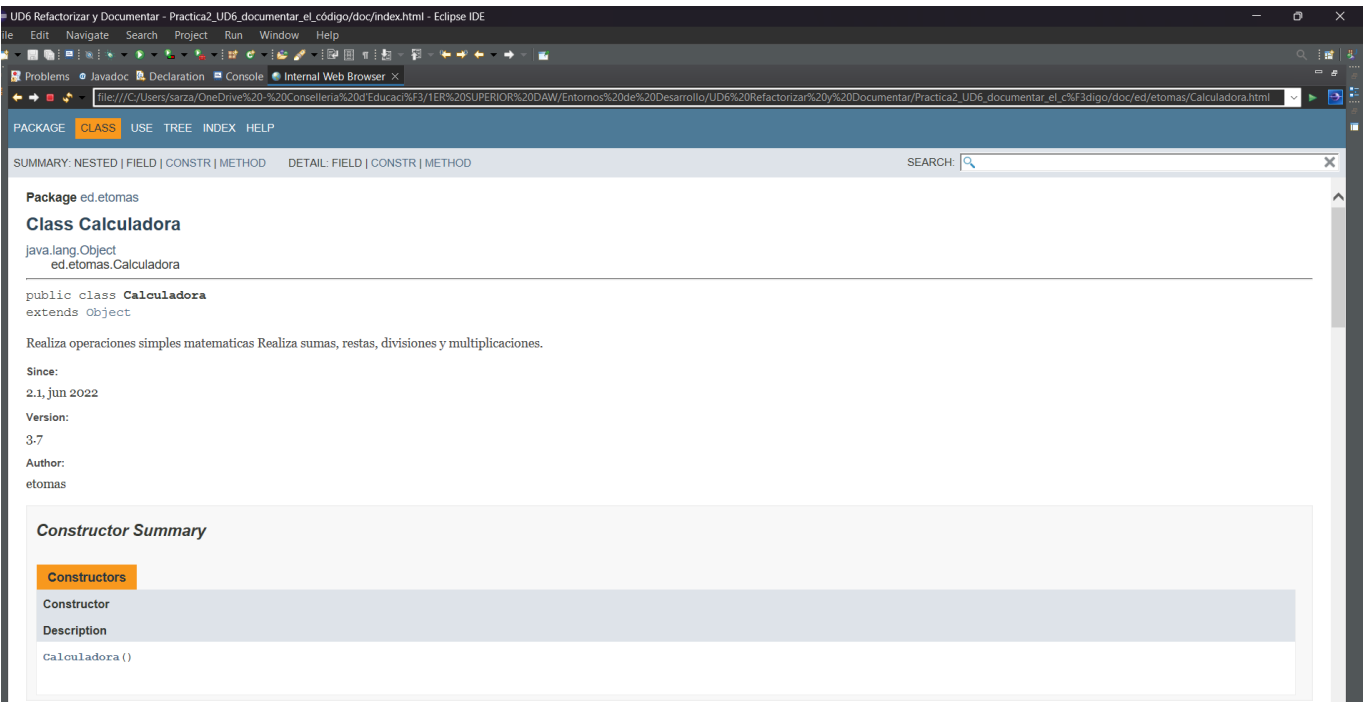
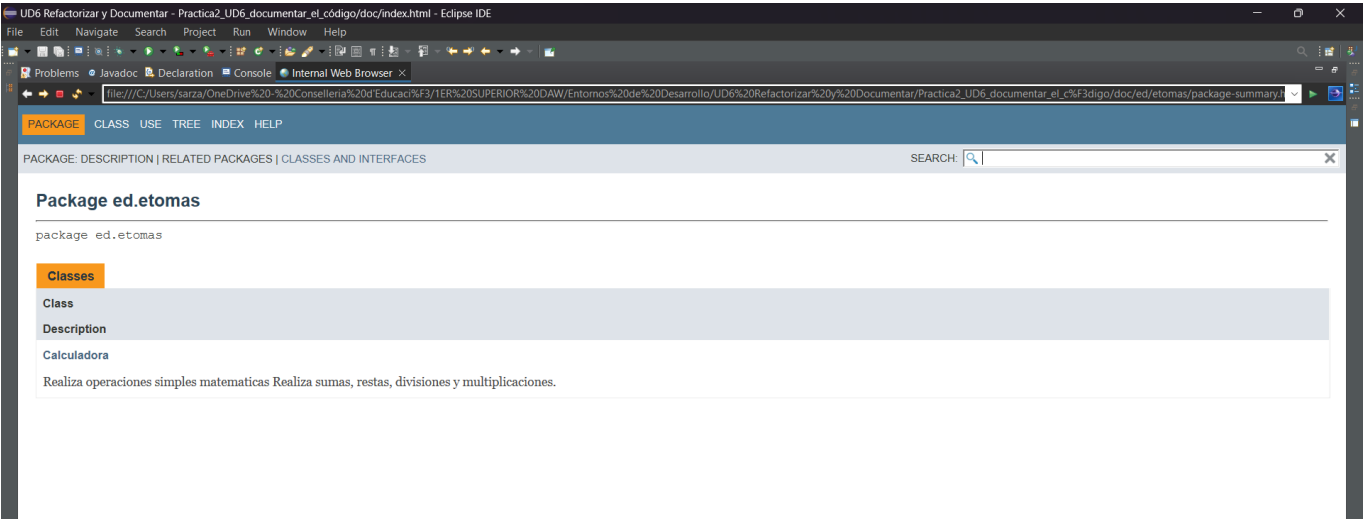
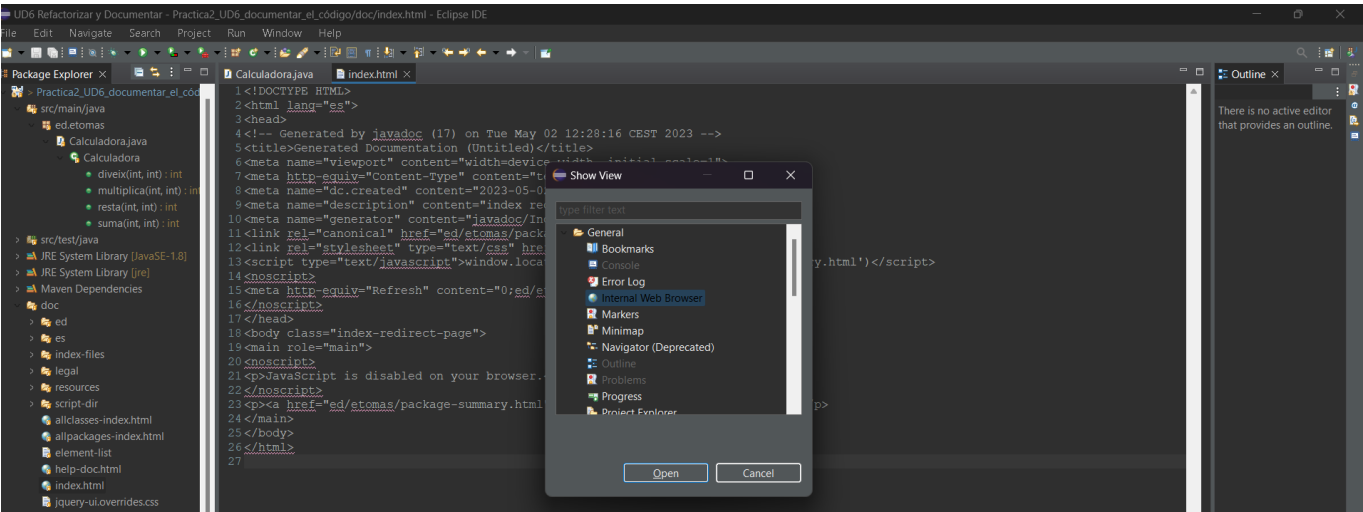
```

Calculadora.java X
1 package ed.etomas;
2
3 /**
4  * Realiza operaciones simples matematicas
5  * Realiza sumas, restas, divisiones y multiplicaciones.
6  * @author etomas
7  * @version 3.7
8  * @since 2.1, jun 2022
9  */
10
11
12 public class Calculadora {
13
14
15     /**
16      * Suma de @param a y @param b
17      * Realiza la suma de 2 numeros
18      * @param a numero entero que actua como sumando
19      * @param b numero entero que actua como sumando
20      * @return valor int resultado de sumar a y b
21      */
22
23
24     public int suma(int a, int b) {
25         return a + b;
26     }
27
28     /**
29      * Resta de @param a y @param b
30      * Realiza la resta de 2 numeros
31      * @param a numero entero que actua como minuendo
32      * @param b numero entero que actua como sustraendo
33      * @return diferencia valor int resultado de restar a y b
34      */
35
36     public int resta(int a, int b) {
37         return a - b;
38     }
39
40     /**
41      * Multiplicacion de @param a y @param b
42      * Realiza la Multiplicacion de 2 numeros
43      * @param a numero entero que actua como multiplicando
44      * @param b numero entero que actua como multiplicador
45      * @return valor int resultado de multiplicar a y b
46      */
47
48     public int multiplica(int a, int b){

```

5. Muestra el resultado de la documentación:

Muestra el fichero index.html en el Navegador integrado en Eclipse y muestra capturas de TODO su contenido. Fijate en la correspondencia entre las descripciones y tags de las páginas web (html) generadas.



Method Summary

- All Methods
- Instance Methods
- Concrete Methods

Modifier and Type
Method
Description
int
divideix(int a, int b)
Division de @param a y @param b Realiza la division de 2 numeros
int
multiplica(int a, int b)
Multiplicacion de @param a y @param b Realiza la Multiplicacion de 2 numeros
int
resta(int a, int b)
Resta de @param a y @param b Realiza la resta de 2 numeros
int
suma(int a, int b)
Suma de @param a y @param b Realiza la suma de 2 numeros

Methods inherited from class java.lang.Object

equals , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Constructor Details

Calculadora

public Calculadora()

Method Details

suma
public int suma(int a, int b)
Suma de @param a y @param b Realiza la suma de 2 numeros
Parameters:
a - numero entero que actua como sumando
b - numero entero que actua como sumando
Returns:
valor int resultado de sumar a y b

resta
public int resta(int a, int b)
Resta de @param a y @param b Realiza la resta de 2 numeros
Parameters:
a - numero entero que actua como minuendo
b - numero entero que actua como sustraendo
Returns:
diferencia valor int resultado de restar a y b

multiplica
public int multiplica(int a, int b)
Multiplicacion de @param a y @param b Realiza la Multiplicacion de 2 numeros
Parameters:
a - numero entero que actua como multiplicando
b - numero entero que actua como multiplicador
Returns:
valor int resultado de multiplicar a y b

## diveix

```
public int diveix(int a,  
                 int b)
```

Division de @param a y @param b Realiza la division de 2 numeros

### Parameters:

a - numero entero que actua como dividendo

b - numero entero que actua como divisor

### Returns:

valor int resultado de dividir a y b

### Throws:

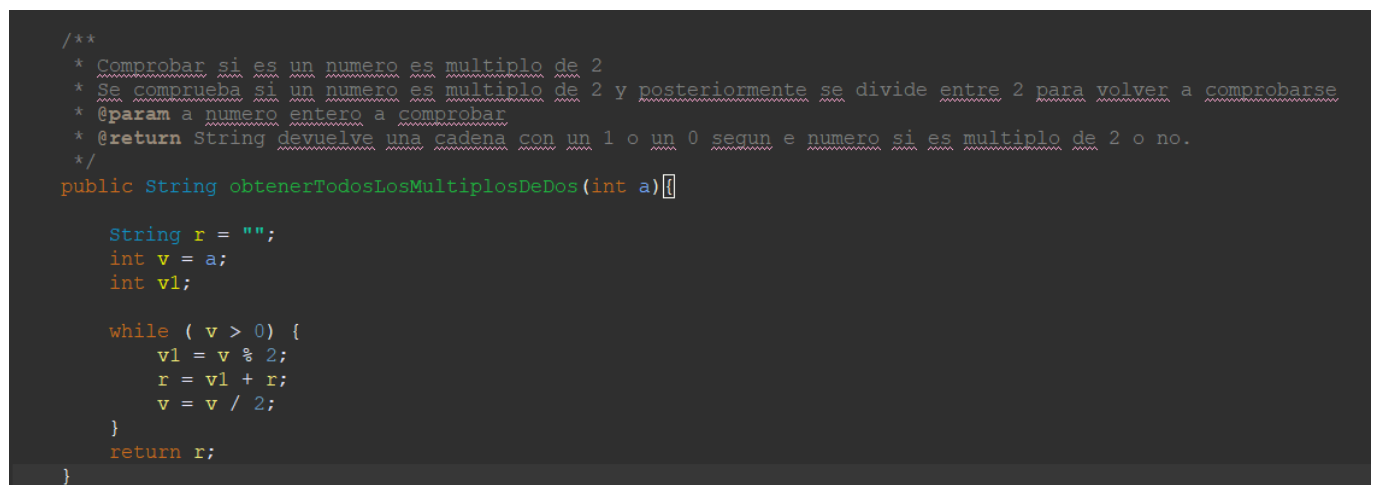
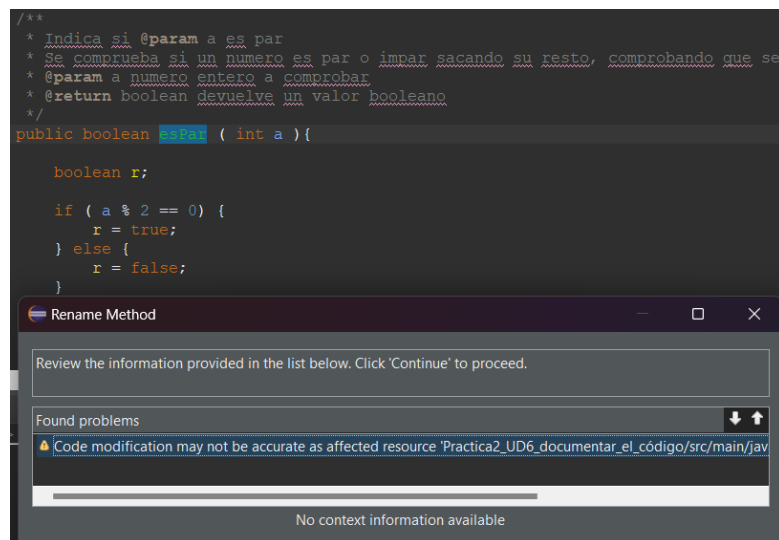
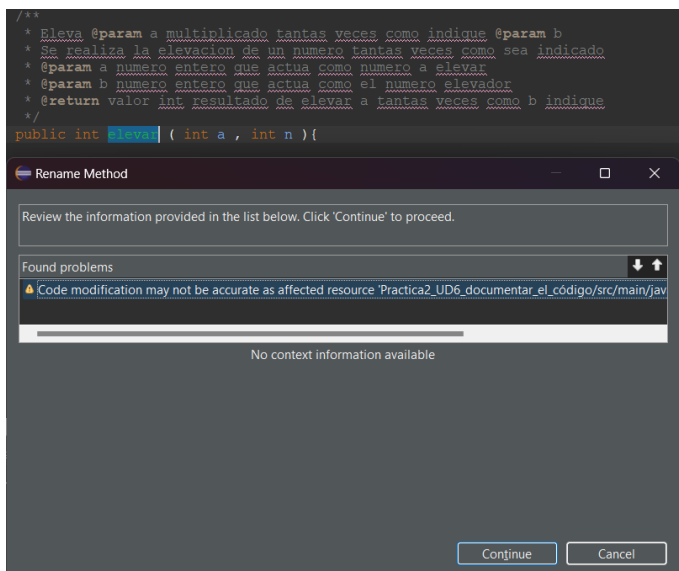
ArithmeticException - division por 0

## 6. Añade 3 nuevos métodos a la clase Calculadora

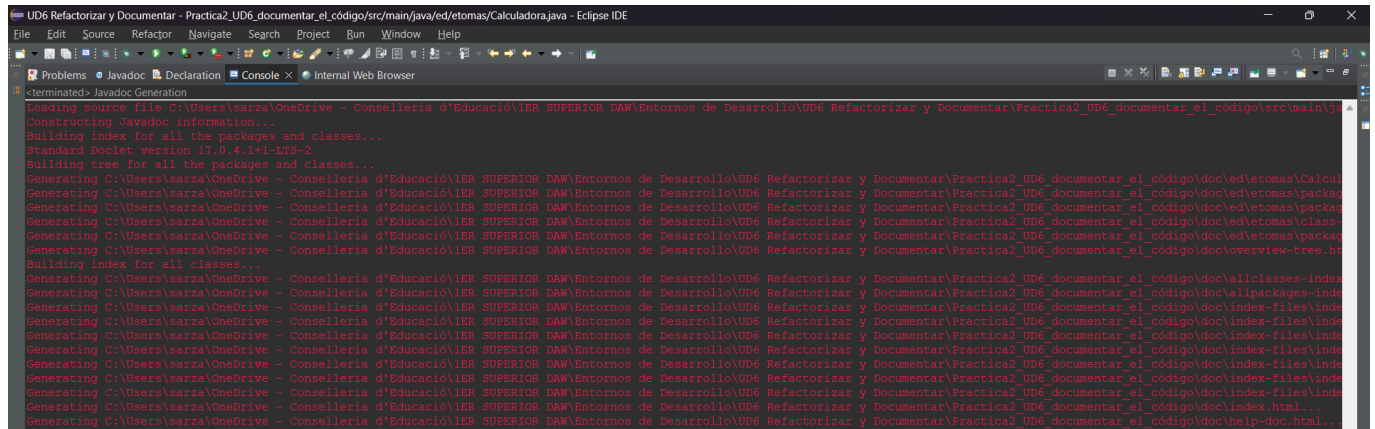
Investiga que cálculo realiza cada uno (ayudate del debugger).

Cambia el nombre de los métodos y sus variables para que correspondan a un nombre más representativo de acuerdo con su funcionamiento(Usa Refactoring -> Rename)

Documenta los nuevos métodos adecuadamente con comentarios JavaDoc.



## Genera la documentación en Javadoc



Muestra capturas de:

código documentado (fomatea el código si es necesario)

```
* Eleva @param a multiplicado tantas veces como indique @param b
* Se realiza la elevación de un número tantas veces como sea indicado
* @param a número entero que actúa como número a elevar
* @param n número entero que actúa como el número elevador
* @return valor int resultado de elevar a tantas veces como b indique
*/
public int elevar ( int a , int n ){

    int t = 1;

    for ( int i = 0; i < n ; i++ ){
        t = t * a;
    }

    return t;
}

/**
 * Indica si @param a es par
 * Se comprueba si un número es par o impar sacando su resto, comprobando que sea igual que 0
 * @param a número entero a comprobar
 * @return boolean devuelve un valor booleano
 */
public boolean esPar ( int a ){

    boolean r;

    if ( a % 2 == 0 ) {
        r = true;
    } else {
        r = false;
    }

    return r;
}
```

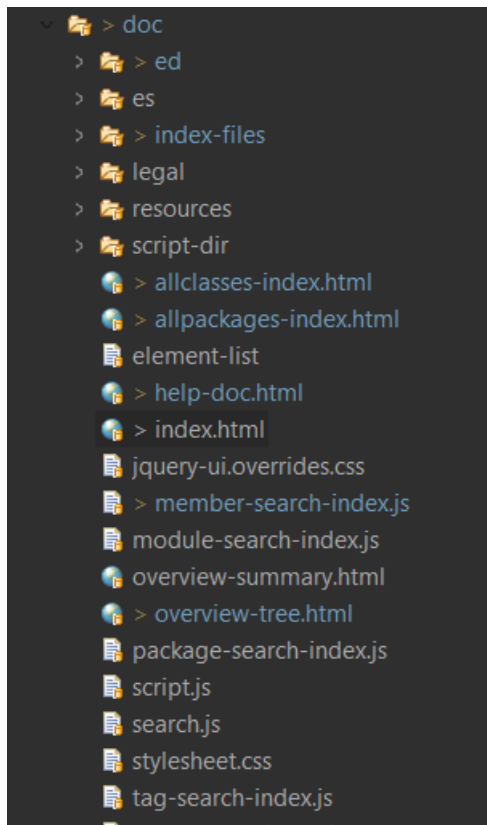
```
/**
 * Comprobar si es un número es múltiplo de 2
 * Se comprueba si un número es múltiplo de 2 y posteriormente se divide entre 2 para volver a comprobarse
 * @param a número entero a comprobar
 * @return String devuelve una cadena con un 1 o un 0 según el número si es múltiplo de 2 o no.
 */
public String obtenerTodosLosMultiplosDeDos(int a){

    String r = "";
    int v = a;
    int v1;

    while ( v > 0 ) {
        v1 = v % 2;
        r = v1 + r;
        v = v / 2;
    }
    return r;
}
```



## Directorio Doc y la documentación resultante en el navegador



### eleva

```
public int eleva(int a,  
                int n)
```

Eleva @param a multiplicado tantas veces como indique @param b Se realiza la elevacion de un numero tantas veces como sea indicado

#### Parameters:

a - numero entero que actua como numero a elevar

n - numero entero que actua como el numero elevador

#### Returns:

valor int resultado de elevar a tantas veces como b indique

### esPar

```
public boolean esPar(int a)
```

Indica si @param a es par Se comprueba si un numero es par o impar sacando su resto, comprobando que sea igual que 0

#### Parameters:

a - numero entero a comprobar

#### Returns:

boolean devuelve un valor booleano

### obtenerTodosLosMultiplosDeDos

```
public String obtenerTodosLosMultiplosDeDos(int a)
```

Comprobar si es un numero es multiplo de 2 Se comprueba si un numero es multiplo de 2 y posteriormente se divide entre 2 para volver a comprobarse

#### Parameters:

a - numero entero a comprobar

#### Returns:

String devuelve una cadena con un 1 o un 0 segun e numero si es multiplo de 2 o no.