



CICLO: [DAM]
MÓDULO DE [PROGRAMACIÓN]

[Tarea N° 03]

Alumno:
[Juan Carlos Filter Martín]
[15456141A]

Contenido

1. Documentos que se adjuntan a este informe.....	3
2. RA03_a) Se ha escrito y probado código que haga uso de estructuras de selección.....	3
Crear un proyecto en Java en NetBeans que se denomine "potencia"	3
3. RA03_b) Se han utilizado estructuras de repetición.....	4
Dentro del proyecto crear una clase "calcularPotencia.java" con un método main.....	4
Declarar el siguiente conjunto de variables	6
1. Declarar e inicializar una variable base. Almacenará el valor del número al cual se desea calcular la potencia. (valor entero).....	6
2. Declarar e inicializar una variable exponente. Almacenará el valor numérico al que se desea elevar la base. (valor entero).....	6
3. Declarar una variable resultado. Almacenará el valor numérico que resulta de elevar la base al exponente. (valor entero).....	7
4. RA03_e) Se han creado programas ejecutables utilizando diferentes estructuras de control....	7
Dentro del cálculo de la operación matemática potencia se distinguirán los siguientes casos:.....	7
1. Variable exponente sea inicializada a 0, en cuyo caso el resultado será 1.....	7
2. Variable base sea inicializada a 1, en cuyo caso el resultado será 1.....	7
3. Que el base sea mayor que 1 y que el exponente sea mayor que 0, en cuyo caso se utilizará una estructura iterativa para realizar el cálculo de la potencia.....	8
4. Mostrar el resultado por pantalla.....	10
5. RA03_d) Se ha escrito código utilizando el control de excepciones.....	11
Todas las operaciones del programa deben de ir dentro de un control de excepciones que no permita que ocurran errores descontrolados.....	11
6. RA03_h) Se han creado excepciones.....	12
Si la base es negativa o el exponente es negativo generaremos una excepción avisando de que los datos deben ser 0 o mayor que 0.....	12
7. RA03_g) Se ha comentado y documentado el código.....	13
Capturas que muestren el resultado de la ejecución del programa en la documentación.....	13
8. RA03_f) Se han probado y depurado los programas.....	14
Hacer pruebas para varias combinaciones de valores de entrada.....	14
1. Base: 0 Exponente:10.....	14
2. Base: 10 Exponente: 0.....	15
3. Base: 1 Exponente: 20.....	16
4. Base: 5 Exponente: 3.....	17

1. Documentos que se adjuntan a este informe.

A continuación se detallan los documentos que componen la presente entrega de la tarea:

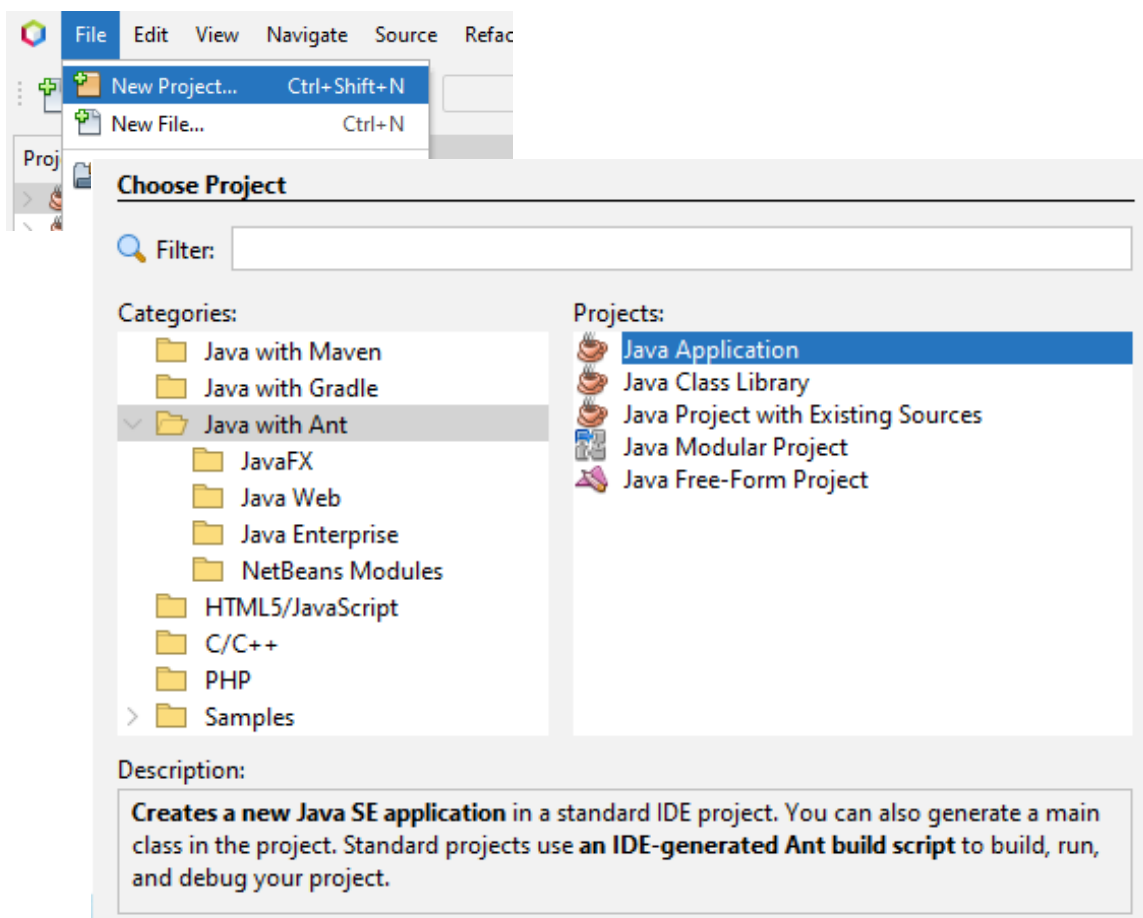
1. Informe de elaboración de la tarea.
2. Proyecto Java

2. RA03_a) Se ha escrito y probado código que haga uso de estructuras de selección.

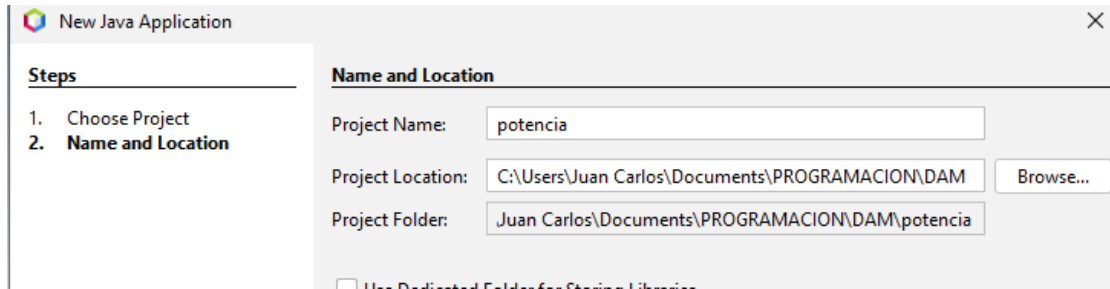
Crear un proyecto en Java en NetBeans que se denomine "potencia".

Para ello vamos a:

File > New Project > Java with Ant > Java Application



Le asignamos un nombre al proyecto : “potencia”



Ya tendríamos creado el proyecto potencia.

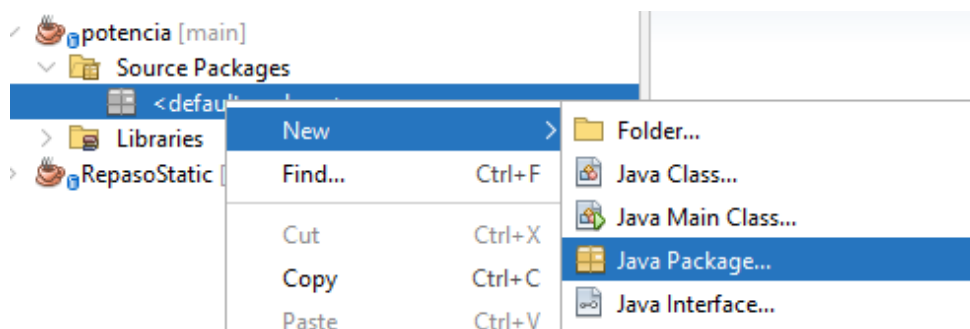
3. RA03_b) Se han utilizado estructuras de repetición.

Dentro del proyecto crear una clase "calcularPotencia.java" con un método main.

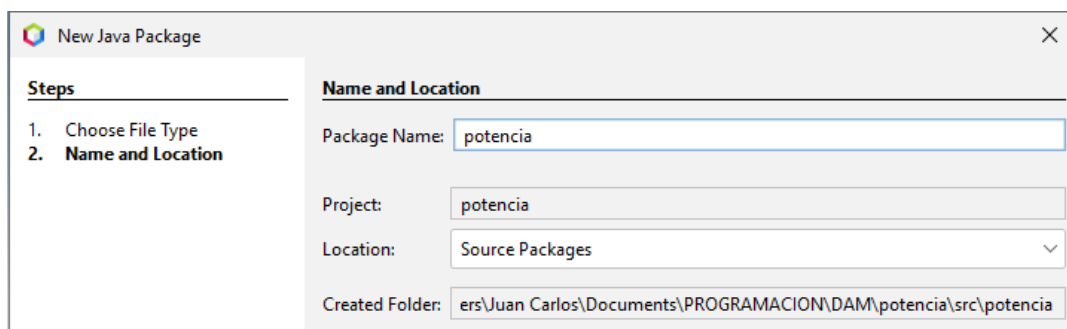
Primero hay que crear un Package que va a contener la clase del proyecto y para ello vamos a:

Click derecho en el Package vacío dentro del proyecto

New > Java Package



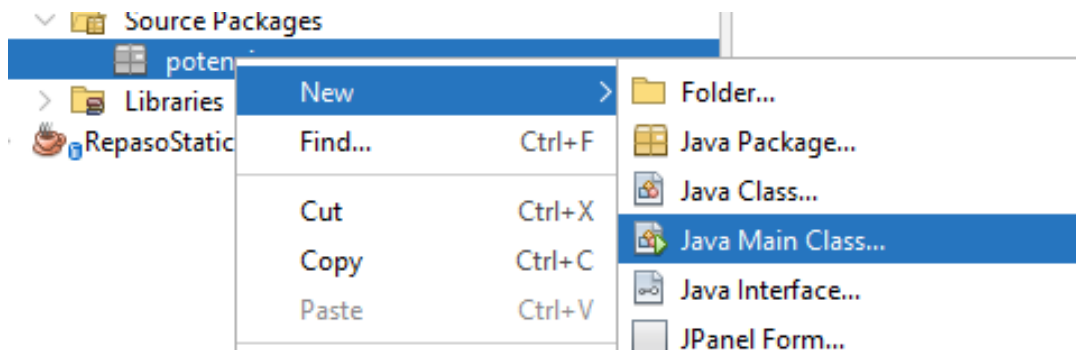
Ponemos el nombre del paquete que puede ser el del proyecto “potencia”



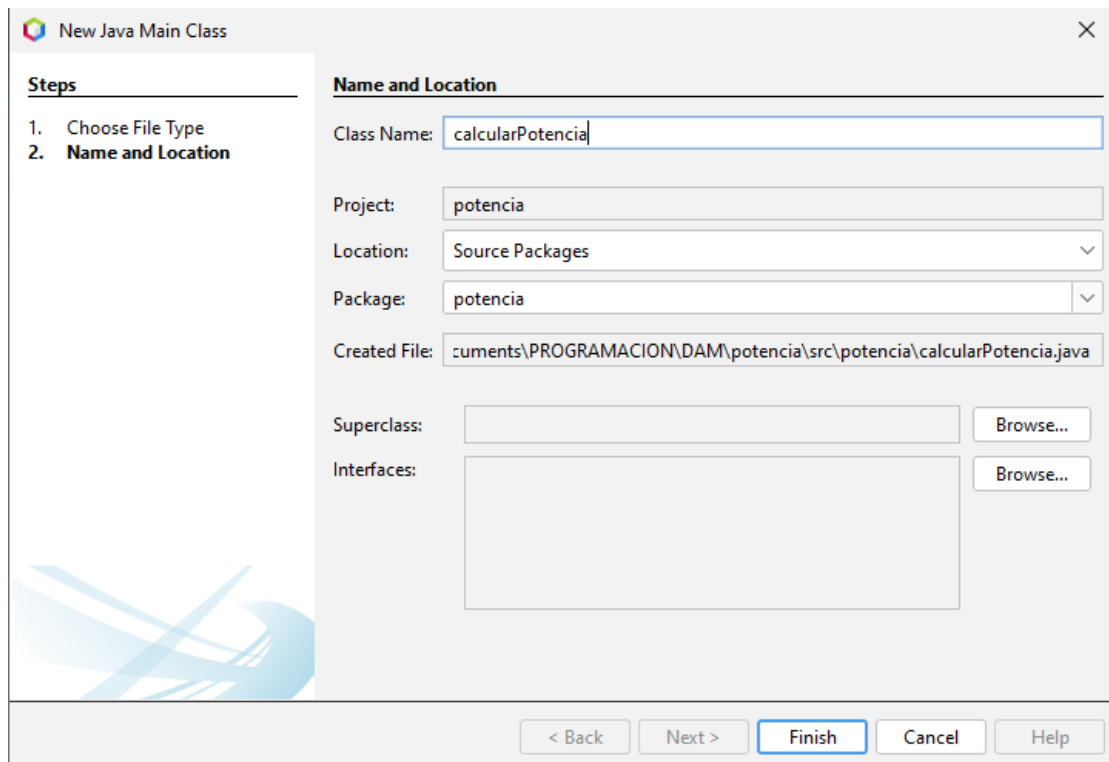
Una vez creado el Package toca crear la clase main:

Click derecho en el Package

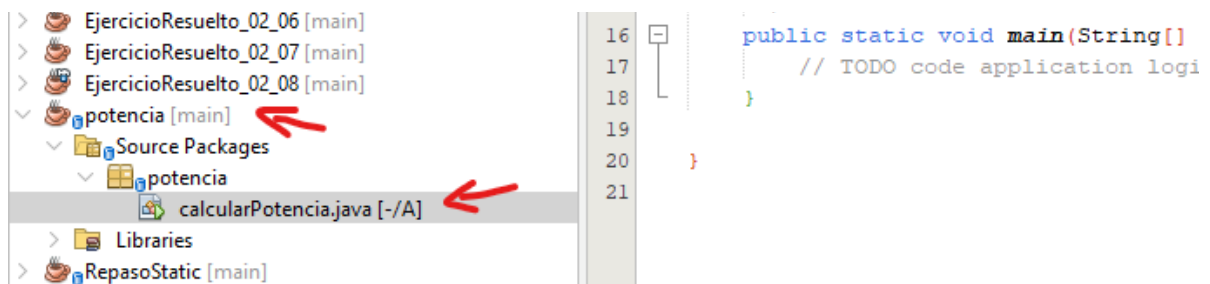
New > Java Main Class



Ponemos el nombre a la clase main calcularPotencia

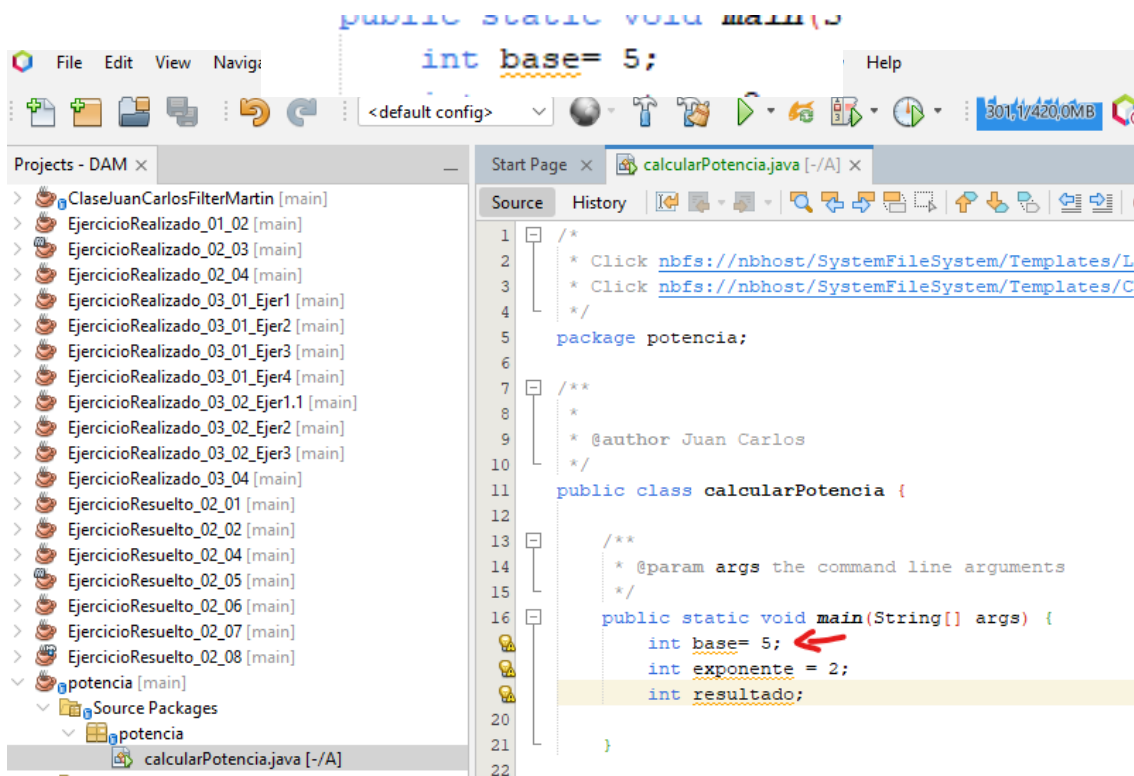


Y ya tendríamos nuestra clase main calcularPotencia.java creada



Declarar el siguiente conjunto de variables .

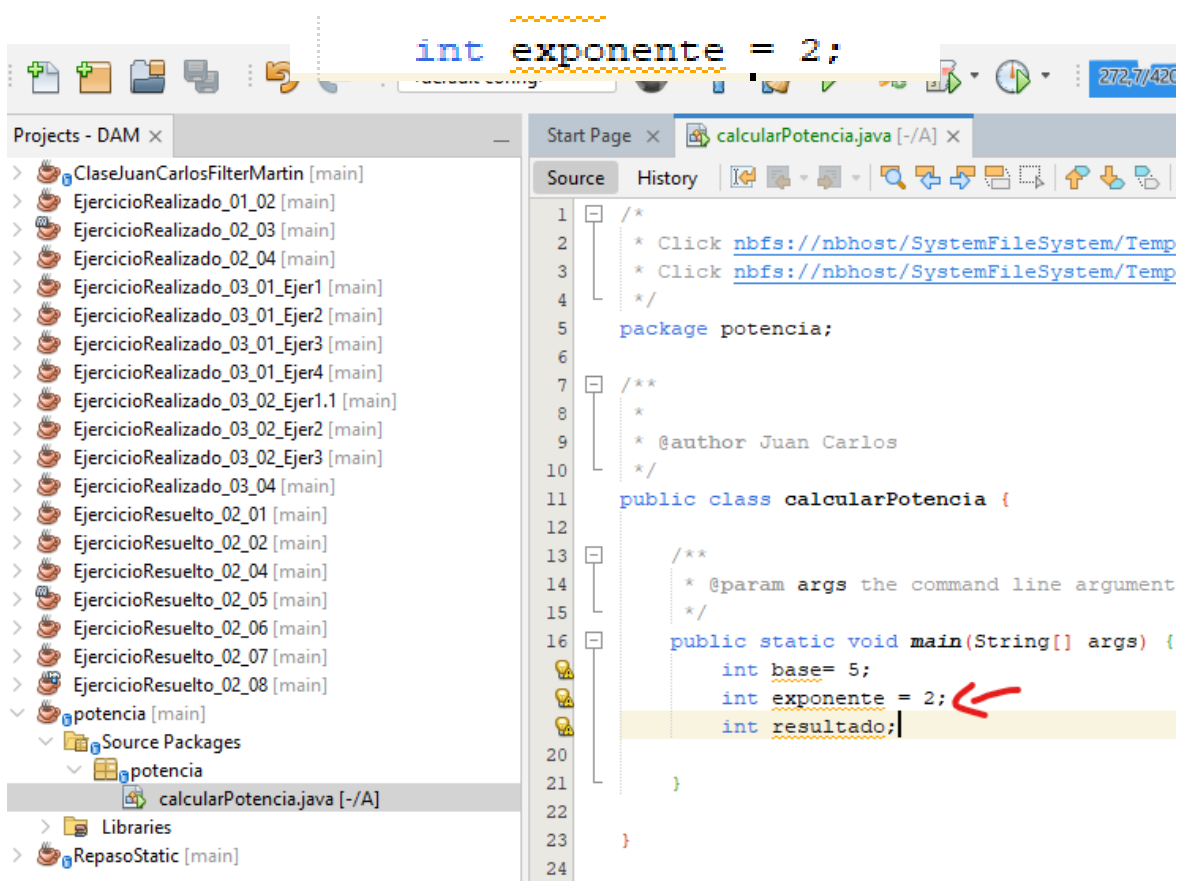
1. Declarar e inicializar una variable base. Almacenará el valor del número al cual se desea calcular la potencia. (valor entero).



The screenshot shows an IDE with the 'calcularPotencia.java' file open. The 'main' method is visible, and the variable 'base' is being declared and initialized with the value 5. A red arrow points to the line 'int base= 5;'. The project structure on the left shows a package named 'potencia'.

```
public static void main()  
{  
    int base= 5;  
    // ...  
}
```

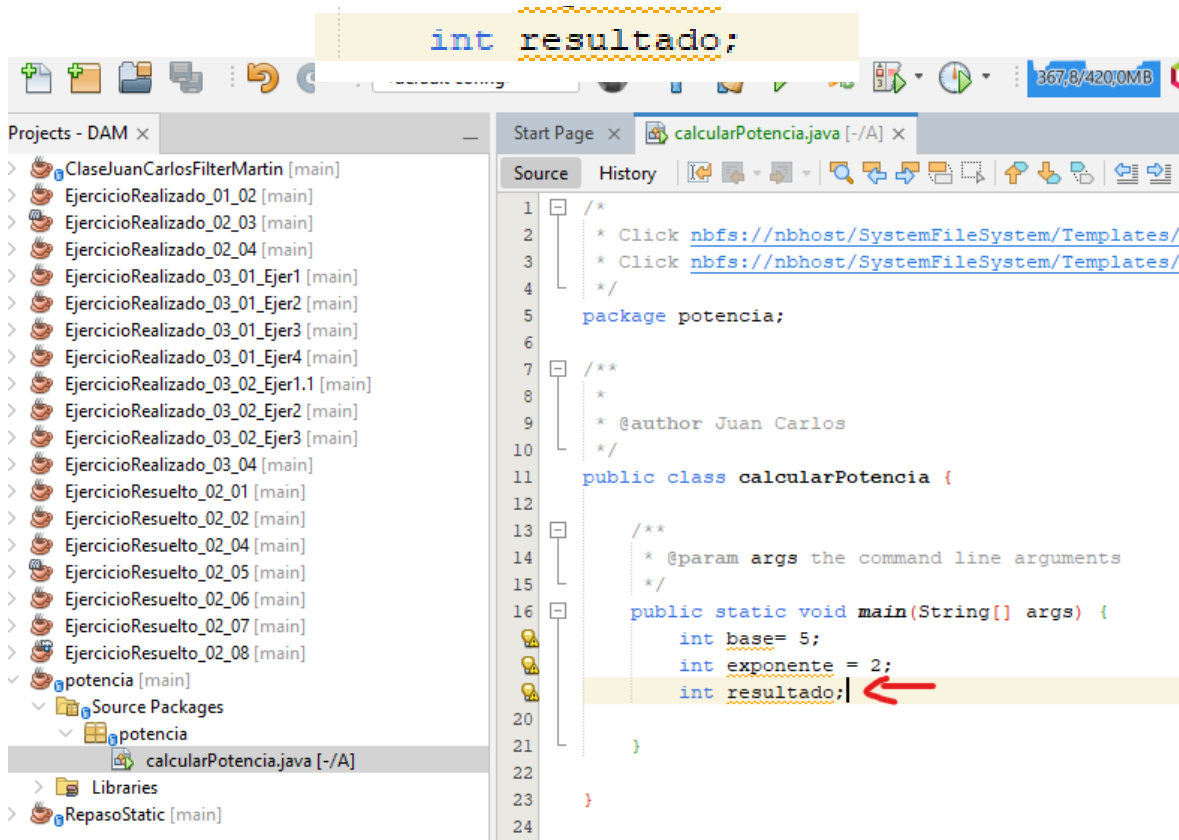
2. Declarar e inicializar una variable exponente. Almacenará el valor numérico al que se desea elevar la base. (valor entero).



The screenshot shows the same IDE with the 'calcularPotencia.java' file open. The 'main' method is visible, and the variable 'exponente' is being declared and initialized with the value 2. A red arrow points to the line 'int exponente = 2;'. The project structure on the left shows a package named 'potencia'.

```
int exponente = 2;  
// ...  
int resultado;
```

3. Declarar una variable resultado. Almacenará el valor numérico que resulta de elevar la base al exponente. (valor entero).



4. RA03_e) Se han creado programas ejecutables utilizando diferentes estructuras de control.

Dentro del cálculo de la operación matemática potencia se distinguirán los siguientes casos:

1. Variable exponente sea inicializada a 0, en cuyo caso el resultado será 1.

Se crea un if diciéndole que si exponente es igual a 0 entonces muestre el resultado a 1

```
}else if (exponente == 0) { //Si expon  
    resultado = 1;
```

2. Variable base sea inicializada a 1, en cuyo caso el resultado será 1.

Se crea un if diciéndole que si base es igual a 1 entonces cambie esa variable base a 1.

```
} else if (base == 1) { //Si base = 1. resultado vale 1 |  
    resultado = 1;
```

3. Que el base sea mayor que 1 y que el exponente sea mayor que 0, en cuyo caso se utilizará una estructura iterativa para realizar el cálculo de la potencia

→ Se ha creado un método en otra clase que calcule la potencia mediante un for, inicializamos la variable resultado en 1 (para que se pueda multiplicar por 1) y entramos en el for.

En el for creamos la variable i y la iniciamos en 1.

Mientras i sea <= exponente (1 <= 2).... entonces:

- Resultado * base (1 * 5) = 5

Ahora resultado vale 5 y volvemos al for incrementando i++:

Mientras i sea <= exponente (2 <= 2).... entonces:

- Resultado * base (5 * 5) = 25

Ahora resultado vale 25 y volvemos al for incrementando i++:

Mientras i sea <= exponente (3 <= 2).... entonces:

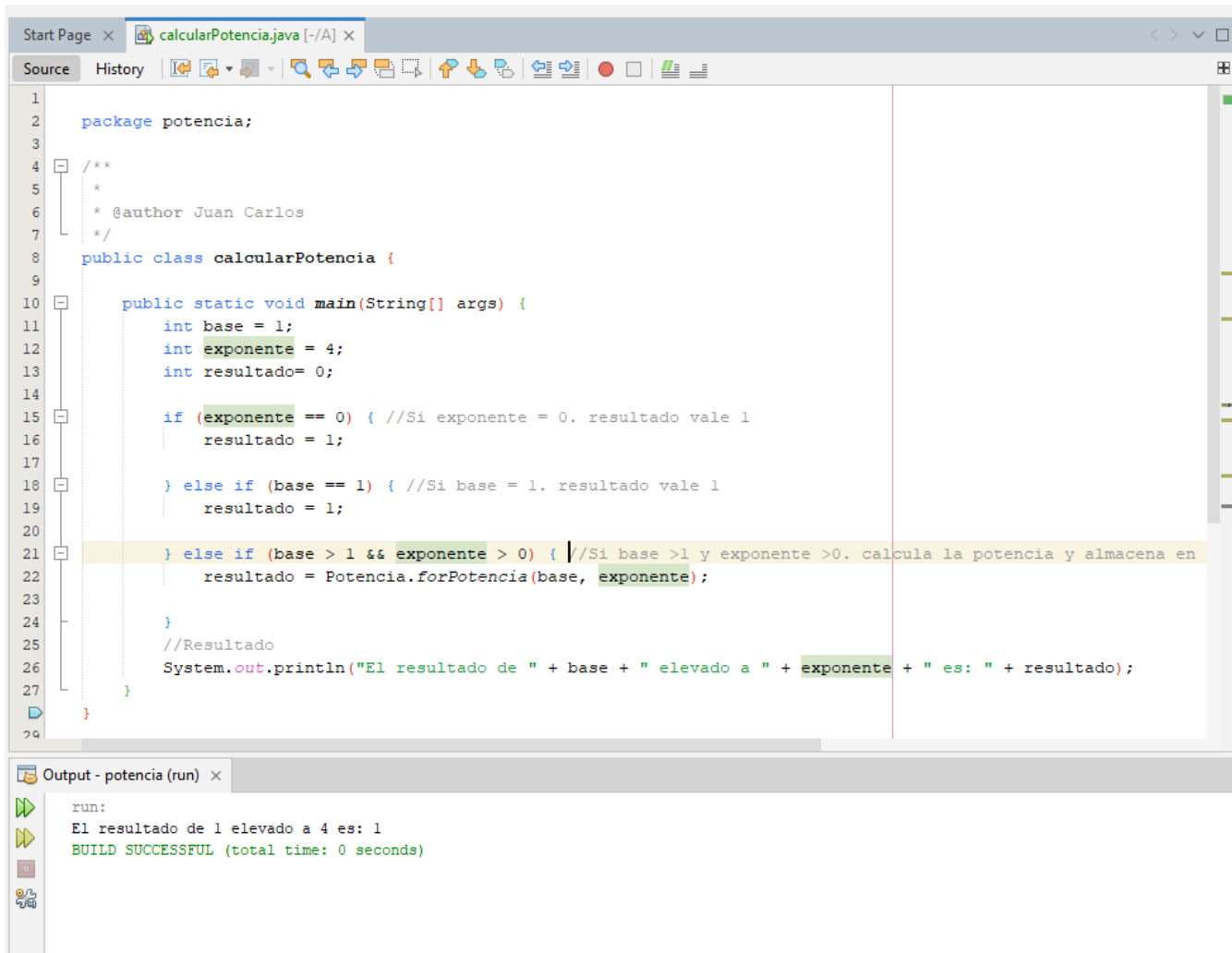
No puede entrar en el bucle for porque i es > exponente entonces sale y imprime por pantalla el resultado (25) con el System.out.print

```
public class Potencia {  
    public static int forPotencia(int base, int exponente) {  
        int resultado = 1;  
        for (int i = 1; i <= exponente; i++) {  
            resultado = resultado * base;  
        }  
        return resultado;  
    }  
}
```

→ Ahora vamos a la clase main y llamamos a ese método dentro de un if indicando que si la base es > 1 y el exponente es >0 entonces

```
    } else if (base > 1 && exponente > 0) { //Si base >1 y expone  
        resultado = Potencia.forPotencia(base, exponente);  
    }
```


Clase main



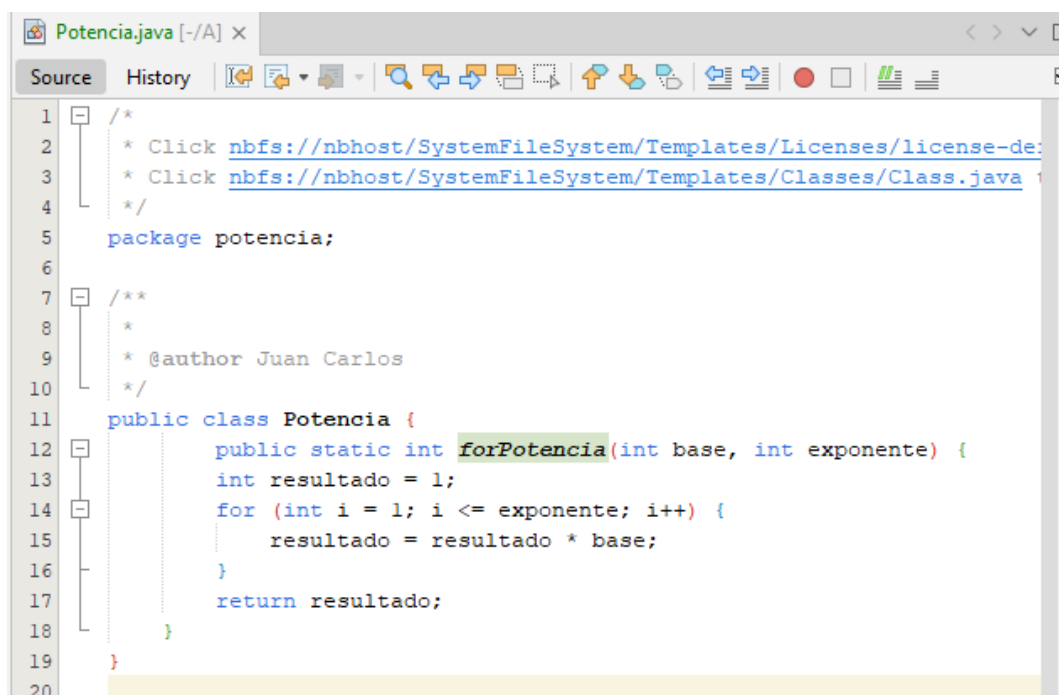
The screenshot shows an IDE window with the file 'calcularPotencia.java' open. The code defines a package 'potencia' and a public class 'calcularPotencia'. Inside the class, there is a static method 'main' that takes an array of strings 'args'. The method initializes 'base' to 1, 'exponente' to 4, and 'resultado' to 0. It then uses a series of conditional statements to calculate the power. If 'exponente' is 0, 'resultado' is set to 1. If 'base' is 1, 'resultado' is set to 1. If both 'base' is greater than 1 and 'exponente' is greater than 0, it calls the static method 'Potencia.forPotencia' with 'base' and 'exponente' as arguments. Finally, it prints the result using 'System.out.println'. The output window at the bottom shows the execution result: 'El resultado de 1 elevado a 4 es: 1' and 'BUILD SUCCESSFUL (total time: 0 seconds)'.

```
1 package potencia;
2
3
4 /**
5  *
6  * @author Juan Carlos
7  */
8 public class calcularPotencia {
9
10     public static void main(String[] args) {
11         int base = 1;
12         int exponente = 4;
13         int resultado = 0;
14
15         if (exponente == 0) { //Si exponente = 0. resultado vale 1
16             resultado = 1;
17
18         } else if (base == 1) { //Si base = 1. resultado vale 1
19             resultado = 1;
20
21         } else if (base > 1 && exponente > 0) { //Si base > 1 y exponente > 0. calcula la potencia y almacena en
22             resultado = Potencia.forPotencia(base, exponente);
23
24         }
25         //Resultado
26         System.out.println("El resultado de " + base + " elevado a " + exponente + " es: " + resultado);
27     }
28 }
```

Output - potencia (run) x

run:
El resultado de 1 elevado a 4 es: 1
BUILD SUCCESSFUL (total time: 0 seconds)

Clase con método para calcular la potencia



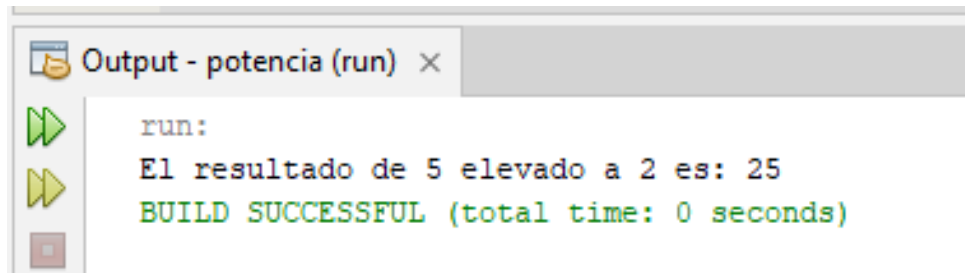
The screenshot shows an IDE window with the file 'Potencia.java' open. The code defines a package 'potencia' and a public class 'Potencia'. Inside the class, there is a static method 'forPotencia' that takes two integers, 'base' and 'exponente', as parameters. The method initializes 'resultado' to 1 and then uses a 'for' loop to calculate the power by multiplying 'base' by itself 'exponente' times. Finally, it returns the result.

```
1 /**
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-de
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java
4  */
5 package potencia;
6
7 /**
8  *
9  * @author Juan Carlos
10  */
11 public class Potencia {
12     public static int forPotencia(int base, int exponente) {
13         int resultado = 1;
14         for (int i = 1; i <= exponente; i++) {
15             resultado = resultado * base;
16         }
17         return resultado;
18     }
19 }
20
```

4. Mostrar el resultado por pantalla.

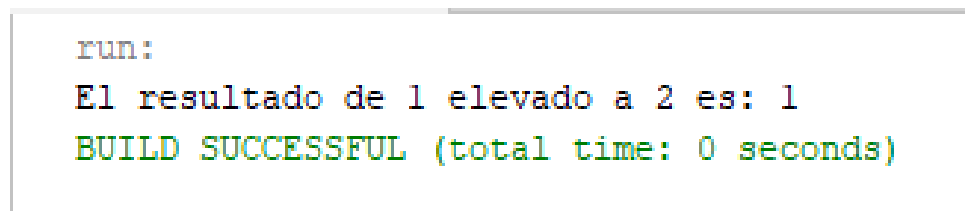
Varias capturas de pantalla con diferentes resultados

Base 5 , exponente 2.



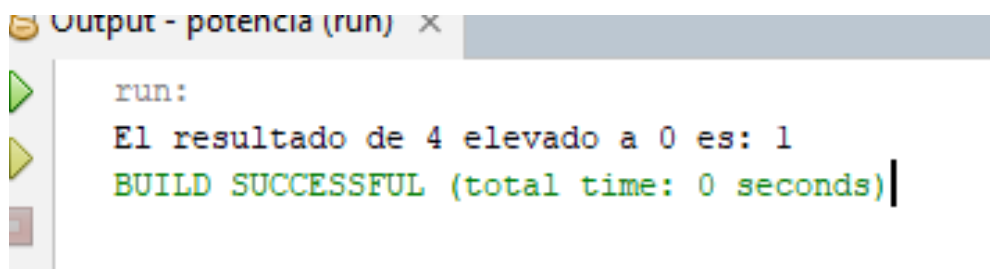
```
Output - potencia (run) x
run:
El resultado de 5 elevado a 2 es: 25
BUILD SUCCESSFUL (total time: 0 seconds)
```

Base 1 , exponente 2.



```
run:
El resultado de 1 elevado a 2 es: 1
BUILD SUCCESSFUL (total time: 0 seconds)
```

Base 4 , exponente 0.



```
Output - potencia (run) x
run:
El resultado de 4 elevado a 0 es: 1
BUILD SUCCESSFUL (total time: 0 seconds)
```

5. RA03_d) Se ha escrito código utilizando el control de excepciones.

Todas las operaciones del programa deben de ir dentro de un control de excepciones que no permita que ocurran errores descontrolados.

(Si sucede un error, simplemente daremos un mensaje indicando qué error se ha producido y el programa finalizará)

```
8 public class calcularPotencia {
9
10     public static void main(String[] args) {
11         int base = 1;
12         int exponente = 4;
13         int resultado = 0;
14
15         try{
16             if (exponente == 0) { //Si exponente = 0. resultado vale 1
17                 resultado = 1;
18
19             } else if (base == 1) { //Si base = 1. resultado vale 1
20                 resultado = 1;
21
22             } else if (base > 1 && exponente > 0) { //Si base >1 y exponente >0. calcula la potencia y almacena en
23                 resultado = Potencia.forPotencia(base, exponente);
24             }
25             //Resultado
26             System.out.println("El resultado de " + base + " elevado a " + exponente + " es: " + resultado);
27
28         } catch (Exception e) {
29             System.out.println("Se ha producido un error: " + e.getMessage());
30         }
31     }
32 }
33
```

6. RA03_h) Se han creado excepciones.

Si la base es negativa o el exponente es negativo generaremos una excepción avisando de que los datos deben ser 0 o mayor que 0.

Para ello se ha creado un `if` diciendo que si `base < 0` **O** `exponente < 0`

Entonces mediante un **`throw new`** indicándole **que** agregue la excepción

(Se ha colocado el `if` en primer lugar ya que si no fuera de esta forma podría entrar antes en otra condición si por ejemplo se le dice que `base = -1` y `exponente = 0`)

```
public class calcularPotencia {  
  
    public static void main(String[] args) {  
  
        //Variables  
        int base = 5;  
        int exponente = 3;  
        int resultado = 0;  
  
        //Control de excepciones mediante un try - catch  
        try {  
            if (base < 0 || exponente < 0) {  
                throw new ArithmeticException("La base y el exponente deben ser igual o mayor a 0");  
            } else if (exponente == 0) { //Si exponente = 0. resultado vale 1  
                resultado = 1;  
            } else if (base == 1) { //Si base = 1. resultado vale 1  
                resultado = 1;  
            } else if (base > 1 && exponente > 0) { //Si base > 1 y exponente > 0. calcula la potencia y almacena e  
                resultado = Potencia.forPotencia(base, exponente);  
            }  
  
            //Resultado  
            System.out.println("El resultado de " + base + " elevado a " + exponente + " es: " + resultado);  
  
            //excepcion para controlar si es igual o mayor que 0  
        } catch (ArithmeticException a) {  
            System.out.println("Se ha producido un error: " + a.getMessage());  
            //excepciones para otros posibles errores en el programa  
        } catch (Exception e) {  
            System.out.println("Se ha producido un error: " + e.getMessage());  
        }  
    }  
}
```

Y al final se ha quedado con 2 catch:

→ El primer catch para controlar la excepción de que no sea menor o igual que 0

→ Segundo catch mas general para controlar el resto del programa

```
        //Variables  
        int base = -1;  
        int exponente = 3;  
        int resultado = 0;
```

Output - potencia (run) x

run:
Se ha producido un error: La base y el exponente deben ser igual o mayor a 0
BUILD SUCCESSFUL (total time: 0 seconds)

7. RA03_g) Se ha comentado y documentado el código.

Capturas que muestren el resultado de la ejecución del programa en la documentación.

Se van a dejar resultados sobre el control de excepciones

Error base = -1

```
public static void main(String[] args) {  
    int base = -1;  
    int exponente = 0;  
    int resultado = 0;
```

run:

Se ha producido un error: La base y el exponente deben ser igual o mayor a 0
BUILD SUCCESSFUL (total time: 0 seconds)

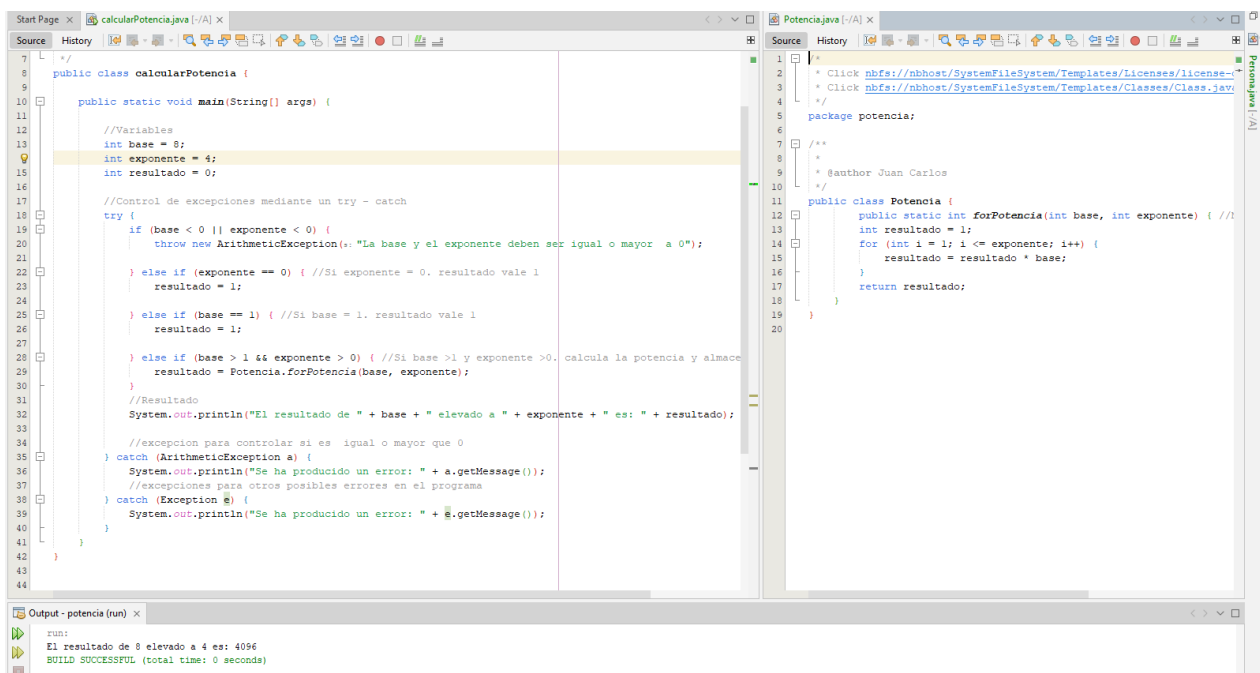
Error exponente = -1

```
public static void main(String[] args) {  
    int base = 5;  
    int exponente = -1;  
    int resultado = 0;
```

run:

Se ha producido un error: La base y el exponente deben ser igual o mayor a 0
BUILD SUCCESSFUL (total time: 0 seconds)

Resultado de todo el programa:

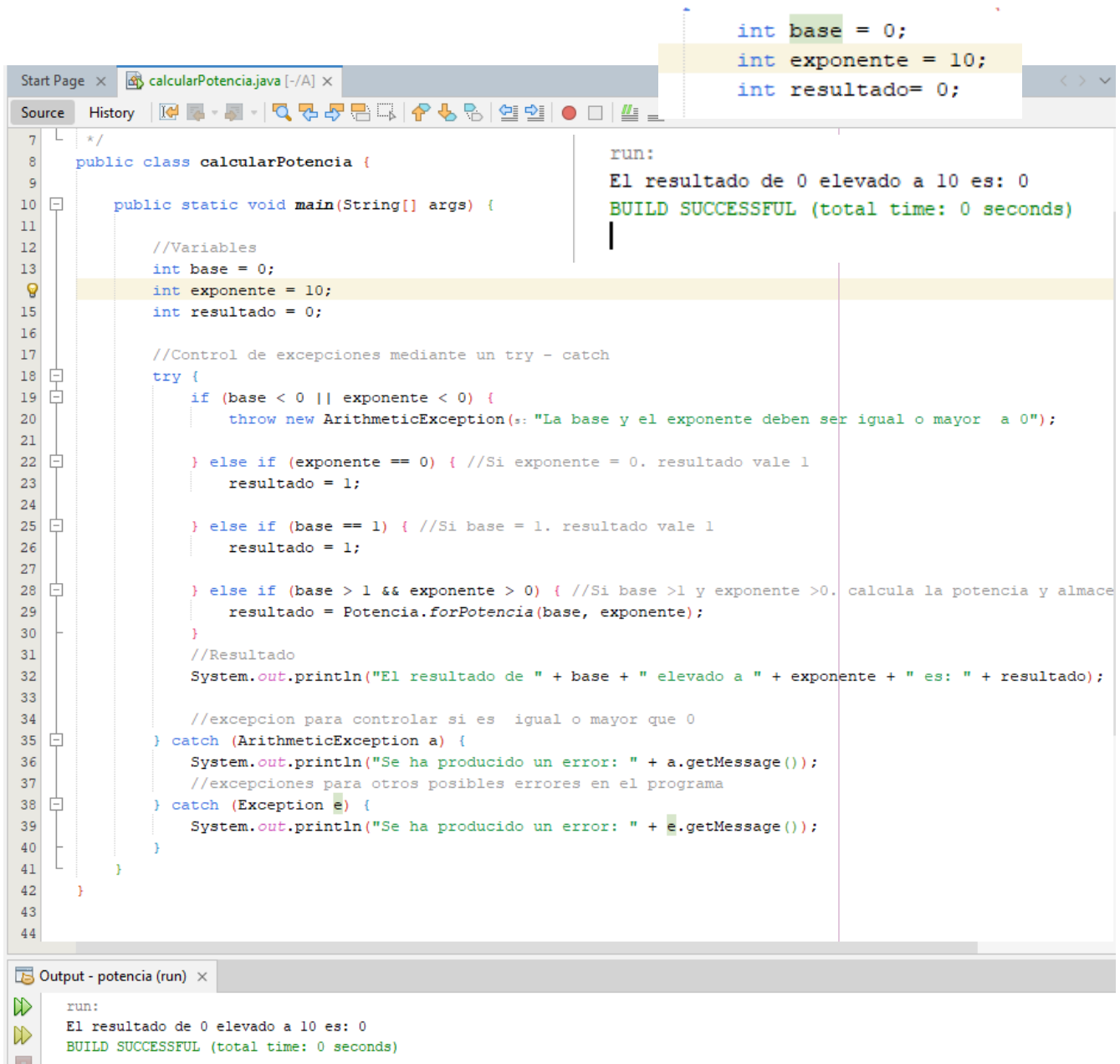


8. RA03_f) Se han probado y depurado los programas.

Hacer pruebas para varias combinaciones de valores de entrada.

Ejecutar la aplicación con estos valores y realizar las capturas de los resultados obtenidos

1. Base: 0 Exponente:10



The screenshot shows an IDE with a Java file named `calcularPotencia.java`. The code defines a `calcularPotencia` class with a `main` method. It initializes `base` to 0 and `exponente` to 10. The `main` method uses a `try-catch` block to handle exceptions. It checks if `base` is less than 0 or `exponente` is less than 0, throwing an `ArithmeticException` if so. If `exponente` is 0, it sets `resultado` to 1. If `base` is 1, it sets `resultado` to 1. Otherwise, it calculates the power using `Potencia.forPotencia`. The output shows the result of 0 raised to the power of 10 is 0.

```
7  /*
8  public class calcularPotencia {
9
10     public static void main(String[] args) {
11
12         //Variables
13         int base = 0;
14         int exponente = 10;
15         int resultado = 0;
16
17         //Control de excepciones mediante un try - catch
18         try {
19             if (base < 0 || exponente < 0) {
20                 throw new ArithmeticException("La base y el exponente deben ser igual o mayor a 0");
21             } else if (exponente == 0) { //Si exponente = 0. resultado vale 1
22                 resultado = 1;
23             } else if (base == 1) { //Si base = 1. resultado vale 1
24                 resultado = 1;
25             } else if (base > 1 && exponente > 0) { //Si base >1 y exponente >0. calcula la potencia y almace
26                 resultado = Potencia.forPotencia(base, exponente);
27             }
28             //Resultado
29             System.out.println("El resultado de " + base + " elevado a " + exponente + " es: " + resultado);
30
31             //excepcion para controlar si es igual o mayor que 0
32         } catch (ArithmeticException a) {
33             System.out.println("Se ha producido un error: " + a.getMessage());
34             //excepciones para otros posibles errores en el programa
35         } catch (Exception e) {
36             System.out.println("Se ha producido un error: " + e.getMessage());
37         }
38     }
39 }
40
41
42
43
44
```

run:
El resultado de 0 elevado a 10 es: 0
BUILD SUCCESSFUL (total time: 0 seconds)

Output - potencia (run) x

run:
El resultado de 0 elevado a 10 es: 0
BUILD SUCCESSFUL (total time: 0 seconds)

2. Base: 10 Exponente: 0

The screenshot shows an IDE with a Java file named `calcularPotencia.java`. The code defines a class `calcularPotencia` with a `main` method. It initializes `base` to 10 and `exponente` to 0. A try-catch block handles exceptions, with specific logic for `exponente == 0` (result is 1) and `base == 1` (result is 1). For other cases, it calls `Potencia.forPotencia`. The output window shows the successful execution of the program.

```
int base = 10;
int exponente = 0;
int resultado = 0;
```

```
7  L  */
8  public class calcularPotencia {
9
10     public static void main(String[] args)
11
12         //Variables
13         int base = 10;
14         int exponente = 0;
15         int resultado = 0;
16
17         //Control de excepciones mediante un try - catch
18         try {
19             if (base < 0 || exponente < 0) {
20                 throw new ArithmeticException(s: "La base y el exponente deben ser igual o mayor a 0");
21             }
22             else if (exponente == 0) { //Si exponente = 0. resultado vale 1
23                 resultado = 1;
24             }
25             else if (base == 1) { //Si base = 1. resultado vale 1
26                 resultado = 1;
27             }
28             else if (base > 1 && exponente > 0) { //Si base >1 y exponente >0. calcula la potencia y almace
29                 resultado = Potencia.forPotencia(base, exponente);
30             }
31             //Resultado
32             System.out.println("El resultado de " + base + " elevado a " + exponente + " es: " + resultado);
33
34             //excepcion para controlar si es igual o mayor que 0
35         } catch (ArithmeticException a) {
36             System.out.println("Se ha producido un error: " + a.getMessage());
37             //excepciones para otros posibles errores en el programa
38         } catch (Exception e) {
39             System.out.println("Se ha producido un error: " + e.getMessage());
40         }
41     }
42 }
43
44
```

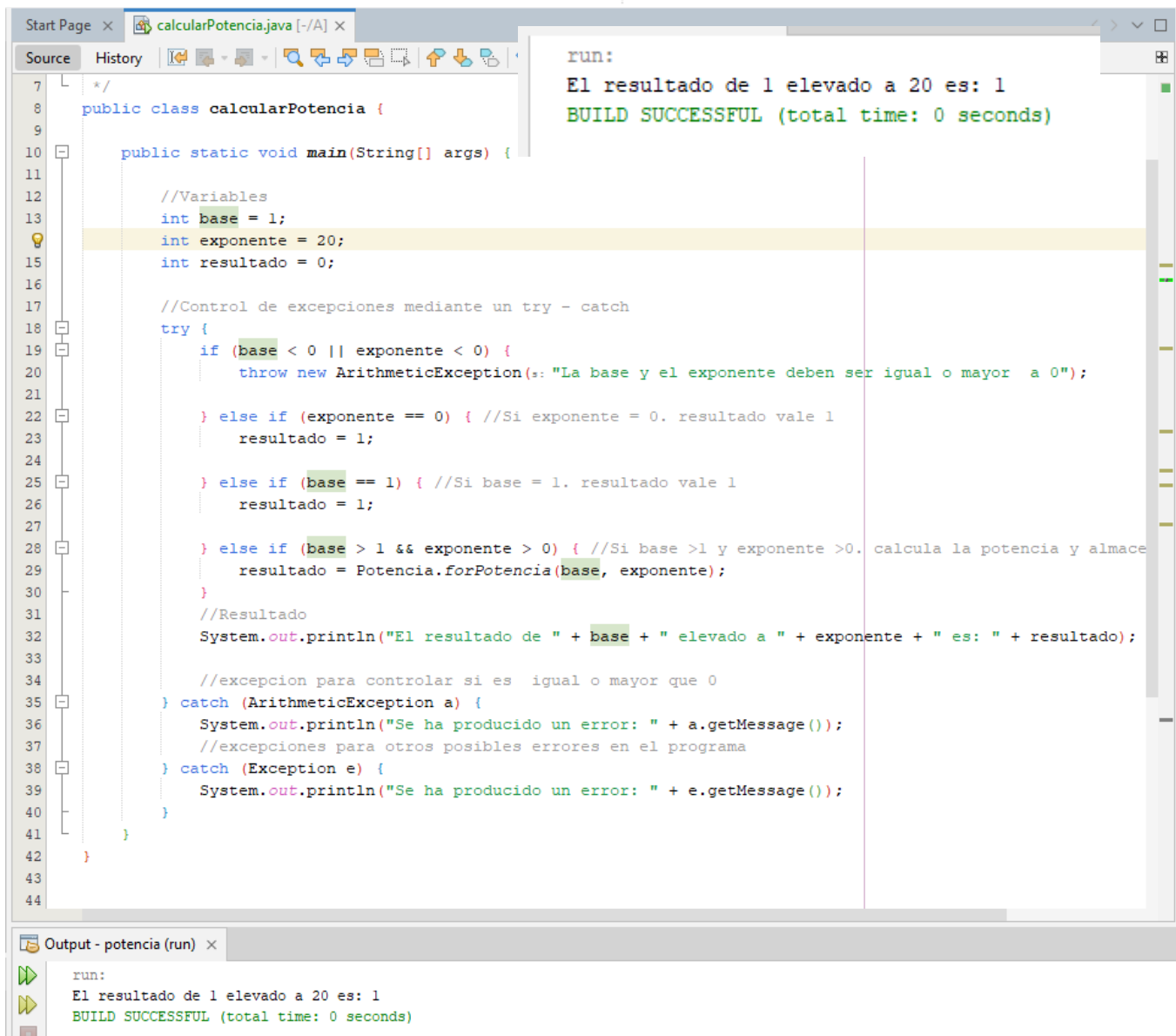
run:
El resultado de 10 elevado a 0 es: 1
BUILD SUCCESSFUL (total time: 0 seconds)

Output - potencia (run) x

run:
El resultado de 10 elevado a 0 es: 1
BUILD SUCCESSFUL (total time: 0 seconds)

3. Base: 1 Exponente: 20

```
int base = 1;  
int exponente = 20;
```



The screenshot shows an IDE with a Java file named `calcularPotencia.java`. The code defines a `calcularPotencia` class with a `main` method. It includes variable declarations for `base`, `exponente`, and `resultado`. A `try-catch` block handles exceptions, with specific logic for `base < 0`, `exponente == 0`, `base == 1`, and `base > 1` and `exponente > 0`. The program prints the result of the calculation. The output window shows the execution result: "El resultado de 1 elevado a 20 es: 1" and "BUILD SUCCESSFUL (total time: 0 seconds)".

```
7  */  
8  public class calcularPotencia {  
9  
10     public static void main(String[] args) {  
11  
12         //Variables  
13         int base = 1;  
14         int exponente = 20;  
15         int resultado = 0;  
16  
17         //Control de excepciones mediante un try - catch  
18         try {  
19             if (base < 0 || exponente < 0) {  
20                 throw new ArithmeticException("La base y el exponente deben ser igual o mayor a 0");  
21             }  
22             else if (exponente == 0) { //Si exponente = 0. resultado vale 1  
23                 resultado = 1;  
24             }  
25             else if (base == 1) { //Si base = 1. resultado vale 1  
26                 resultado = 1;  
27             }  
28             else if (base > 1 && exponente > 0) { //Si base >1 y exponente >0. calcula la potencia y almace  
29                 resultado = Potencia.forPotencia(base, exponente);  
30             }  
31             //Resultado  
32             System.out.println("El resultado de " + base + " elevado a " + exponente + " es: " + resultado);  
33  
34             //excepcion para controlar si es igual o mayor que 0  
35         } catch (ArithmeticException a) {  
36             System.out.println("Se ha producido un error: " + a.getMessage());  
37             //excepciones para otros posibles errores en el programa  
38         } catch (Exception e) {  
39             System.out.println("Se ha producido un error: " + e.getMessage());  
40         }  
41     }  
42 }  
43  
44
```

run:
El resultado de 1 elevado a 20 es: 1
BUILD SUCCESSFUL (total time: 0 seconds)

Output - potencia (run) X

run:
El resultado de 1 elevado a 20 es: 1
BUILD SUCCESSFUL (total time: 0 seconds)

4. Base: 5 Exponente: 3

```
int base = 5;  
int exponente = 3;  
int resultado = 0;
```

The screenshot shows an IDE with a Java file named `calcularPotencia.java`. The code defines a class `calcularPotencia` with a `main` method. It initializes `base` to 5 and `exponente` to 3. The `main` method uses a `try-catch` block to handle exceptions. Inside the `try` block, it checks if `base` or `exponente` are less than 0, if `exponente` is 0, or if `base` is 1. If none of these conditions are met, it calls `Potencia.forPotencia(base, exponente)` to calculate the result. The result is then printed to the console. The `catch` block handles `ArithmeticException` and other `Exception` types, printing error messages. The output window shows the successful execution of the program, displaying the result 125.

```
7  /*  
8  public class calcularPotencia {  
9  
10     public static void main(String[] args) {  
11  
12         //Variables  
13         int base = 5;  
14         int exponente = 3;  
15         int resultado = 0;  
16  
17         //Control de excepciones mediante un try - catch  
18         try {  
19             if (base < 0 || exponente < 0) {  
20                 throw new ArithmeticException("La base y el exponente deben ser igual o mayor a 0");  
21             } else if (exponente == 0) { //Si exponente = 0. resultado vale 1  
22                 resultado = 1;  
23             } else if (base == 1) { //Si base = 1. resultado vale 1  
24                 resultado = 1;  
25             } else if (base > 1 && exponente > 0) { //Si base >1 y exponente >0. calcula la potencia y almace  
26                 resultado = Potencia.forPotencia(base, exponente);  
27             }  
28             //Resultado  
29             System.out.println("El resultado de " + base + " elevado a " + exponente + " es: " + resultado);  
30  
31             //excepcion para controlar si es igual o mayor que 0  
32         } catch (ArithmeticException a) {  
33             System.out.println("Se ha producido un error: " + a.getMessage());  
34             //excepciones para otros posibles errores en el programa  
35         } catch (Exception e) {  
36             System.out.println("Se ha producido un error: " + e.getMessage());  
37         }  
38     }  
39 }  
40  
41  
42  
43  
44
```

run:
El resultado de 5 elevado a 3 es: 125
BUILD SUCCESSFUL (total time: 0 seconds)

Output - potencia (run) ×

run:
El resultado de 5 elevado a 3 es: 125
BUILD SUCCESSFUL (total time: 0 seconds)