

ciclo: [DAM]

MÓDULO DE [PROGRAMACIÓN]

[Tarea Nº 03]

Alumno:

[Juan Carlos Filter Martín]

[15456141A]

**Contenido**

[1. Documentos que se adjuntan a este informe. 3](#__RefHeading___Toc77_1091771104)

[2. RA03\_a) Se ha escrito y probado código que haga uso de estructuras de selección. 3](#__RefHeading___Toc79_1091771104)

[Crear un proyecto en Java en NetBeans que se denomine "potencia”. 3](#__RefHeading___Toc81_1091771104)

[3. RA03\_b) Se han utilizado estructuras de repetición. 4](#__RefHeading___Toc83_1091771104)

[Dentro del proyecto crear una clase "calcularPotencia.java" con un método main. 4](#__RefHeading___Toc85_1091771104)

[Declarar el siguiente conjunto de variables . 6](#__RefHeading___Toc87_1091771104)

[1. Declarar e inicializar una variable base. Almacenará el valor del número al cual se desea calcular la potencia. (valor entero). 6](#__RefHeading___Toc89_1091771104)

[2. Declarar e inicializar una variable exponente. Almacenará el valor numérico al que se desea elevar la base. (valor entero). 6](#__RefHeading___Toc91_1091771104)

[3. Declarar una variable resultado. Almacenará el valor numérico que resulta de elevar la base al exponente. (valor entero). 7](#__RefHeading___Toc93_1091771104)

[4. RA03\_e) Se han creado programas ejecutables utilizando diferentes estructuras de control. 7](#__RefHeading___Toc83_1091771104_Copia_1)

[Dentro del cálculo de la operación matemática potencia se distinguirán los siguientes casos: 7](#__RefHeading___Toc102_1091771104)

[1. Variable exponente sea inicializada a 0, en cuyo caso el resultado será 1. 7](#__RefHeading___Toc104_1091771104)

[2. Variable base sea inicializada a 1, en cuyo caso el resultado será 1. 7](#__RefHeading___Toc106_1091771104)

[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](#__RefHeading___Toc108_1091771104)

[4. Mostrar el resultado por pantalla. 10](#__RefHeading___Toc110_1091771104)

[5. RA03\_d) Se ha escrito código utilizando el control de excepciones. 11](#__RefHeading___Toc83_1091771104_Copia_1_)

[Todas las operaciones del programa deben de ir dentro de un control de excepciones que no permita que ocurran errores descontrolados. 11](#__RefHeading___Toc142_1091771104)

[6. RA03\_h) Se han creado excepciones. 12](#__RefHeading___Toc83_1091771104_Copia_11)

[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](#__RefHeading___Toc144_1091771104)

[7. RA03\_g) Se ha comentado y documentado el código. 13](#__RefHeading___Toc83_1091771104_Copia_12)

[Capturas que muestren el resultado de la ejecución del programa en la documentación. 13](#__RefHeading___Toc146_1091771104)

[8. RA03\_f) Se han probado y depurado los programas. 14](#__RefHeading___Toc83_1091771104_Copia_13)

[Hacer pruebas para varias combinaciones de valores de entrada. 14](#__RefHeading___Toc148_1091771104)

[1. Base: 0 Exponente:10 14](#__RefHeading___Toc150_1091771104)

[2. Base: 10 Exponente: 0 15](#__RefHeading___Toc152_1091771104)

[3. Base: 1 Exponente: 20 16](#__RefHeading___Toc154_1091771104)

[4. Base: 5 Exponente: 3 17](#__RefHeading___Toc156_1091771104)

# 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

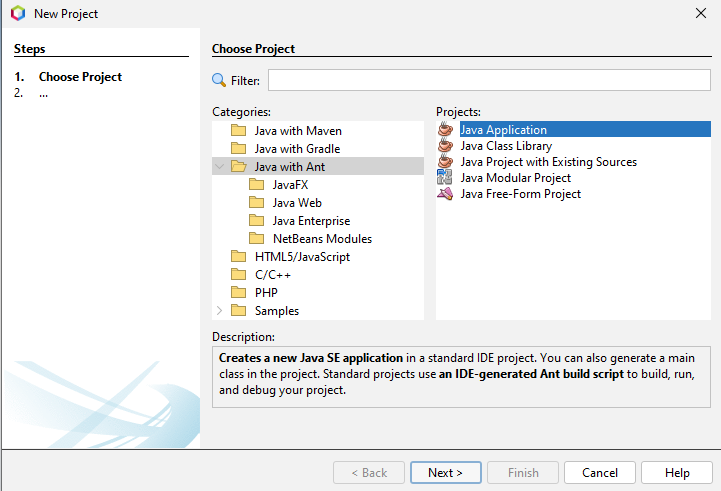
# RA03\_a) Se ha escrito y probado código que haga uso de estructuras de selección.

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

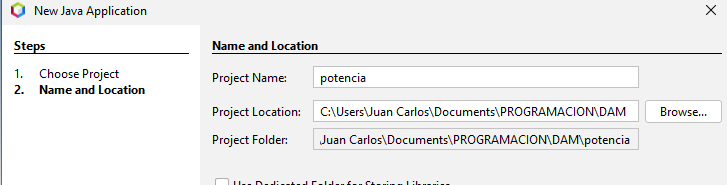
Para ello vamos a:

**File> New Project > Java with Ant > Java Application**

## 



Le asignamos un nombre al proyecto : “potencia”



Ya tendríamos creado el proyecto potencia.

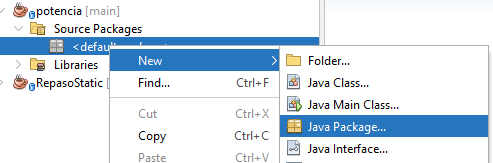
# RA03\_b) Se han utilizado estructuras de repetición.

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

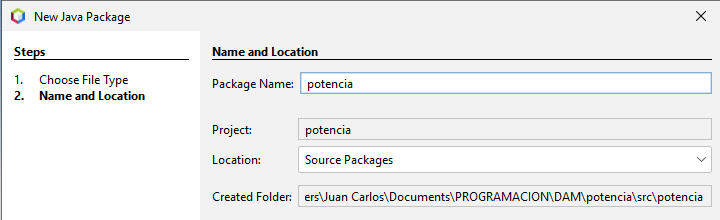
Primero hay que crear un Package que va a contener la clase del proyecto y para ello vamos a:

Click derecho en el Package vacío dentro del proyecto

**New > Java Package**



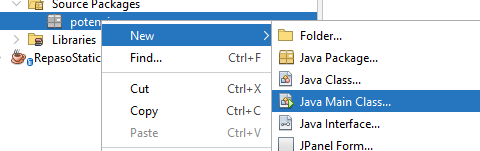
Ponemos el nombre del paquete que puede ser el del proyecto “potencia”



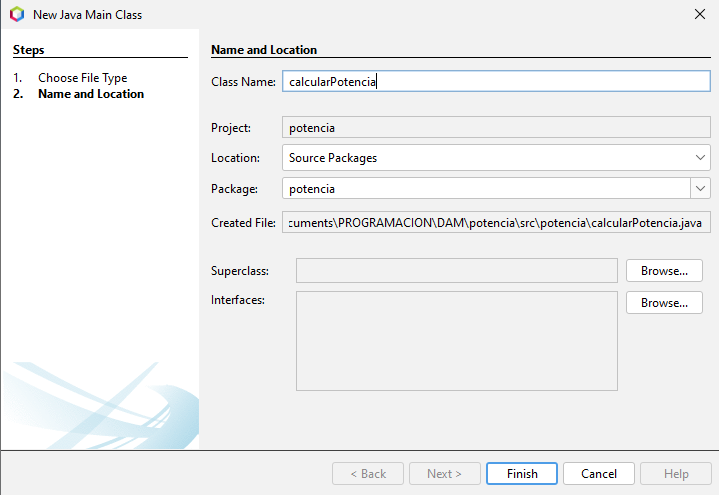
Una vez creado el Package toca crear la clase main:

Click derecho en el Package

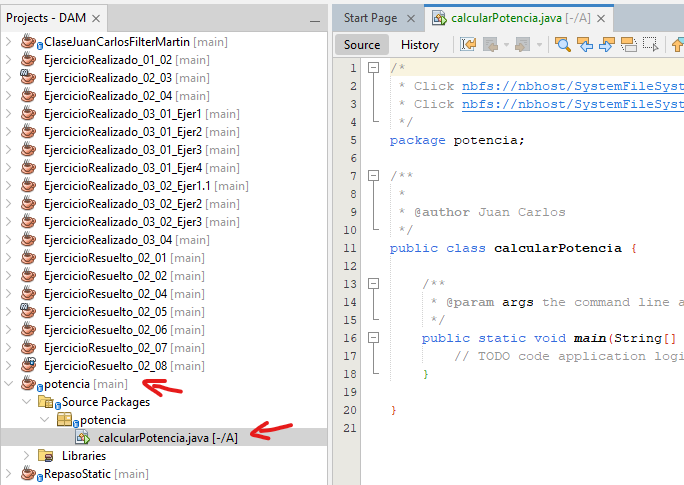
**New > Java Main Class**



Ponemos el nombre a la clase main calcularPotencia



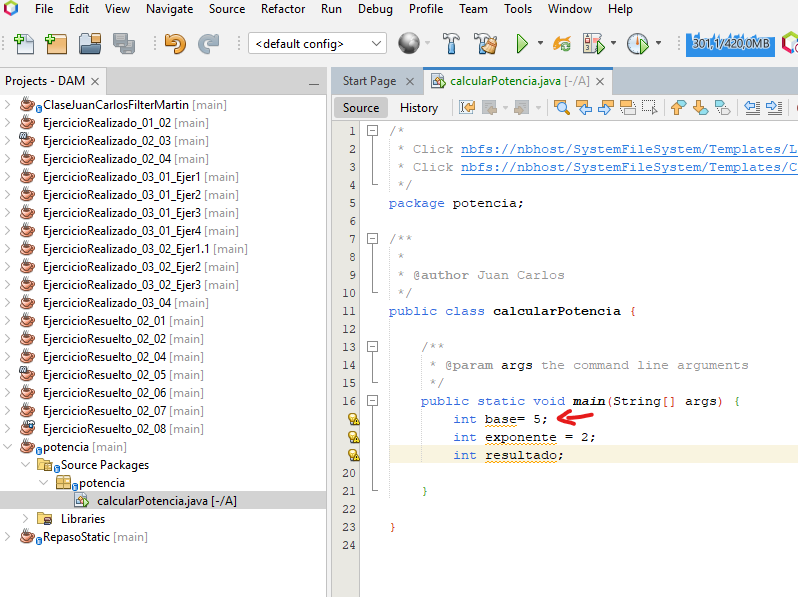
Y ya tendríamos nuestra clase main calcularPotencia,java creada



## **Declarar el siguiente conjunto de variables .**

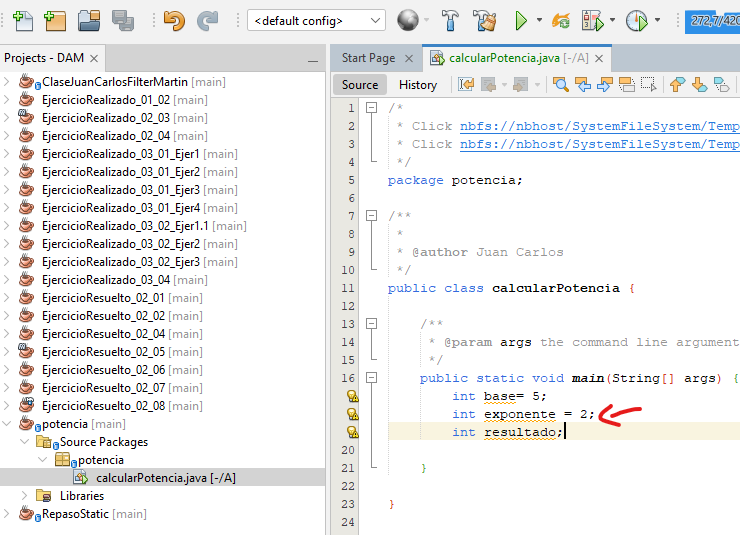
### Declarar e inicializar una variable base. Almacenará el valor del número al cual se desea calcular la potencia. (valor entero).





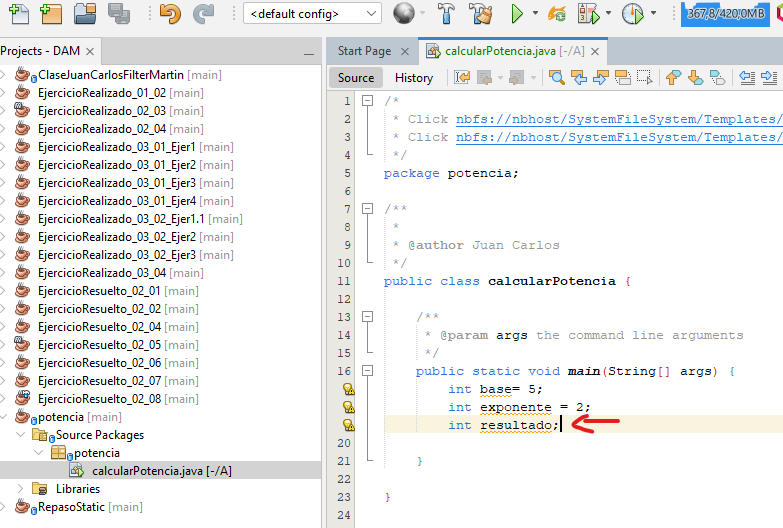
### **Declarar e inicializar una variable exponente. Almacenará el valor numérico al que se desea elevar la base. (valor entero).**





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





# 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:**

### 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



### 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.



### 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++:

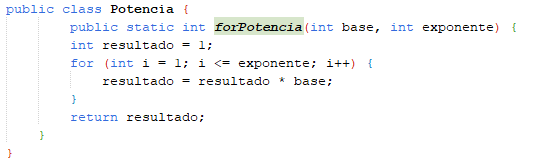
**Mientras i sea <= exponente ( 2 <= 2 )…. entonces:**

* Resultado \* base ( 5 \* 5 ) = 25

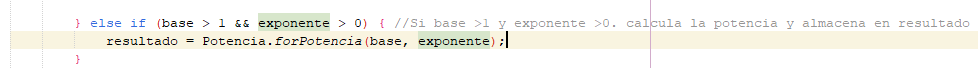
*Ahora resultado vale 25 y volvemos al for incrementando i++:*

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

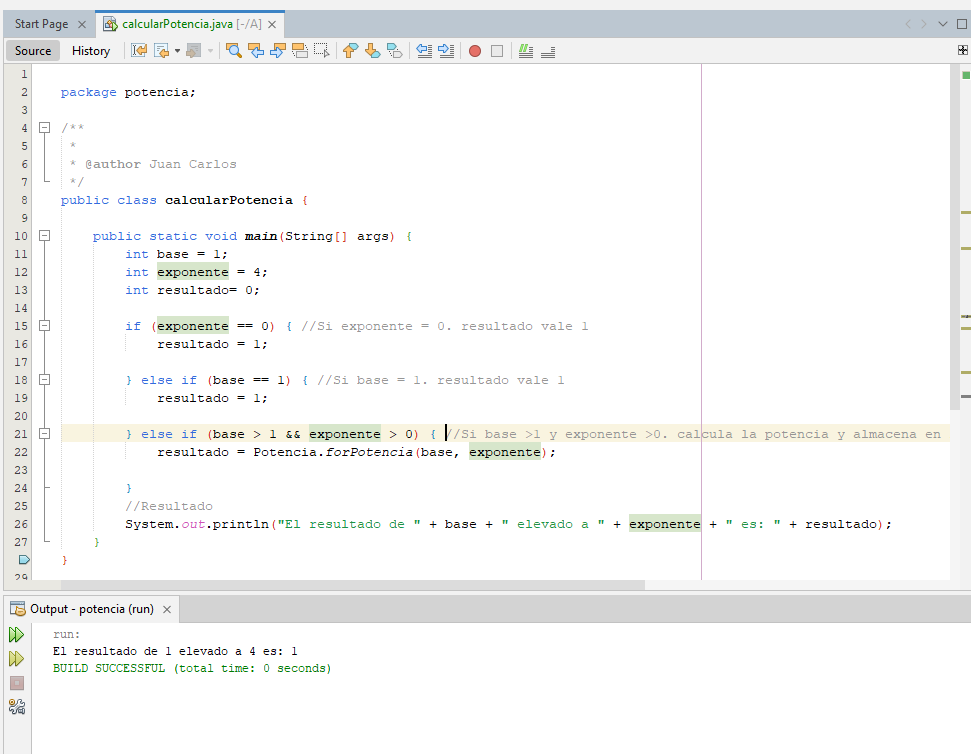
No puede entrar en el bucle for porque i es > exponente entonces sale y imprime por pantalla el resultado ( 25 ) con el System.out.print



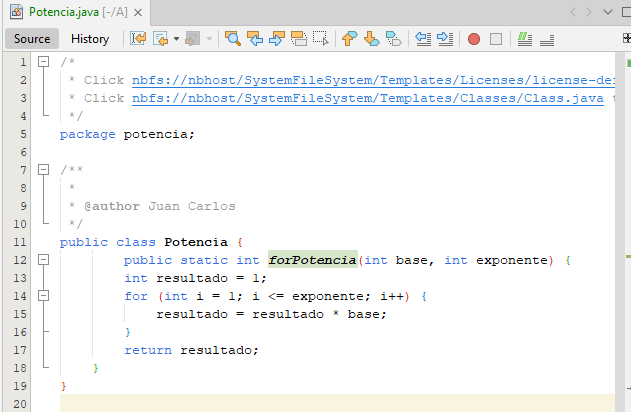
→ Ahora vamos a la clase main y llamamos a ese método dentro de un if indicando que si la base es > 1 y el exponente es >0 entonces



Clase main



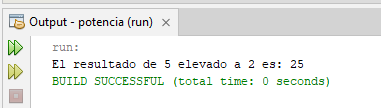
Clase con método para calcular la potencia



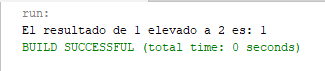
### Mostrar el resultado por pantalla.

Varias capturas de pantalla con diferentes resultados

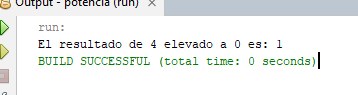
**Base 5 , exponente 2.**



**Base 1 , exponente 2.**



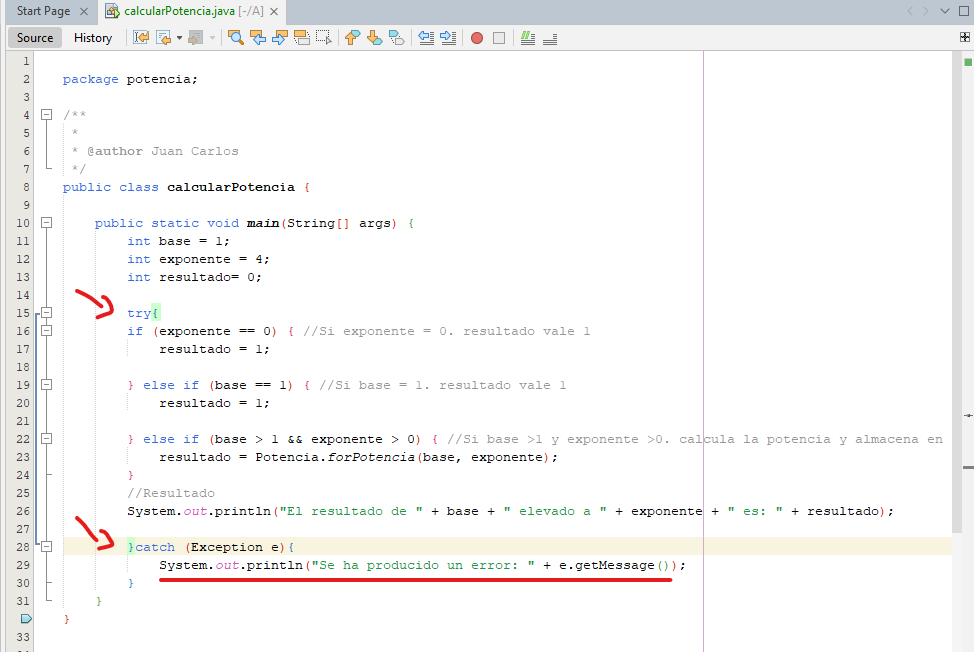
**Base 4 , exponente 0.**



# 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)



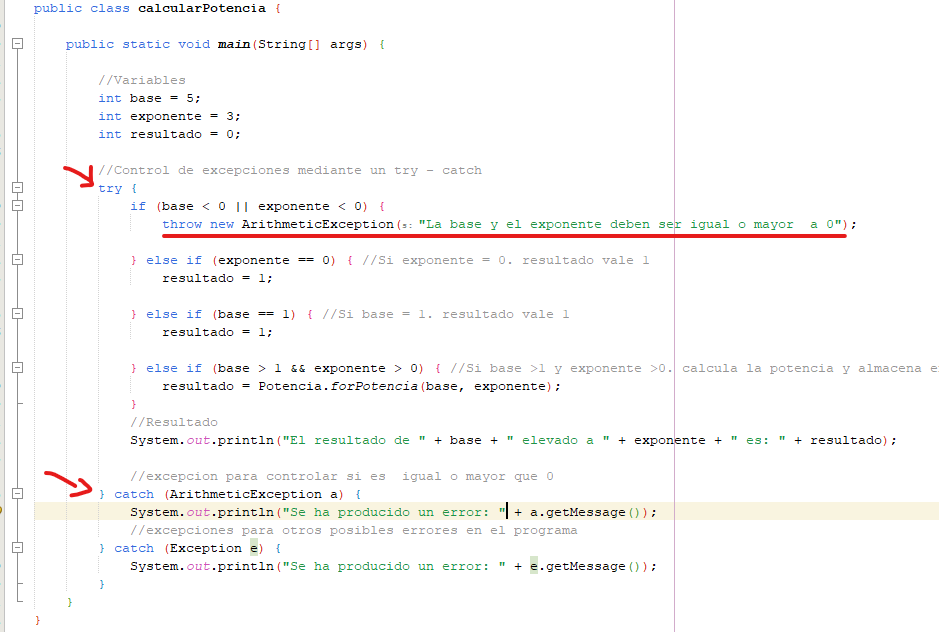
# RA03\_h) Se han creado excepciones.

## 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** indicandole **que** agregue la excepción

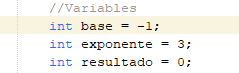
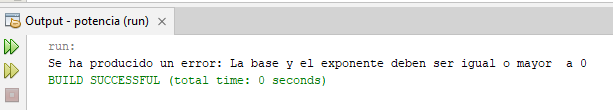
*(Se ha colocado el if en primer lugar ya que si no fuera de esta forma podría entrar antes en otra condición si por ejemplo se le dice que base = -1 y exponente = 0)*



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

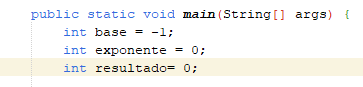
→ Segundo catch mas general para controlar el resto del programa

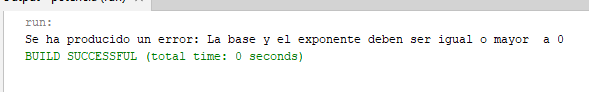


# RA03\_g) Se ha comentado y documentado el código.

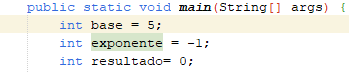
## Capturas que muestren el resultado de la ejecución del programa en la documentación.

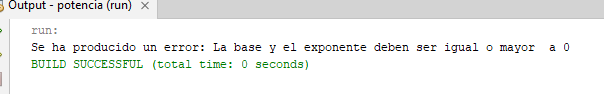
Se van a dejar resultados sobre el control de excepciones

**Error base = -1**

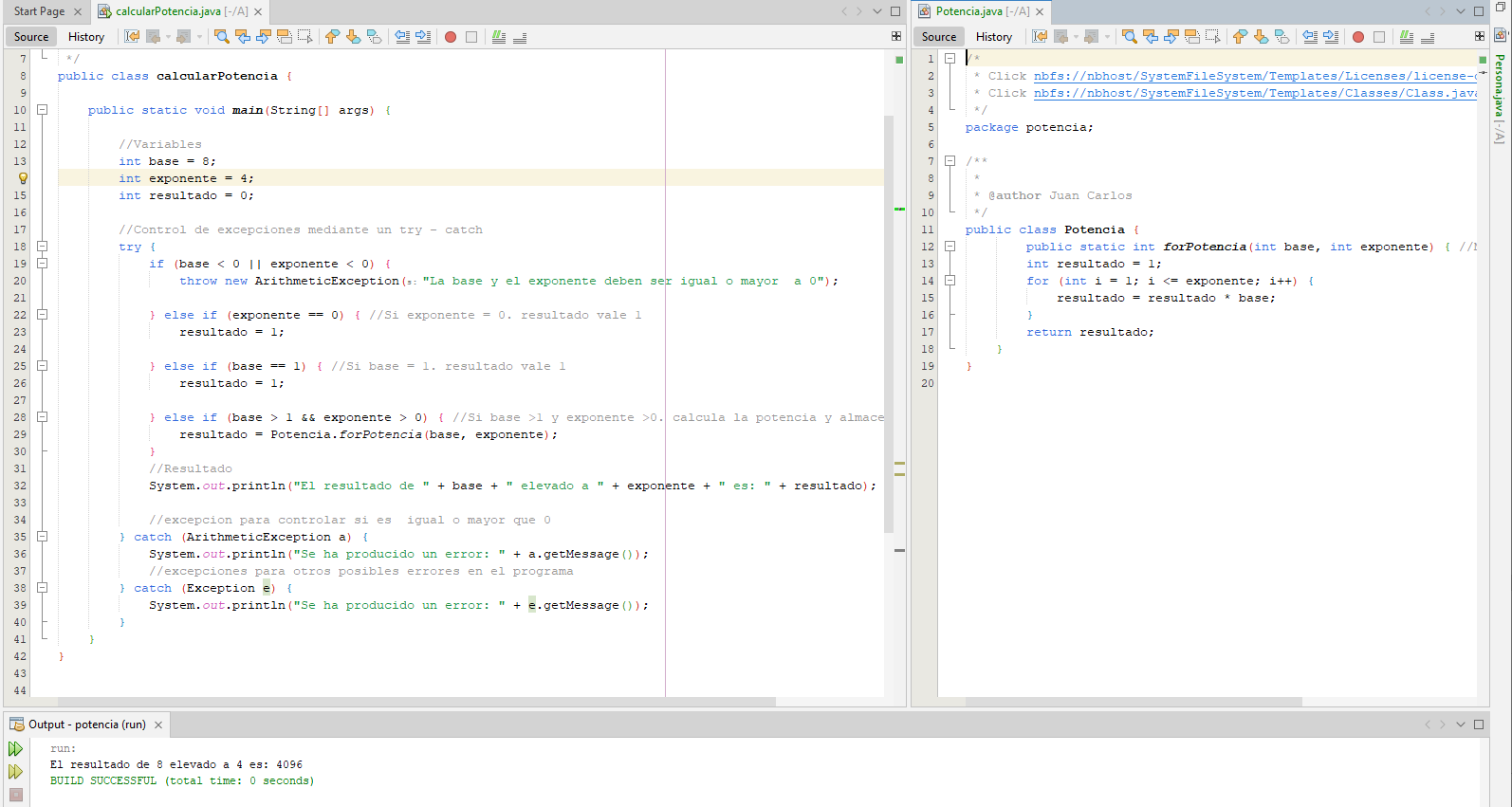
****

**Error exponente = -1**





**Resultado de todo el programa:**

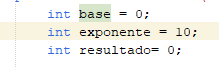
****

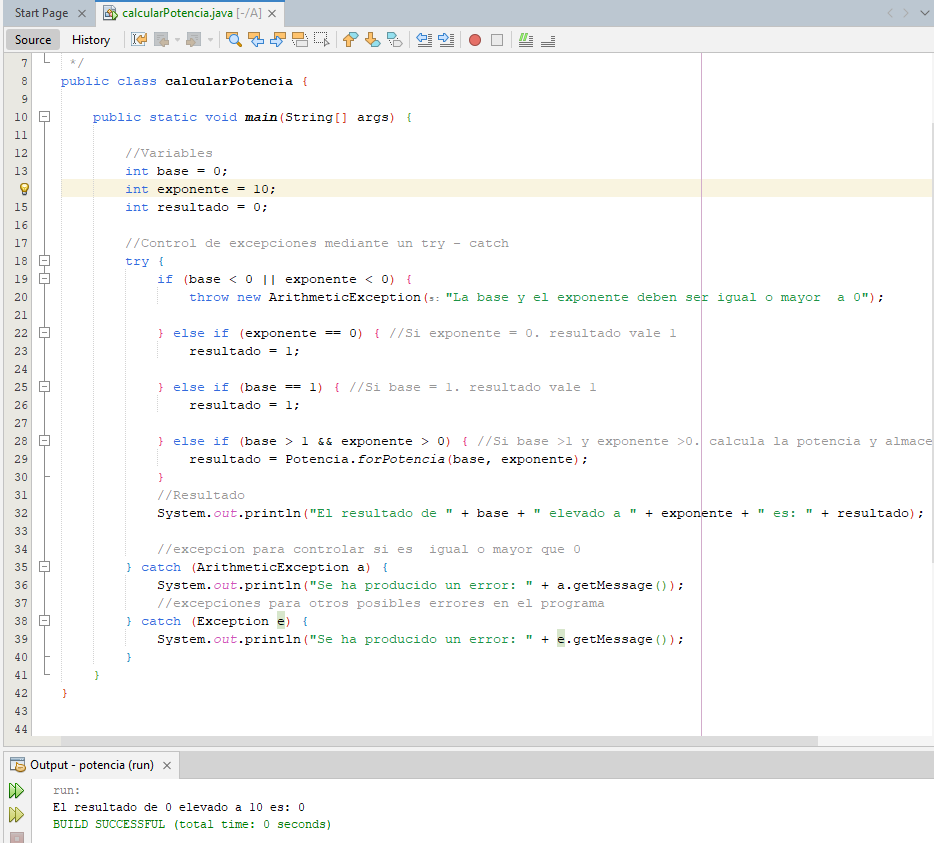
# RA03\_f) Se han probado y depurado los programas.

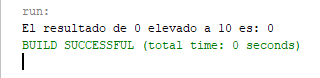
## **Hacer pruebas para varias combinaciones de valores de entrada.**

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

### Base: 0 Exponente:10

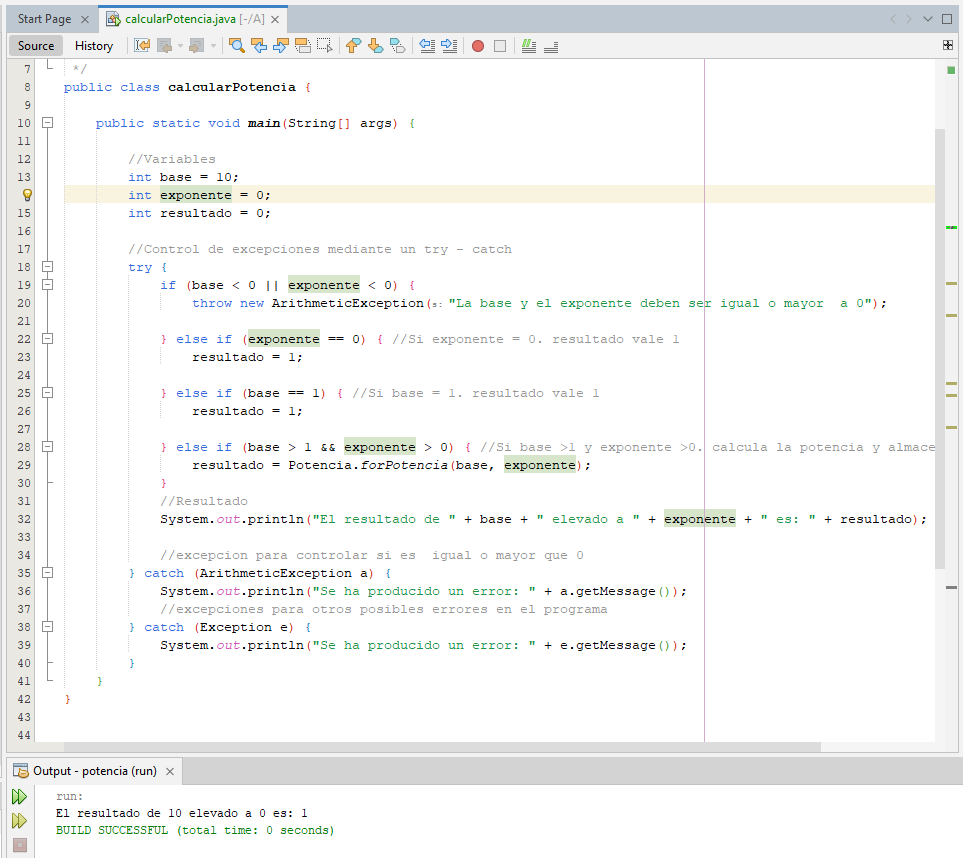


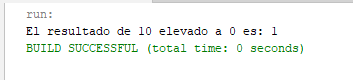




### Base: 10 Exponente: 0

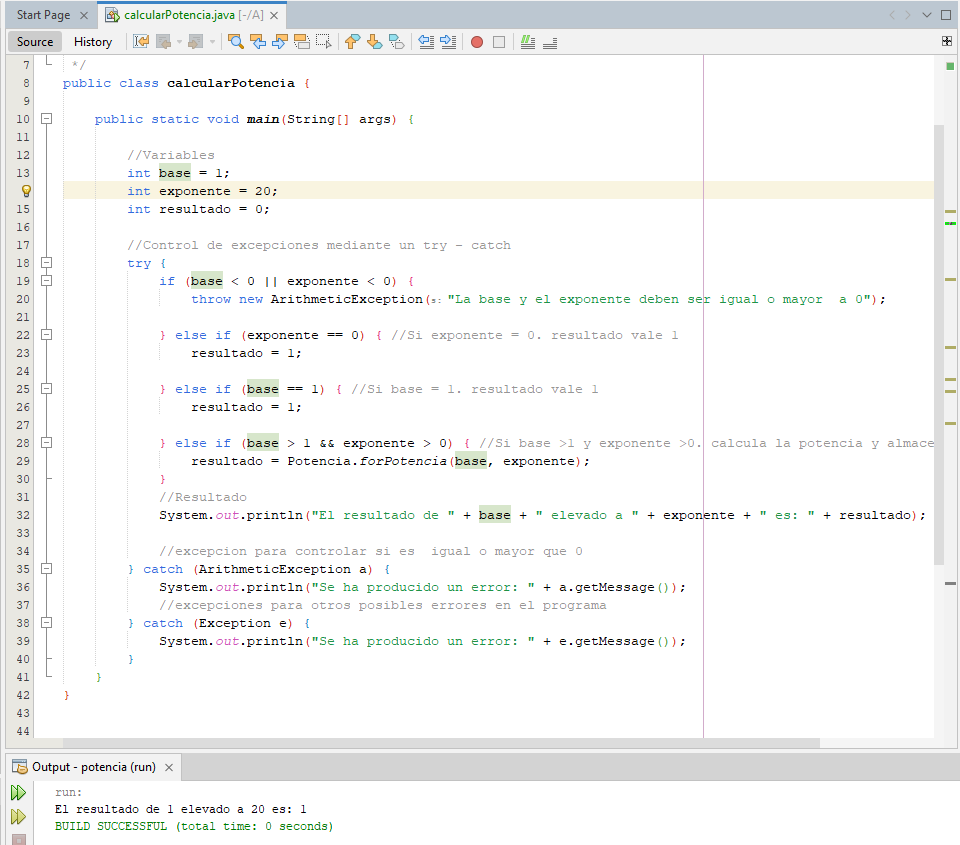


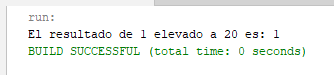




### Base: 1 Exponente: 20

### 





### Base: 5 Exponente: 3



