

# Tarea para ED03

1. Crear un proyecto en Apache NetBeans cuyo nombre sea "Tarea3\_TuNombre" e incorpora las dos clases proporcionadas en el package es.maestre.

The screenshot shows a dual-monitor setup. On the left monitor, a web browser displays course details for 'LE-Entornos de desarrollo DAW E'. It includes sections for General, Activities, Search forums, Private files, My courses, and Topic 1. On the right monitor, an Apache NetBeans IDE window is open, showing the source code for 'Tarea3\_Nelson.java' and 'Calculadora.java'. The code implements a simple calculator with methods for addition, subtraction, multiplication, division, and exponentiation. The Java code uses static methods to handle command-line arguments and print results to the console.

```
public class Tarea3_Nelson {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("INICIO DE EXECUCION");
        System.out.println("Nº dígitos del valor 0 = 1");
        System.out.println("Nº dígitos del valor 2147483647 es 10");
        2 elevado a 8 = 256
    }
}
```

```
public class Calculadora {
    /**
     * Devuelve el nº de cifras de un número entero.
     * @param n número positivo o negativo; también puede ser 0.
     * @return número de cifras que tiene el número pasado
     */
    public static int cuentaCifras(int n) {
        int aux;
        if (n == 0) {
            return 1;
        }
        if (n < 0) {
            // Si es negativo, obtengo el valor absoluto.
            aux = -n;
        } else {
            aux = n;
        }
        // Hago divisiones sucesivas entre 10
        while (aux > 0) {
            aux /= 10;
            cifras++;
        }
        return cifras;
    }
}
```

2. Compilar y ejecutar la aplicación, viendo la salida que da por consola.

This screenshot is nearly identical to the previous one, showing the same course details on the left and the Java code in the NetBeans IDE on the right. The code remains the same, demonstrating the implementation of basic arithmetic operations and exponentiation.

```
public class Tarea3_Nelson {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("INICIO DE EXECUCION");
        System.out.println("Nº dígitos del valor 0 = 1");
        System.out.println("Nº dígitos del valor 2147483647 es 10");
        2 elevado a 8 = 256
    }
}
```

```
public class Calculadora {
    /**
     * Devuelve el nº de cifras de un número entero.
     * @param n número positivo o negativo; también puede ser 0.
     * @return número de cifras que tiene el número pasado
     */
    public static int cuentaCifras(int n) {
        int aux;
        if (n == 0) {
            return 1;
        }
        if (n < 0) {
            // Si es negativo, obtengo el valor absoluto.
            aux = -n;
        } else {
            aux = n;
        }
        // Hago divisiones sucesivas entre 10
        while (aux > 0) {
            aux /= 10;
            cifras++;
        }
        return cifras;
    }
}
```

3. Establecer dos breakpoints:

- Uno en la linea 20(18 en mi caso) de Tarea3Main.java (la que pone INICIO DE EJECUCION).
- Otro en la linea 34(32 en mi caso) de Tarea3Main.java (la que pone -POTENCIA--).

This screenshot shows the Java code in the NetBeans IDE with two breakpoints set: one at line 20 (marked with a red line) and another at line 34 (also marked with a red line). The code is identical to the previous screenshots, illustrating the setup for debugging.

```
public class Tarea3_Nelson {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("INICIO DE EXECUCION");
        System.out.println("Nº dígitos del valor 0 = 1");
        System.out.println("Nº dígitos del valor 2147483647 es 10");
        2 elevado a 8 = 256
    }
}
```

```
public class Calculadora {
    /**
     * Devuelve el nº de cifras de un número entero.
     * @param n número positivo o negativo; también puede ser 0.
     * @return número de cifras que tiene el número pasado
     */
    public static int cuentaCifras(int n) {
        int aux;
        if (n == 0) {
            return 1;
        }
        if (n < 0) {
            // Si es negativo, obtengo el valor absoluto.
            aux = -n;
        } else {
            aux = n;
        }
        // Hago divisiones sucesivas entre 10
        while (aux > 0) {
            aux /= 10;
            cifras++;
        }
        return cifras;
    }
}
```

#### 4. Iniciar una sesión con el depurador para ejecutar paso a paso la aplicación. La ejecución se detendrá en el primer breakpoint (línea 18).

LE-Entornos de desarrollo DAW E

Situación actual:  
Antes de iniciar la primera unidad os recomendamos que reviséis la [Guía del alumno](#), ubicada en el Curso común a todo el alumnado "CFGDS Desarrollo de Aplicaciones Web" y el [Programa del módulo de Entornos de Desarrollo](#).

Si te surge cualquier duda puedes contactar con:  
Tutor del módulo: Antonio Cano Supervia  
Horario de atención: Martes de 17:20 a 18:15 y de 18:30 a 19:25 / Viernes de 19:25 a 20:20

General

Activities

Search forums

Private files

No files available

Manage private files...

My courses

Topic 1

Unidad de Trabajo 1.- Desarrollo de software.

IDE: Apache NetBeans IDE 12.5

Tarea3\_Nelson - Apache NetBeans IDE 12.5

Source

```
/*
 * Application that performs various simple calculations, invoking methods
 * from the Calculator class.
 */
public class Tarea3_Nelson {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("***** INICIO *****");
        System.out.println("***** FIN *****");
    }

    /**
     * Llama al método cuentaCifras
     */
    int num = 0;
    int numDigitos = Calculadora.cuentaCifras(num);
    System.out.println("El número " + num + " tiene " + numDigitos + " dígitos");

    num = Integer.MAX_VALUE;
    numDigitos = Calculadora.cuentaCifras(num);
    System.out.println("El número " + num + " tiene " + numDigitos + " dígitos");

    System.out.println("***** FIN *****");
}

int valBase = 2, valExpo = 8;
int result = Calculadora.potencia(valBase, valExpo);
System.out.println("El resultado es " + result);
System.out.println("valBase, valExpo, result: " + valBase + ", " + valExpo + ", " + result);
}
```

Output

Name	Type	Value
args	String	2

VARIABLES

Name	Type	Value
args	String	2

a) Avanza paso a paso hasta la línea 23 (la llamada al `cuentaCifras`) e inspecciona las variables (num tiene que tener 0). Como se puede ver en la captura, la variable “num” tiene valor 0.

LE-Entornos de desarrollo DAW E

Situación actual:  
Antes de iniciar la primera unidad os recomendamos que reviséis la [Guía del alumno](#), ubicada en el Curso común a todo el alumnado "CFGDS Desarrollo de Aplicaciones Web" y el [Programa del módulo de Entornos de Desarrollo](#).

Si te surge cualquier duda puedes contactar con:  
Tutor del módulo: Antonio Cano Supervia  
Horario de atención: Martes de 17:20 a 18:15 y de 18:30 a 19:25 / Viernes de 19:25 a 20:20

General

Activities

Search forums

Private files

No files available

Manage private files...

My courses

Topic 1

Unidad de Trabajo 1.- Desarrollo de software.

IDE: Apache NetBeans IDE 12.5

Tarea3\_Nelson - Apache NetBeans IDE 12.5

Source

```
/*
 * Application that performs various simple calculations, invoking methods
 * from the Calculator class.
 */
public class Tarea3_Nelson {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("***** INICIO *****");
        System.out.println("***** FIN *****");
    }

    /**
     * Llama al método cuentaCifras
     */
    int num = 0;
    int numDigitos = Calculadora.cuentaCifras(num);
    System.out.println("El número " + num + " tiene " + numDigitos + " dígitos");

    num = Integer.MAX_VALUE;
    numDigitos = Calculadora.cuentaCifras(num);
    System.out.println("El número " + num + " tiene " + numDigitos + " dígitos");

    System.out.println("***** FIN *****");
}

int valBase = 2, valExpo = 8;
int result = Calculadora.potencia(valBase, valExpo);
System.out.println("El resultado es " + result);
System.out.println("valBase, valExpo, result: " + valBase + ", " + valExpo + ", " + result);
}
```

Output

Name	Type	Value
args	String	2

VARIABLES

Name	Type	Value
args	String	2

b) Avanza paso a paso sin entrar en el método hasta la línea 30 (la 2ª llamada al `cuentaCifras`) e inspecciona el valor de la variable num (tiene que tener un valor muy grande).  
Como se puede ver en la captura, la variable “num” tiene valor 2147483647.

LE-Entornos de desarrollo DAW E

Situación actual:  
Antes de iniciar la primera unidad os recomendamos que reviséis la [Guía del alumno](#), ubicada en el Curso común a todo el alumnado "CFGDS Desarrollo de Aplicaciones Web" y el [Programa del módulo de Entornos de Desarrollo](#).

Si te surge cualquier duda puedes contactar con:  
Tutor del módulo: Antonio Cano Supervia  
Horario de atención: Martes de 17:20 a 18:15 y de 18:30 a 19:25 / Viernes de 19:25 a 20:20

General

Activities

Search forums

Private files

No files available

Manage private files...

My courses

Topic 1

Unidad de Trabajo 1.- Desarrollo de software.

IDE: Apache NetBeans IDE 12.5

Tarea3\_Nelson - Apache NetBeans IDE 12.5

Source

```
/*
 * Application that performs various simple calculations, invoking methods
 * from the Calculator class.
 */
public class Tarea3_Nelson {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("***** INICIO *****");
        System.out.println("***** FIN *****");
    }

    /**
     * Llama al método cuentaCifras
     */
    int num = 0;
    int numDigitos = Calculadora.cuentaCifras(num);
    System.out.println("El número " + num + " tiene " + numDigitos + " dígitos");

    num = Integer.MAX_VALUE;
    numDigitos = Calculadora.cuentaCifras(num);
    System.out.println("El número " + num + " tiene " + numDigitos + " dígitos");

    System.out.println("***** FIN *****");
}

int valBase = 2, valExpo = 8;
int result = Calculadora.potencia(valBase, valExpo);
System.out.println("El resultado es " + result);
System.out.println("valBase, valExpo, result: " + valBase + ", " + valExpo + ", " + result);
}
```

Output

Name	Type	Value
args	String	2

VARIABLES

Name	Type	Value
args	String	2
numDigitos	int	2147483647

- c) Ahora avanza paso a paso entrando en el método cuentaCifras.  
d) Avanza paso a paso dentro del método hasta que hayas ejecutado el bucle while al menos 4 veces. Inspecciona el valor de la variable cifras.

Como se puede ver en la captura, la variable "cifras" tiene valor 4.

```

    /**
     * Devuelve el nº de cifras de un número entero.
     * @param num número positivo o negativo; también puede ser 0.
     * @return nº de cifras que tiene el número pasado
     */
    public static int cuentaCifras(int num) {
        int cifras;
        int aux;
        if (num < 0) {
            aux = -num;
        } else {
            aux = num;
        }
        // Llegó divisiones sucesivas entre 10
        while (aux > 0) {
            aux /= 10;
            cifras++;
        }
        return cifras;
    }
}

```

- e) Reanuda la ejecución hasta llegar al 2º breakpoint. Inspecciona el valor de las variables valBase y valExpo antes de llamar al método potencia.

Como se puede ver en la captura, la variable "valBase" tiene valor 2 y la variable "valExpo" tiene valor 8.

```

    /**
     * @param args
     */
    public static void main(String[] args) {
        System.out.println("1 2 3 4 5 6 7 8 9");
        System.out.println("-----");
        // Llamo el método cuentaCifras
        int numCifras = Calculadora.cuentaCifras(num);
        System.out.println("El nº de cifras del valor " + num + " es " + numCifras);
        num = Integer.MAX_VALUE;
        numDigits = Calculadora.cuentaCifras(num);
        System.out.println("El nº de cifras del valor " + num + " es " + numDigits);
        System.out.println("-----");
        int valBase = 2, valExpo = 8;
        int result = Calculadora.potencia(valBase, valExpo);
        System.out.println("El resultado es " + result);
        System.out.println("-----");
    }
}

```

- f) Finaliza la sesión de depuración.

```

    /**
     * @param args
     */
    public static void main(String[] args) {
        System.out.println("1 2 3 4 5 6 7 8 9");
        System.out.println("-----");
        // Llamo el método cuentaCifras
        int numCifras = Calculadora.cuentaCifras(num);
        System.out.println("El nº de cifras del valor " + num + " es " + numCifras);
        num = Integer.MAX_VALUE;
        numDigits = Calculadora.cuentaCifras(num);
        System.out.println("El nº de cifras del valor " + num + " es " + numDigits);
        System.out.println("-----");
        int valBase = 2, valExpo = 8;
        int result = Calculadora.potencia(valBase, valExpo);
        System.out.println("El resultado es " + result);
        System.out.println("-----");
    }
}

```

5. Crear una Aplicación JUnit para la clase Calculador. Utiliza la estrategia de caja blanca para diseñar los casos de prueba. (para este ejercicio, además de los CONTENIDOS interactivos de la plataforma, conviene estudiar bien el anexo que se ha puesto en el apartado CONTENIDOS de la unidad)

The screenshot shows a course page titled 'LE-Entornos de desarrollo DAW E'. The main content area displays a welcome message and a 'Situation actual' box. The sidebar contains sections for 'General', 'Activities', 'Search forums', 'Private files', and 'My courses'.

**Code Editor (NetBeans IDE 12.5):**

```

package tarea3_nelson;
import org.junit.Test;
import static org.junit.Assert.*;
* @author Nelson Blanco Charro
public class CalculadorTest {
    /**
     * Test of cuentaCifras method, of class Calculador.
     */
    public void testCuentaCifras() {
        System.out.println("Primer caso de prueba para el metodo cuentaCifras");
        int num = 10;
        int resultado = Calculador.cuentaCifras(num);
        assertEquals(expResult, resultado);
    }
    /**
     * Segundo caso de prueba para el metodo cuentaCifras.
     */
    public void testCuentaCifras2() {
        System.out.println("Segundo caso de prueba para el metodo cuentaCifras");
        int num = 12345;
        int resultado = Calculador.cuentaCifras(num);
        assertEquals(expResult, resultado);
    }
    /**
     * Tercer caso de prueba para el metodo cuentaCifras.
     */
    public void testCuentaCifras3() {
        System.out.println("Tercer caso de prueba para el metodo cuentaCifras");
        int num = -250;
        int resultado = Calculador.cuentaCifras(num);
        assertEquals(expResult, resultado);
    }
    /**
     * Cuarto caso de prueba para el metodo cuentaCifras.
     */
    public void testCuentaCifras4() {
        System.out.println("Cuarto caso de prueba para el metodo cuentaCifras");
        int num = -1222333444;
        int resultado = Calculador.cuentaCifras(num);
        assertEquals(expResult, resultado);
    }
}

```

**Output (NetBeans IDE 12.5):**

```

Tarea3_Nelson (debug) > Debugger Console <
=====
D E B U G   C O N F I G U R A T I O N
=====
Nº dígitos del valor es 1
Nº dígitos del valor 2167483647 es 10
Nº dígitos del valor -250 es 3
2 dígitos a 8 es 256
BUILD SUCCESSFUL (total time: 2 minutes 24 seconds)

```

### Piensa en casos de prueba significativos.

a) Escribe cuatro casos de prueba para el método cuentaCifras.

- > El primer test es el número 10, por lo que el resultado esperado son 2 cifras.
  - > El segundo test es el número 12345, por lo que el resultado esperado son 5 cifras.
  - > El tercer test es el número -250, por lo que el resultado esperado son 3 cifras.
  - > El cuarto test es el número -1222333444, por lo que el resultado esperado son 10 cifras.
- b) Escribe tres casos de prueba para el método potencia.
- > El primer test la base es el número 3 y como exponente el 10, por lo que el resultado esperado es 59049.
  - > El primer test la base es el número -2 y como exponente el 6, por lo que el resultado esperado es 64.
  - > El primer test la base es el número 0 y como exponente el 100, por lo que el resultado esperado es 0.

The screenshot shows the same course page and sidebar as the previous one.

**Code Editor (NetBeans IDE 12.5):**

```

package tarea3_nelson;
import org.junit.Test;
import static org.junit.Assert.*;
* @author Nelson Blanco Charro
public class CalculadorTest {
    /**
     * Test of cuentaCifras method, of class Calculador.
     */
    public void testCuentaCifras() {
        System.out.println("Primer caso de prueba para el metodo cuentaCifras");
        int num = 10;
        int resultado = Calculador.cuentaCifras(num);
        assertEquals(expResult, resultado);
    }
    /**
     * Segundo caso de prueba para el metodo cuentaCifras.
     */
    public void testCuentaCifras2() {
        System.out.println("Segundo caso de prueba para el metodo cuentaCifras");
        int num = 12345;
        int resultado = Calculador.cuentaCifras(num);
        assertEquals(expResult, resultado);
    }
    /**
     * Tercer caso de prueba para el metodo cuentaCifras.
     */
    public void testCuentaCifras3() {
        System.out.println("Tercer caso de prueba para el metodo cuentaCifras");
        int num = -250;
        int resultado = Calculador.cuentaCifras(num);
        assertEquals(expResult, resultado);
    }
    /**
     * Cuarto caso de prueba para el metodo cuentaCifras.
     */
    public void testCuentaCifras4() {
        System.out.println("Cuarto caso de prueba para el metodo cuentaCifras");
        int num = -1222333444;
        int resultado = Calculador.cuentaCifras(num);
        assertEquals(expResult, resultado);
    }
}

```

**Output (NetBeans IDE 12.5):**

```

Tarea3_Nelson (debug) > Debugger Console <
=====
D E B U G   C O N F I G U R A T I O N
=====
Nº dígitos del valor es 1
Nº dígitos del valor 2167483647 es 10
Nº dígitos del valor -250 es 3
2 dígitos a 8 es 256
BUILD SUCCESSFUL (total time: 2 minutes 24 seconds)

```

6. Ejecuta la aplicación de prueba, visualizando la consola y el Panel de Resultados. No deberían aparecer defectos en la implementación de la clase Calculador.

Como se puede ver en la captura, los test son válidos.

The screenshot displays a dual-task interface. On the left, a web browser window for 'LE-Entornos de desarrollo DAW E' shows course information and navigation links. On the right, an Apache NetBeans IDE window titled 'Tarea3\_Nelson - Apache NetBeans IDE 12.5' shows the source code for a Java application and its corresponding test class. The 'Test Results' tab in the IDE indicates that all tests have passed successfully, with detailed logs of each test execution.

Ahora fuerza algún error en algún método, y ejecuta algún caso de prueba en el que se vea que detecta el error. En este caso, he creado un cuarto test para el método “potencia”, en el que uso la base 0 y el exponente 100, y fuerzo el error de esperar el resultado con valor 1. El test genera un fallo, ya que esperaba un 1 y se obtuvo un 0.

This screenshot shows the same dual-task setup as the previous one. The browser window remains the same. In the NetBeans IDE, the 'Test Results' window now shows a failed test named 'testPotencia()'. The log output highlights the discrepancy between the expected result (1) and the actual result (0), indicating a failure in the power method's implementation for the specific input values tested.