

# **PROYECTO DE ESTRUCTURA SELECTIVA DOBLE**

A continuación resolveremos el siguiente proyecto.

## **LABORATORIONº03-EJERCICIO01**

Determinar si un alumno aprueba a desaprueba un curso, conociendo que aprobará si su promedio de tres calificaciones es mayor o igual a 10.5; desaprobadado en caso contrario.

Para resolver este ejercicio abriremos un nuevo proyecto.



Figura 01: Eleccion de un nuevo proyecto



Figura 02: Eleccion de un proyecto Java Application

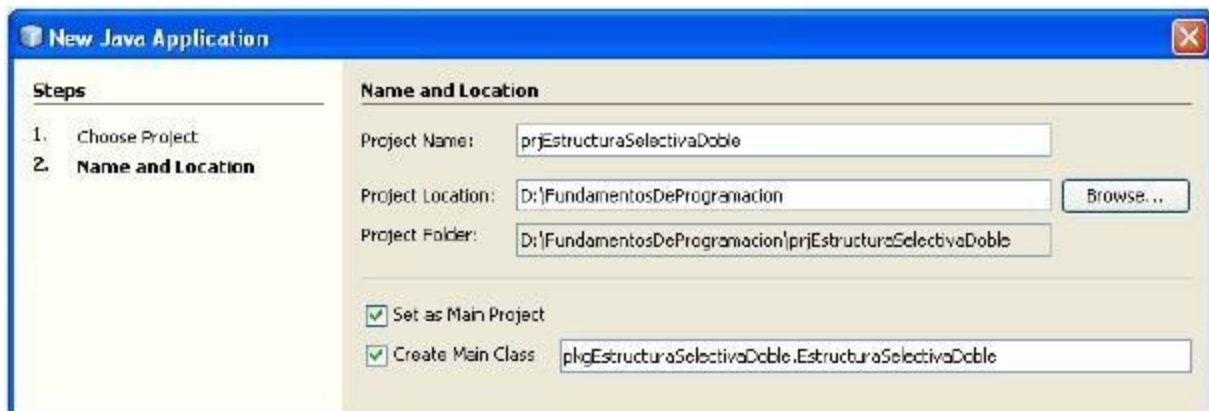


Figura 03: Ingresar nombre del proyecto, paquete y de la clase principal

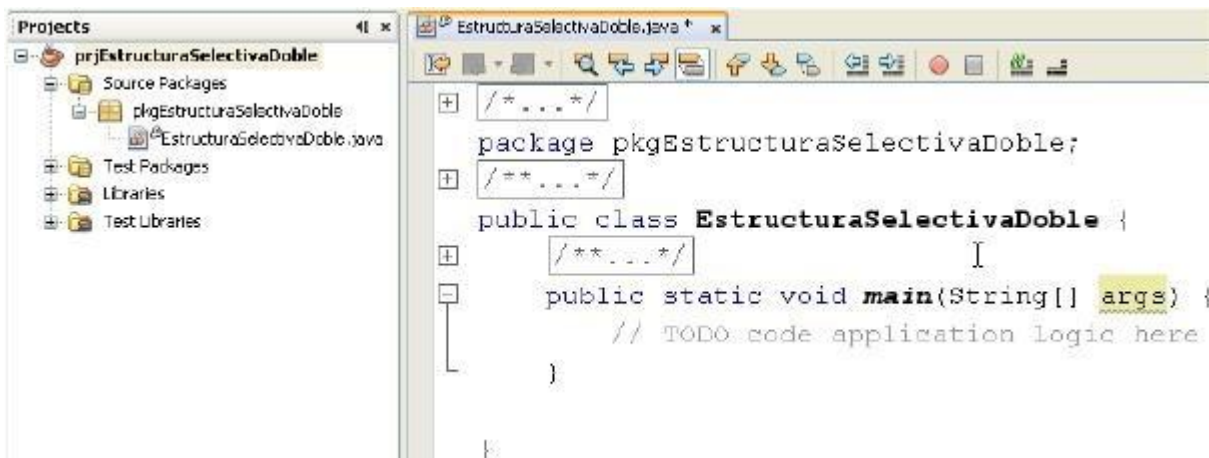


Figura 04: Proyecto listo para programar

Ahora estamos listo para comenzar con la programación de este ejercicio. Antes hay que recordar como esta resuelto en algoritmo.

```

algoritmo Lab3Ejercicio01
var
    entero : calif1, calif2, calif3
    real : prom

inicio
    Leer calif1, calif2, calif3
    prom = (calif1 + calif2 + calif3)/3
    si prom >= 10.5 entonces
        Escribir 'ALUMNO APROBADO'
    si no
        Escribir 'ALUMNO DESAPROBADO'
    fin_si

fin
  
```

La solución en código de este ejercicio es el siguiente:

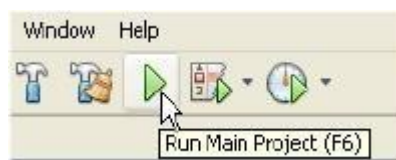
```
package pkgEstructuraSelectivaDoble;
import javax.swing.JOptionPane;

public class EstructuraSelectivaDoble {

    public static void main(String[] args) {
        int calif1, calif2, calif3;
        double prom;
        calif1=Integer.parseInt(JOptionPane.showInputDialog("Ingrese 1era Calificación"));
        calif2=Integer.parseInt(JOptionPane.showInputDialog("Ingrese 2da Calificación"));
        calif3=Integer.parseInt(JOptionPane.showInputDialog("Ingrese 3era Calificación"));
        prom=(calif1 + calif2 + calif3)/3.0;
        if(prom>=10.5)
            JOptionPane.showMessageDialog(null,"ALUMNO APROBADO");
        else
            JOptionPane.showMessageDialog(null,"ALUMNO DESAPROBADO");
    }
}
```



Comprendido el código de la estructura selectiva doble en Java, ahora pasamos a ejecutar el programa para ver los resultados obtenidos.



Si ingresamos los siguientes datos:



Entrada

? Ingrese 1era Calificación

14

Aceptar Cancelar



Entrada

? Ingrese 2da Calificación

19

Aceptar Cancelar



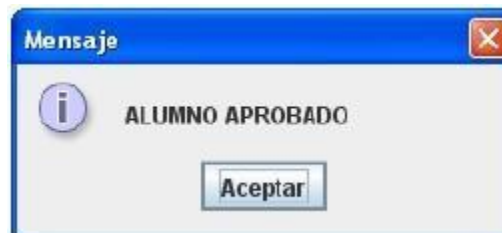
Entrada

? Ingrese 3era Calificación

17

Aceptar Cancelar

Debe de obtenerse el siguiente resultado:



Mensaje

i ALUMNO APROBADO

Aceptar

**Dato adicional:**

Estimados estudiantes pueden ustedes mejorar el código a manera de práctica y mostrar en el mismo mensaje de aprobado o desaprobado la nota promedio obtenida.



Mensaje

i ALUMNO APROBADO, NOTA PROMEDIO = 16.666666666666668

Aceptar

Mensaje

i ALUMNO DESAPROBADO, NOTA PROMEDIO = 8.0

Aceptar

# ESTRUCTURA SELECTIVA MÚLTIPLE

A continuación resolveremos el siguiente proyecto de estructura selectiva múltiple.

## **EJERCICIO 03**

Realizar un algoritmo que ingrese número del 1 al 10 y lo muestre en letras.

Para resolver este ejercicio abriremos un nuevo proyecto.



Figura 01: Eleccion de un nuevo proyecto



Figura 02: Eleccion de un proyecto Java Application



Figura 03: Ingresar nombre del proyecto, paquete y de la clase principal

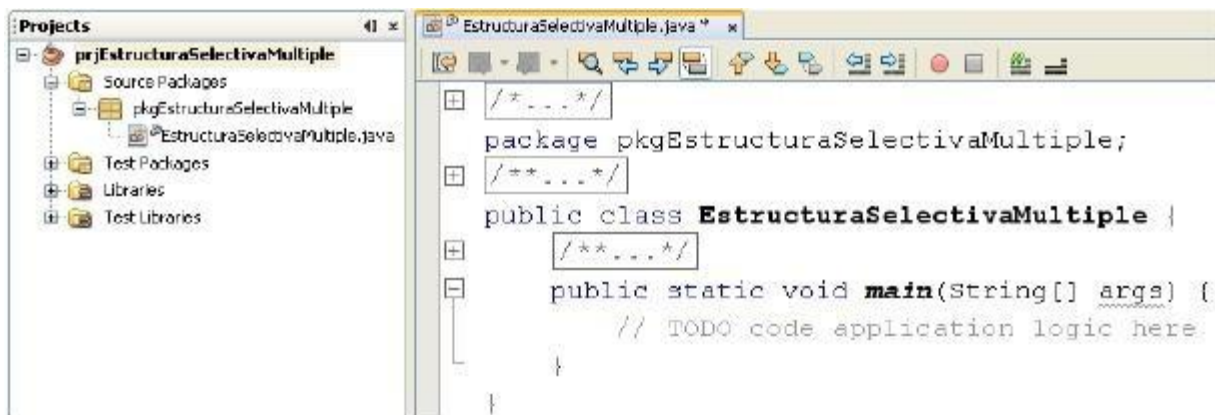


Figura 04: Proyecto listo para programar

Ahora estamos listo para comenzar con la programación de este ejercicio. Antes hay que recordar como esta resuelto en algoritmo.

#### Algoritmo Lab04Ejercicio03

**var**

entero: Numero

**inicio**

Leer Numero

en\_caso (Numero)

1 : Mostrar 'UNO'

2 : Mostrar 'DOS'

3 : Mostrar 'TRES'

4 : Mostrar 'CUATRO'

5 : Mostrar 'CINCO'

6 : Mostrar 'SEIS'

7 : Mostrar 'SIETE'

8 : Mostrar 'OCHO'

9 : Mostrar 'NUEVE'

10 : Mostrar 'DIEZ'

sino : Mostrar 'Numero fuera del rango establecido'

fin\_caso

**fin**

La solución en código de este ejercicio es el siguiente:

```
package pkgEstructuraSelectivaMultiple;
import javax.swing.JOptionPane;

public class EstructuraSelectivaMultiple {
    public static void main(String[] args) {
        int Numero;
        Numero=Integer.parseInt(JOptionPane.showInputDialog("Ingrese numero (1-10)"));
        switch(Numero)
        {
            case 1 : JOptionPane.showMessageDialog(null, "UNO");break;
            case 2 : JOptionPane.showMessageDialog(null, "DOS");break;
            case 3 : JOptionPane.showMessageDialog(null, "TRES");break;
            case 4 : JOptionPane.showMessageDialog(null, "CUATRO");break;
            case 5 : JOptionPane.showMessageDialog(null, "CINCO");break;
            case 6 : JOptionPane.showMessageDialog(null, "SEIS");break;
            case 7 : JOptionPane.showMessageDialog(null, "SIETE");break;
            case 8 : JOptionPane.showMessageDialog(null, "OCHO");break;
            case 9 : JOptionPane.showMessageDialog(null, "NUEVE");break;
            case 10: JOptionPane.showMessageDialog(null, "DIEZ");break;
            default: JOptionPane.showMessageDialog(null, "Numero fuera del rango establecido");
                    break;
        }
    }
}
```



```
package pkgEstructuraSelectivaMultiple;
import javax.swing.JOptionPane;
public class EstructuraSelectivaMultiple {
    public static void main(String[] args) {
        int Numero;
        Numero=Integer.parseInt(JOptionPane.showInputDialog("Ingrese numero (1-10)"));
        switch(Numero)
        {
            case 1 : JOptionPane.showMessageDialog(null, "UNO");break;
            case 2 : JOptionPane.showMessageDialog(null, "DOS");break;
            case 3 : JOptionPane.showMessageDialog(null, "TRES");break;
            case 4 : JOptionPane.showMessageDialog(null, "CUATRO");break;
            case 5 : JOptionPane.showMessageDialog(null, "CINCO");break;
            case 6 : JOptionPane.showMessageDialog(null, "SEIS");break;
            case 7 : JOptionPane.showMessageDialog(null, "SIETE");break;
            case 8 : JOptionPane.showMessageDialog(null, "OCHO");break;
            case 9 : JOptionPane.showMessageDialog(null, "NUEVE");break;
            case 10: JOptionPane.showMessageDialog(null, "DIEZ");break;
            default: JOptionPane.showMessageDialog(null, "Numero fuera del rango establecido");
                    break;
        }
    }
}
```



## ESTRUCTURA SELECTIVA MÚLTIPLE

En las estructuras selectivas múltiples solo se pueden evaluar variables de tipo entero:

```
int Numero;  
Numero=Integer.parseInt(JOptionPane  
switch(Numero)  
{  
    case 1 : JOptionPane.showMessageDialog
```

No se puede evaluar variables reales (double), de cadena (String) y ni de carácter (char).

La orden **break** hace que la ejecución salga del switch; de no ponerlo en cada case se ejecutarán todas las órdenes hacia abajo sin importar el valor de los case hasta encontrar un break o la llave de cierre del switch.

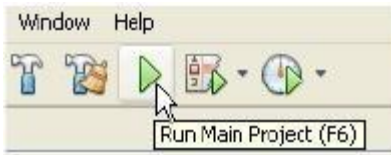
### Nota

Los ejercicios 1 y 2 del Laboratorio 4 no se podrán llevar a NetBeans igual como está en el pseudocódigo, porque estaríamos evaluando una **variable de tipo carácter**.

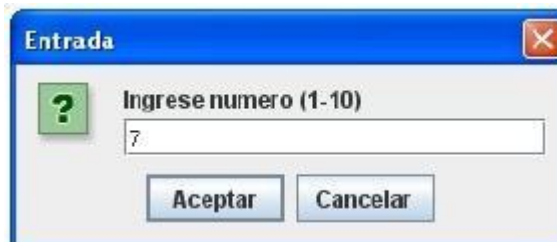
La realización de un algoritmo viene hacer la solución de un problema independiente del lenguaje de programación. En este caso el NetBeans solo evalua variables de valor tipo entero.

## EJECUCIÓN DEL PROGRAMA

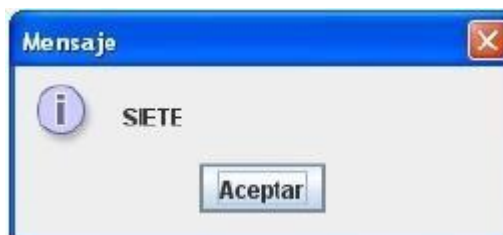
Ahora pasamos a ejecutar el programa para ver los resultados obtenidos.



Si ingresamos el siguiente dato:



Debemos de obtener el siguiente resultado:





# **PROYECTO DE ESTRUCTURA SELECTIVA ANIDADA**

A continuación resolveremos el siguiente proyecto.

## **EJERCICIO 01**

Leer un número entero y decir si es positivo, negativo o neutro.

Para resolver este ejercicio abriremos un nuevo proyecto.



Figura 01: Eleccion de un nuevo proyecto



Figura 02: Eleccion de un proyecto Java Application

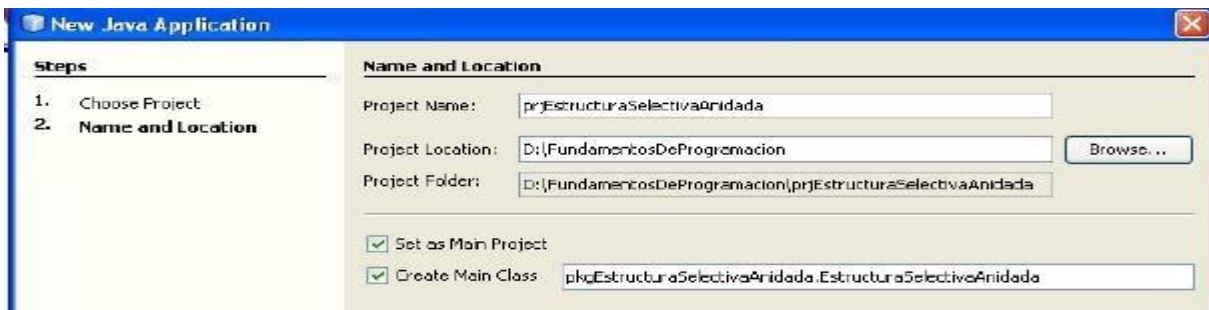


Figura 03: Ingresar nombre del proyecto, paquete y de la clase principal

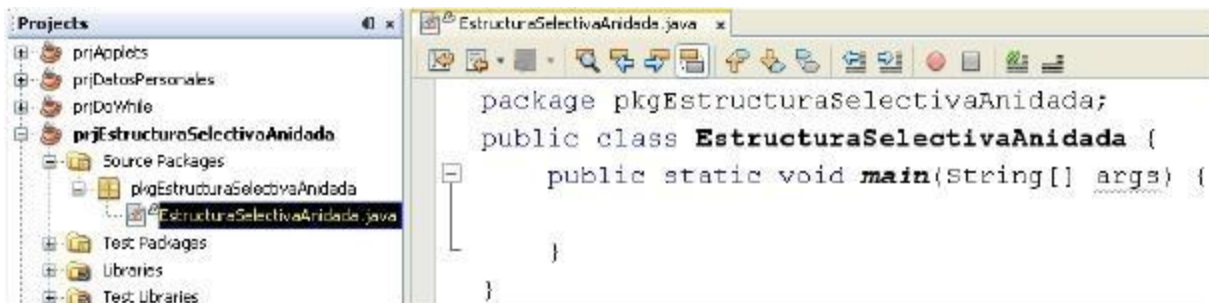


Figura 04: Proyecto listo para programar

Ahora estamos listo para comenzar con la programación de este ejercicio. Antes hay que recordar como esta resuelto en algoritmo.

```

algoritmo Lab5Ejercicio01
var
    entero : num
inicio
    Leer num
    si (num = 0) entonces
        Escribir 'NÚMERO NEUTRO'
    si no
        si (num > 0) entonces
            Escribir 'NÚMERO POSITIVO'
        sino
            Escribir 'NÚMERO NEGATIVO'
        fin_si
    fin_si
fin
    
```

La solución en código de este ejercicio es el siguiente:

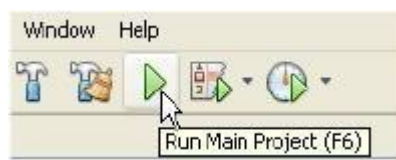
```
package pkgEstructuraSelectivaAnidada;
import javax.swing.JOptionPane;

public class EstructuraSelectivaAnidada {
    public static void main(String[] args) {
        int num;
        num=Integer.parseInt(JOptionPane.showInputDialog("Ingrese numero entero"));
        if(num==0)
            JOptionPane.showMessageDialog(null, "NUMERO NEUTRO");
        else
        {
            if(num>0)
                JOptionPane.showMessageDialog(null, "NUMERO POSITIVO");
            else
                JOptionPane.showMessageDialog(null, "NUMERO NEGATIVO");
        }
    }
}
```

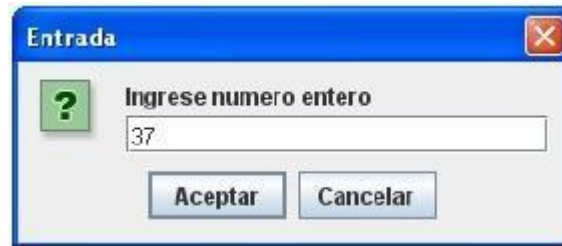
```
package pkgEstructuraSelectivaAnidada;
import javax.swing.JOptionPane;

public class EstructuraSelectivaAnidada {
    public static void main(String[] args) {
        int num;
        num=Integer.parseInt(JOptionPane.showInputDialog("Ingrese numero entero"));
        if(num==0)
            JOptionPane.showMessageDialog(null, "NUMERO NEUTRO");
        else
        {
            if(num>0)
                JOptionPane.showMessageDialog(null, "NUMERO POSITIVO");
            else
                JOptionPane.showMessageDialog(null, "NUMERO NEGATIVO");
        }
    }
}
```

Comprendido el código de la estructura selectiva anidada en Java, ahora pasamos a ejecutar el programa para ver los resultados obtenidos.

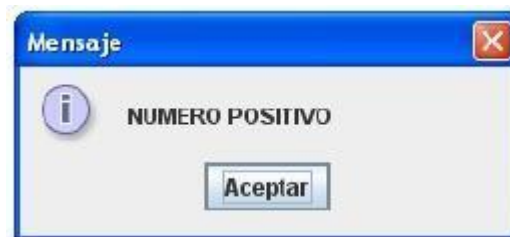


Si ingresamos el siguiente dato:



A screenshot of a Windows-style dialog box titled "Entrada" (Input). It has a blue title bar with a close button (X) in the top right corner. The main area has a light gray background. On the left, there is a green square icon with a white question mark. To its right, the text "Ingrese numero entero" (Enter integer number) is displayed. Below this text is a text input field containing the number "37". At the bottom of the dialog, there are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

Debe de obtenerse el siguiente resultado:



A screenshot of a Windows-style dialog box titled "Mensaje" (Message). It has a blue title bar with a close button (X) in the top right corner. The main area has a light gray background. On the left, there is a purple circular icon with a white lowercase 'i'. To its right, the text "NUMERO POSITIVO" (POSITIVE NUMBER) is displayed. At the bottom of the dialog, there is a single button: "Aceptar" (Accept).

También pueden obtenerse los siguientes resultados:



Three screenshots of dialog boxes are shown, illustrating different input scenarios. The first screenshot shows an "Entrada" dialog box with the input field containing "0". Below it is a "Mensaje" dialog box with the message "NUMERO NEUTRO" (NEUTRAL NUMBER). The second screenshot shows an "Entrada" dialog box with the input field containing "-87". Below it is a "Mensaje" dialog box with the message "NUMERO NEGATIVO" (NEGATIVE NUMBER). All dialog boxes have a blue title bar, a close button (X) in the top right corner, and a light gray main area. The "Entrada" boxes have a green question mark icon and "Aceptar" and "Cancelar" buttons. The "Mensaje" boxes have a purple 'i' icon and an "Aceptar" button.

# PROYECTO DE ESTRUCTURA REPETITIVA

## FOR

A continuación resolveremos el siguiente proyecto.

### EJERCICIO 01

Diseñar un algoritmo que permita visualizar la tabla de multiplicar de un número entero ingresado por teclado. Por ejemplo:

1 x 5 = 5  
2 x 5 = 10  
3 x 5 = 15  
.....  
12 x 5 = 60

Para resolver este ejercicio abriremos un nuevo proyecto.



Figura 01: Eleccion de un nuevo proyecto



Figura 02: Eleccion de un proyecto Java Application



Figura 03: Ingresar nombre del proyecto, paquete y de la clase principal

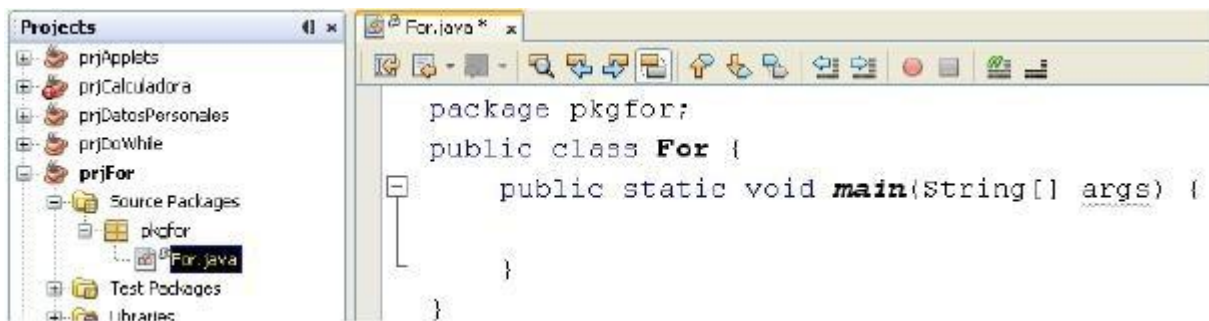


Figura 04: Proyecto listo para programar

Ahora estamos listo para comenzar con la programación de este ejercicio. Antes hay que recordar como esta resuelto en algoritmo.

```

algoritmo Lab06Ejercicio01
var
    entero : num, producto, i
    cadena: cad
inicio
    Leer num
    cad = ''
    si (num >0) entonces
        desde i =1 hasta 12 inc 1 hacer
            producto = i * num
            cad = cad, i , ' * ' , num, ' = ', producto
        fin_desde
        Mostrar (cad)
    sino
        Mostrar ('Error de ingreso...!! El Numero debe ser mayor que cero')
    fin_si
fin

```

La solución en código de este ejercicio es el siguiente:

```
package pkgfor;
import javax.swing.JOptionPane;

public class For {
    public static void main(String[] args) {
        int num, producto, i;
        String cad="";

        num=Integer.parseInt(JOptionPane.showInputDialog("Ingrese numero"));
        if(num>0)
        {
            for(i=1;i<=12;i++)
            {
                producto=num*i;
                cad=cad+i+"x "+num+"=" +producto+"\n";
            }
            JOptionPane.showMessageDialog(null, cad);
        }
        else
            JOptionPane.showMessageDialog(null, "Error de ingreso...!! El numero debe ser mayor que cero");
    }
}
```



```
package pkgfor;
import javax.swing.JOptionPane;
public class For {
    public static void main(String[] args) {
        int num, producto, i;
        String cad="";

        num=Integer.parseInt(JOptionPane.showInputDialog("Ingrese numero"));
        if(num>0)
        {
            for(i=1;i<=12;i++)
            {
                producto=num*i;
                cad=cad + i + " x " + num + " = " + producto + "\n";
            }
            JOptionPane.showMessageDialog(null, cad);
        }
        else
        {
            JOptionPane.showMessageDialog(null, "Error de ingreso...!! El numero debe ser mayor que cero");
        }
    }
}
```

Diagram illustrating the flow of execution for the provided Java code, with numbered markers (1-8) indicating key points:

- End of the `main` method.
- Input dialog box: `JOptionPane.showInputDialog("Ingrese numero")`.
- Conditional check: `if(num > 0)`.
- Start of the `for` loop: `for(i=1; i<=12; i++)`.
- Start of the loop body.
- Calculation and string building: `cad=cad + i + " x " + num + " = " + producto + "\n";`.
- Displaying the result: `JOptionPane.showMessageDialog(null, cad);`.
- Error handling: `JOptionPane.showMessageDialog(null, "Error de ingreso...!! El numero debe ser mayor que cero");`.

En este ejercicio de estructura repetitiva For nos mostrará la tabla de multiplicar de un número entero positivo.

### La explicación del programa paso a paso es el siguiente

1. Declaración de variables; en estas instrucciones también se pueden inicializar dichas variables. Inicializamos la variable **cad** = "".
2. Ingreso de los datos de entrada.
3. Esta instrucción es una condición para determinar:
  - 3.1. Si la condición es verdadera, quiere decir que el número es positivo y se ingresa al cuerpo del **if**. Luego se ejecuta el paso 4.
  - 3.2. Si la condición es falsa se salta al paso 8.
4. Para ingresar al bucle del For, se llevarán acabo las siguientes acciones:
  - 4.1. Si es la primera vez que se ejecuta la instrucción del For, a la variable inicial de tipo entero que se le asigna un valor; para este ejemplo: **i = 1**.
  - 4.2. Para ingresar al bucle se evaluará la condición; para este ejemplo: **i <= 12**, si es verdad se ingresa y se pasa al paso 5, de lo contrario se pasa al paso 7.
5. En la variable **producto** se almacenarán los valores obtenidos del producto de **num \* i**.
6. Esta instrucción es la mas importante porque en la variable **cad** almacenamos toda la tabla de multiplicar de un número **N**.
  - 6.1. Analizaremos primero esta parte del código: **cad + i + " \* " + num + " = " + producto**, la variable **cad** al comenzar el programa se le inicializo como una cadena vacia, a esta se le concatena el valor de la variable **i** (De 1 a 12 dependiendo de la interacción en que se encuentra), y tambien se le concatena el símbolo \*, el valor de la variable **num**, el símbolo = y el valor de la variable **producto** obtenido en el paso 5. Formando así una cadena de la siguiente forma, para este ejemplo: **1 x 5 = 5**
  - 6.2. Esta parte de la instrucción: **+ "\n"** significa que a la cadena obtenida se le va a concatenar un salto de línea. Esto hace que el resultado que se va a mostrar salga línea por línea.
  - 6.3. Toda esta cadena concatenada se almacena en la variable **cad**, que nos seguirá sirviendo en las iteraciones del bucle como repositorio de toda la tabla de multiplicar.
  - 6.4. Terminado de realizar todas las instrucciones del cuerpo del For, se pasa al paso 4 (acción 4.2) pero antes se incrementa o decrementa la variable **i** un valor constante, esto depende de la instrucción del For, para este ejemplo: **i++**, significa que se incrementará el valor de uno en uno.
7. Se muestra el valor almacenado en la variable **cad**, que viene hacer la tabla de multiplicar de un número **N**.
8. Muestra el mensaje de "Error de ingreso..!!" por ser un número negativo y se acaba el programa.

Comprendido el código de la estructura repetitiva For en Java, ahora pasamos a ejecutar el programa para ver los resultados obtenidos.



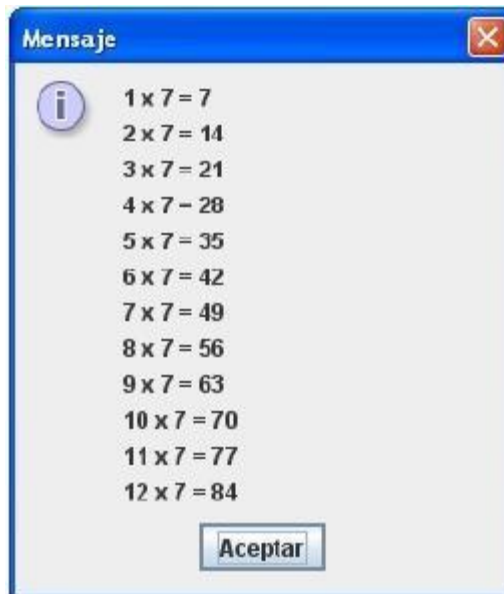
### 1ercaso

Ingresaremos un numero entero positivo:



A screenshot of a Windows-style dialog box titled "Entrada" (Input). It has a blue title bar with a close button (X) in the top right corner. Inside the dialog, there is a green square icon with a white question mark on the left. To its right, the text "Ingrese numero" (Enter number) is displayed above a text input field. The input field contains the number "7". Below the input field, there are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

Debe de obtenerse el siguiente resultado:



A screenshot of a Windows-style dialog box titled "Mensaje" (Message). It has a blue title bar with a close button (X) in the top right corner. Inside the dialog, there is a purple circular icon with a white lowercase 'i' on the left. To its right, a list of multiplication facts for the number 7 is displayed:  $1 \times 7 = 7$ ,  $2 \times 7 = 14$ ,  $3 \times 7 = 21$ ,  $4 \times 7 = 28$ ,  $5 \times 7 = 35$ ,  $6 \times 7 = 42$ ,  $7 \times 7 = 49$ ,  $8 \times 7 = 56$ ,  $9 \times 7 = 63$ ,  $10 \times 7 = 70$ ,  $11 \times 7 = 77$ , and  $12 \times 7 = 84$ . At the bottom right of the dialog, there is an "Aceptar" (Accept) button.

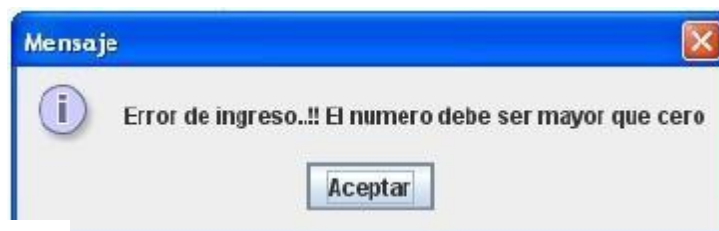
### 2docaso

Ingresaremos un numero entero negativo:



A screenshot of a Windows-style dialog box titled "Entrada" (Input). It has a blue title bar with a close button (X) in the top right corner. Inside the dialog, there is a green square icon with a white question mark on the left. To its right, the text "Ingrese numero" (Enter number) is displayed above a text input field. The input field contains the number "-7". Below the input field, there are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

Debe de obtenerse el siguiente resultado:



A screenshot of a Windows-style dialog box titled "Mensaje" (Message). It has a blue title bar with a close button (X) in the top right corner. Inside the dialog, there is a purple circular icon with a white lowercase 'i' on the left. To its right, an error message is displayed: "Error de ingreso...!! El numero debe ser mayor que cero" (Input error...!! The number must be greater than zero). At the bottom right of the dialog, there is an "Aceptar" (Accept) button.

# **PROYECTO DE ESTRUCTURA REPETITIVA** **DO WHILE**

A continuación resolveremos el siguiente proyecto.

## **EJERCICIO 01**

Realizar un programa para un Supermercado que calcule el total a pagar de un cliente, por la compra de varios productos, el ingreso de las compras debe continuar si digitamos el valor 1, y 2 se termina el ingreso y muestra el resultado.

Para resolver este ejercicio abriremos un nuevo proyecto.



Figura 01: Eleccion de un nuevo proyecto



Figura 02: Eleccion de un proyecto Java Application



Figura 03: Ingresar nombre del proyecto, paquete y de la clase principal

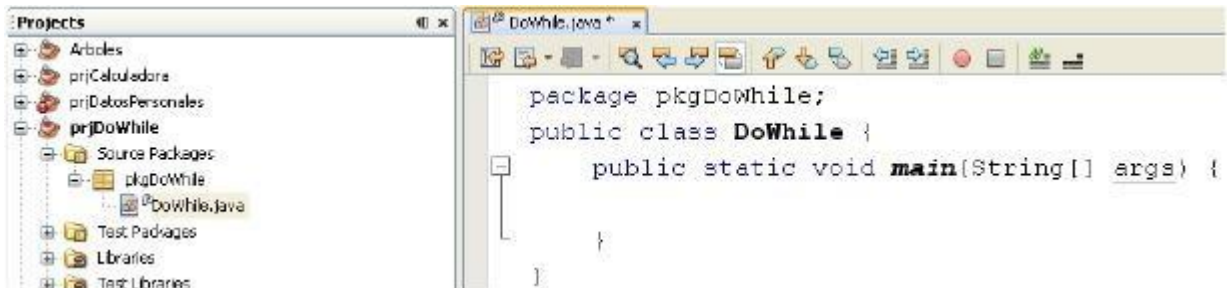


Figura 04: Proyecto listo para programar

Ahora estamos listo para comenzar con la programación de este ejercicio. Antes hay que recordar como esta resuelto en algoritmo.

algoritmo Ejercicio01

**var**

entero : CantProducto, opcion, cont  
real : PrecioProducto, compra

**inicio**

compra=0  
cont = 0

Hacer

cont = cont + 1

Leer PrecioProducto, CantProducto

compra = compra + PrecioProducto \* CantProducto

Escribir 'Desea continuar [si=1 , no=2] : '

Leer opcion

Mientras (opcion = 1)

Mostrar ('La compra total por ', cont, ' productos es : ', compra)

**fin**

La solución en código de este ejercicio es el siguiente:

```
package pkgDdWhile;
import javax.swing.JOptionPane;
public class DdWhile {
    public static void main(String[] args) {
        int cantProducto,opcion,cont;
        double precioProducto,compra;

        compra=0;cont=0;
        do
        {
            //contador de productos ingresados
            cont=cont+1;
            //Ingreso de cantidad y precio del producto a comprar
            cantProducto=Integer.parseInt(JOptionPane.showInputDialog("Ingrese Cantidad del Producto " + cont));
            precioProducto=Double.parseDouble(JOptionPane.showInputDialog("Ingrese Precio Producto " + cont));
            //Obtencion del subtotal de la compra de uno o mas productos
            compra=compra+cantProducto*precioProducto;

            //Pregunta de si DESEA SEGUIR ingresando al bucle para seguir
            //acumulando el subtotal de los productos comprados.
            opcion=Integer.parseInt(JOptionPane.showInputDialog("Desea continuar[Si=1 , No=2]"));
        }while(opcion==1);
        JOptionPane.showMessageDialog(null, "El total a pagar por la compra de " + cont + " productos" +
            "nes : " + compra);
    }
}
```





```

package pkgDoWhile;
import javax.swing.JOptionPane;
public class DoWhile {
    public static void main(String[] args) {
        int cantProducto,opcion,cont;
        double precioProducto,compra;

        compra=0;cont=0;
        do
        {
            //contador de productos ingresados
            cont=cont+1;
            //Ingreso de cantidad y precio del producto a comprar
            cantProducto=Integer.parseInt(JOptionPane.showInputDialog("Ingrese Cantidad del Producto " + cont))
            precioProducto=Double.parseDouble(JOptionPane.showInputDialog("Ingrese Precio Producto " + cont));
            //Obtencion del subtotal de la compra de uno o mas productos
            compra=compra + cantProducto * precioProducto;

            //Pregunta de si DESEA SEGUIR ingresando al bucle para seguir
            //acumulando el subtotal de los productos comprados.
            opcion=Integer.parseInt(JOptionPane.showInputDialog("Desea continuar[Si=1 , No=2]"));
        }while(opcion==1);
        JOptionPane.showMessageDialog(null, "El total a pagar por la compra de " + cont + " productos" +
            "\nes : " + compra);
    }
}

```

The flowchart illustrates the execution of the provided Java code. It begins at node 1, which points to the declaration of variables `int cantProducto,opcion,cont;` and `double precioProducto,compra;`. Node 2 points to the initialization `compra=0;cont=0;`. Node 3 points to the start of the `do` loop. Node 4 points to the increment `cont=cont+1;`. Node 5 points to the input for quantity `cantProducto=Integer.parseInt(...)`. Node 6 points to the input for price `precioProducto=Double.parseDouble(...)`. Node 7 points to the calculation of the subtotal `compra=compra + cantProducto * precioProducto;`. Node 8 points to the decision point `opcion=Integer.parseInt(...)` and the `while` loop condition `while(opcion==1);`. Node 9 points to the final output message `JOptionPane.showMessageDialog(...)`.

En este ejercicio de estructura repetitiva Do While podemos ingresar un sin número de compras siempre y cuando digitemos el valor “1” a la consulta “Desea continuar [si=1 , no=2] : ”, si la respuesta es positiva el programa va acumulando las compras subtotales en la variable **compra**. Recordemos que una compra viene hacer el producto de la cantidad y el precio de un producto, es decir:

$$\text{compra} = \text{PrecioProducto} * \text{CantProducto}$$

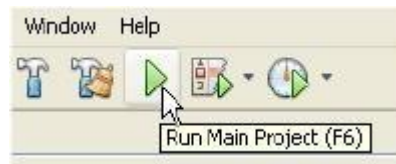
Ahora la instrucción que me permite acumular los subtotales de las compras es:

$$\text{compra} = \text{compra} + \text{PrecioProducto} * \text{CantProducto}$$

### La explicación del programa paso a paso es el siguiente

1. Declaración de variables.
2. Se inicializa las variables **compra** y **cont** con un valor igual a cero.
3. **Do**, instrucción que determina el comienzo del bucle Do While.
4. Esta instrucción nos sirve de contador de cuantas compras vamos realizando.
5. Ingreso de los datos de entrada de la 1era compra.
6. Esta instrucción se ejecuta de la siguiente manera:
  - 8.1. Se multiplica el **PrecioProducto \* CantProducto**
  - 8.2. El resultado se suma con el valor de la variable compra, esta variable fue inicializado con cero al comenzar el programa, es decir:
$$0 + \text{PrecioProducto} * \text{CantProducto}$$
  - 8.3. El resultado de toda esta instrucción se almacena en la misma variable **compra**, que nos servirá como variable que va acumulando los subtotales de las compras.
7. En esta parte el programa nos pregunta si deseamos seguir ingresando compras.
8. Se evalua la condición del While, aquí pueden pasar dos cosas:
  - 8.1. Si la condición es **verdadera** vuelve a ingresar al bucle y se continua con:
    - a. El paso 4, se incrementa el contador.
    - b. El paso 5; ingresamos los valores de una segunda compra.
    - c. El paso 6; se suma el valor de la primera compra que se tenia almacenado en la variable **compra** con el producto de los datos de entrada de la segunda compra, almacenando nuevamente el resultado en la variable **compra**.
    - d. Se realiza el paso 7.
    - e. Se realiza el paso 8, si vuelve a ingresar al bucle se incrementa el contador, se ingresan los datos de la 3era compra y asi sucesivamente hasta llegar al paso 9.
  - 8.2. Si la condición es **falso** se continua con el paso 9.
9. Se muestra el resultado total de los **N** productos comprados.

Comprendido el código de la estructura repetitiva Do While en Java, ahora pasamos a ejecutar el programa para ver los resultados obtenidos.



Ingresaremos los datos de tres compras:

### 1eracompra

A dialog box titled 'Entrada' with a blue header bar and a red close button. It contains a green question mark icon, the text 'Ingrese Cantidad del Producto 1', a text input field with the value '3', and two buttons at the bottom: 'Aceptar' and 'Cancelar'.A dialog box titled 'Entrada' with a blue header bar and a red close button. It contains a green question mark icon, the text 'Ingrese Precio Producto 1', a text input field with the value '23.5', and two buttons at the bottom: 'Aceptar' and 'Cancelar'.A dialog box titled 'Entrada' with a blue header bar and a red close button. It contains a green question mark icon, the text 'Desea continuar[Si=1 , No=2]', a text input field with the value '1', and two buttons at the bottom: 'Aceptar' and 'Cancelar'.

### 2dacompra

A dialog box titled 'Entrada' with a blue header bar and a red close button. It contains a green question mark icon, the text 'Ingrese Cantidad del Producto 2', a text input field with the value '5', and two buttons at the bottom: 'Aceptar' and 'Cancelar'.A dialog box titled 'Entrada' with a blue header bar and a red close button. It contains a green question mark icon, the text 'Ingrese Precio Producto 2', a text input field with the value '20', and two buttons at the bottom: 'Aceptar' and 'Cancelar'.A dialog box titled 'Entrada' with a blue header bar and a red close button. It contains a green question mark icon, the text 'Desea continuar[Si=1 , No=2]', a text input field with the value '1', and two buttons at the bottom: 'Aceptar' and 'Cancelar'.

### 3eracompra



Entrada

? Ingrese Cantidad del Producto 3

8

Aceptar Cancelar

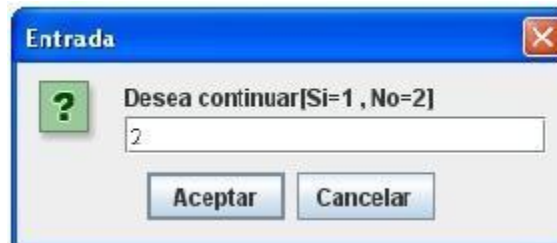


Entrada

? Ingrese Precio Producto 3

2.5

Aceptar Cancelar



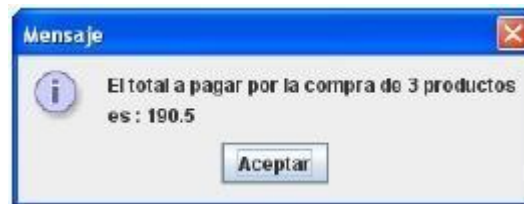
Entrada

? Desea continuar[Si=1, No=2]

2

Aceptar Cancelar

Debe de obtenerse el siguiente resultado:



Mensaje

i El total a pagar por la compra de 3 productos es: 190.5

Aceptar

# PROYECTO DE ESTRUCTURA REPETITIVA

## WHILE

A continuación resolveremos el siguiente proyecto.

### **EJERCICIO 01**

Diseñar un algoritmo que permita visualizar la tabla de multiplicar de un número entero ingresado por teclado. Por ejemplo:

1 x 5 = 5  
2 x 5 = 10  
3 x 5 = 15  
.....  
12 x 5 = 60

Para resolver este ejercicio abriremos un nuevo proyecto.



Figura 01: Eleccion de un nuevo proyecto



Figura 02: Eleccion de un proyecto Java Application

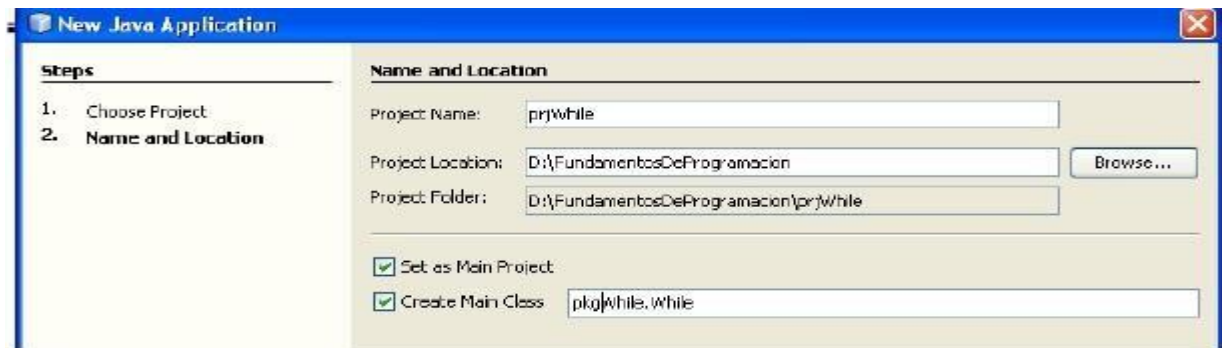


Figura 03: Ingresar nombre del proyecto, paquete y de la clase principal

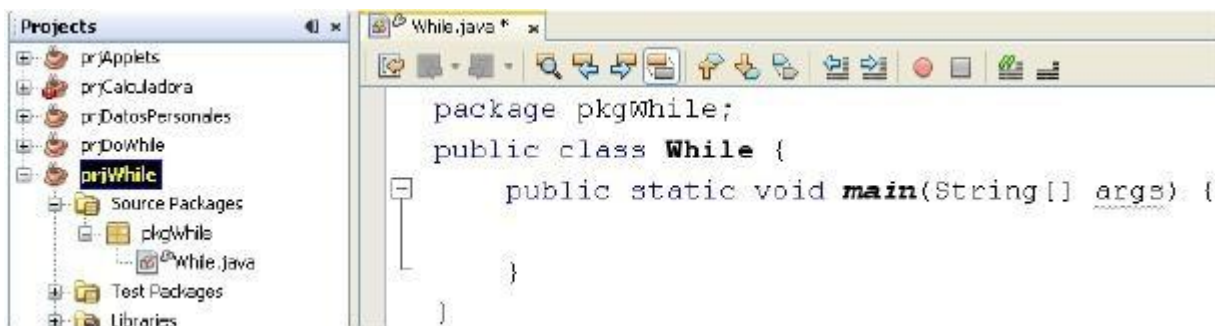


Figura 04: Proyecto listo para programar

Ahora estamos listo para comenzar con la programación de este ejercicio. Antes hay que recordar como esta resuelto en algoritmo.

```

algoritmo Lab06Ejercicio01
var
    entero : num, producto, i
    cadena: cad
inicio
    Leer num
    i=1
    cad = ''
    si (num >0) entonces
        mientras(i <=12)
            producto = i * num
            cad = cad, i, ' * ', num, ' = ', producto
            i=i+1
        fin_mientras
        Mostrar (cad)
    sino
        Mostrar ('Error de ingreso....!! El Numero debe ser mayor que cero')
    fin_si
fin
  
```

---

La solución en código de este ejercicio es el siguiente:

```
package pkg.While;
import javax.swing.JOptionPane;
public class While {
    public static void main(String[] args) {
        int num, producto, i=1;
        String cad="";

        num=Integer.parseInt(JOptionPane.showInputDialog("Ingrese numero"));
        if(num>0)
        {
            while(i<=12)
            {
                producto=num*i;
                cad=cad+i+"x"+num+"="+producto+"\n";
                i++;
            }
            JOptionPane.showMessageDialog(null, cad);
        }
        else
            JOptionPane.showMessageDialog(null, "Error de ingreso...!! El numero debe ser mayor que cero");
    }
}
```



```
package pkgWhile;
import javax.swing.JOptionPane;
public class While {
    public static void main(String[] args) {
        int num, producto, i=1; } ← 1
        String cad="";

        num=Integer.parseInt(JOptionPane.showInputDialog("Ingrese numero")); ← 2
        if(num>0) ← 3
        {
            while(i<=12) ← 4
            {
                producto= i * num; ← 5
                cad=cad + i + " x " + num + " = " + producto + "\n"; ← 6
                i=i+1; ← 7
            }
            JOptionPane.showMessageDialog(null, cad); ← 8
        }
        else
            JOptionPane.showMessageDialog(null, "Error de ingreso...!! El numero debe ser mayor que cero");
    }
} ← 9
```

En este ejercicio de estructura repetitiva While nos mostrará la tabla de multiplicar de un número entero positivo.

### La explicación del programa paso a paso es el siguiente

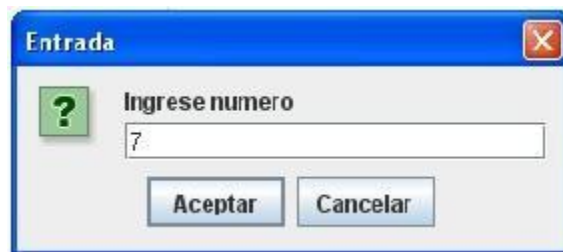
1. Declaración de variables; en estas instrucciones también se pueden inicializar dichas variables. Inicializamos las variables **i = 1** y **cad = ""**.
2. Ingreso de los datos de entrada.
3. Esta instrucción es una condición para determinar:
  - 3.1. Si la condición es verdadera, quiere decir que el número es positivo y se ingresa al cuerpo del **if**. Luego se ejecuta el paso 4.
  - 3.2. Si la condición es falsa se salta al paso 9.
4. Se ingresa al bucle While siempre y cuando la condición sea verdadera. Si la condición es verdadera se sigue al paso 5, de lo contrario se pasa al paso 8.
5. En la variable **producto** se almacenarán los valores obtenidos del producto de **num \* i**.
6. Esta instrucción es la mas importante porque en la variable **cad** almacenamos toda la tabla de multiplicar de un número **N**.
  - 6.1. Analizaremos primero esta parte del código: **cad + i + " \* " + num + " = " + producto**, la variable **cad** al comenzar el programa se le inicializo como una cadena vacia, a esta se le concatena el valor de la variable **i** (De 1 a 12 dependiendo de la interacción en que se encuentra), y tambien se le concatena el símbolo **\***, el valor de la variable **num**, el símbolo **=** y el valor de la variable **producto** obtenido en el paso 5. Formando así una cadena de la siguiente forma, por ejemplo: **1 x 5 = 5**
  - 6.2. Esta parte de la instrucción: **+ "\n"** significa que a la cadena obtenida se le va a concatenar un salto de línea. Esto hace que el resultado que se va a mostrar salga línea por línea.
  - 6.3. Toda esta cadena concatenada se almacena en la variable **cad**, que nos seguirá sirviendo en las iteraciones del bucle como repositorio de toda la tabla de multiplicar.
7. Se va incrementando el valor de la variable **i**, que es la que en algún momentos nos permitirá salir del bucle While. Al terminar esta instrucción se regresa al paso 4.
8. Se muestra el valor almacenado en la variable **cad**, que viene hacer la tabla de multiplicar de un número **N**.
9. Muestra el mensaje de "Error de ingreso...!" por ser un número negativo y se acaba el programa.

Comprendido el código de la estructura repetitiva While en Java, ahora pasamos a ejecutar el programa para ver los resultados obtenidos.

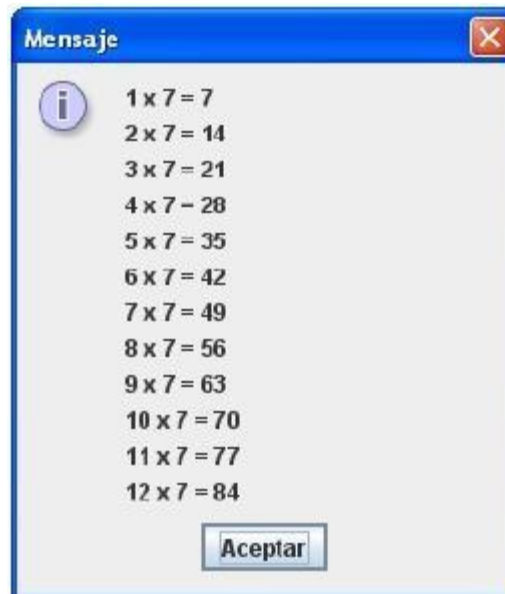


### 1er caso

Ingresaremos un numero entero positivo:

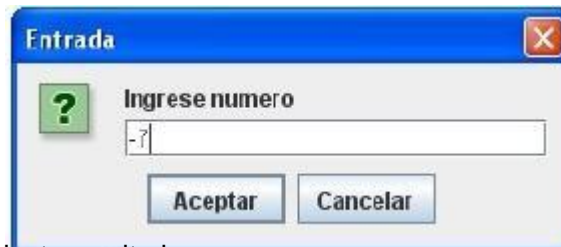


Debe de obtenerse el siguiente resultado:



## 2docaso

Ingresaremos un numero entero negativo:



Debe de obtenerse el siguiente resultado:

