Argentina programa 4.0

Introducción a Algoritmos y Java Parte II

"Desarrollador Java Inicial"

Agenda

Argentina programa 4.0

- Repaso de algoritmo y programa
- Estructuras de control de flujo
  - Estructura secuencial
  - Estructura condicional
  - Estructura repetitiva
- Prueba de ejecución
- Ejercicios

Repaso de algoritmo y programa

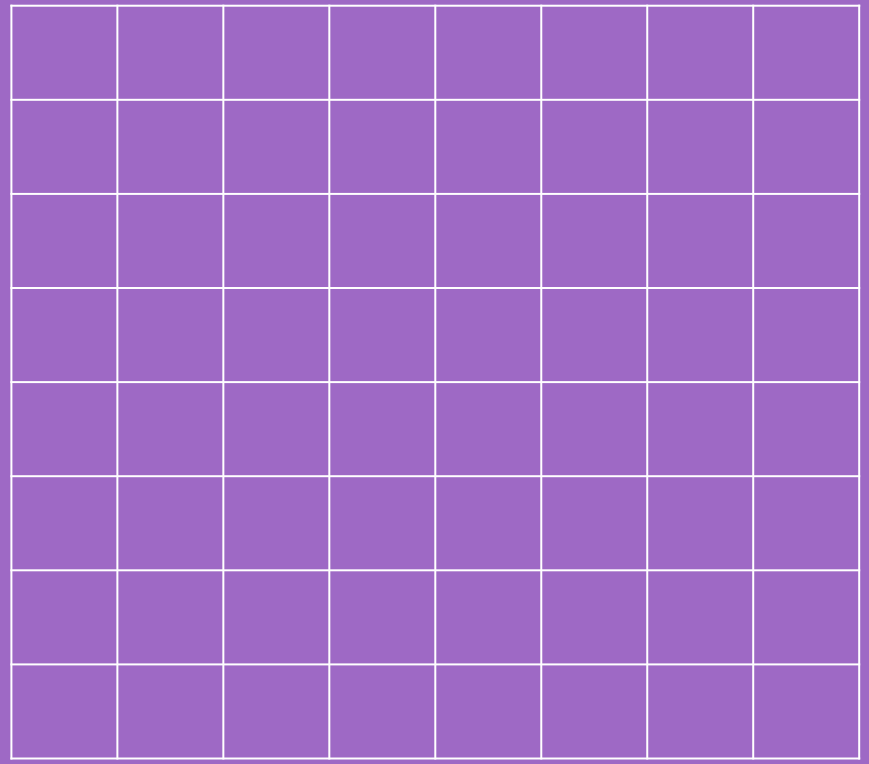
Planteo del algoritmo

- Comenzar identificando los resultados pedidos, porque así quedan claros los objetivos a cumplir.
- Luego individualizar los datos con que se cuenta y determinar si con estos es suficiente para llegar a los resultados pedidos.
- Finalmente si los datos son completos y los objetivos claros, se intentan plantear los procesos necesarios para convertir esos datos en los resultados esperados.

Programa

- Es el algoritmo codificado o escrito en un lenguaje de programación.

ESTRUCTURAS DE CONTROL



Estructuras de control de flujo

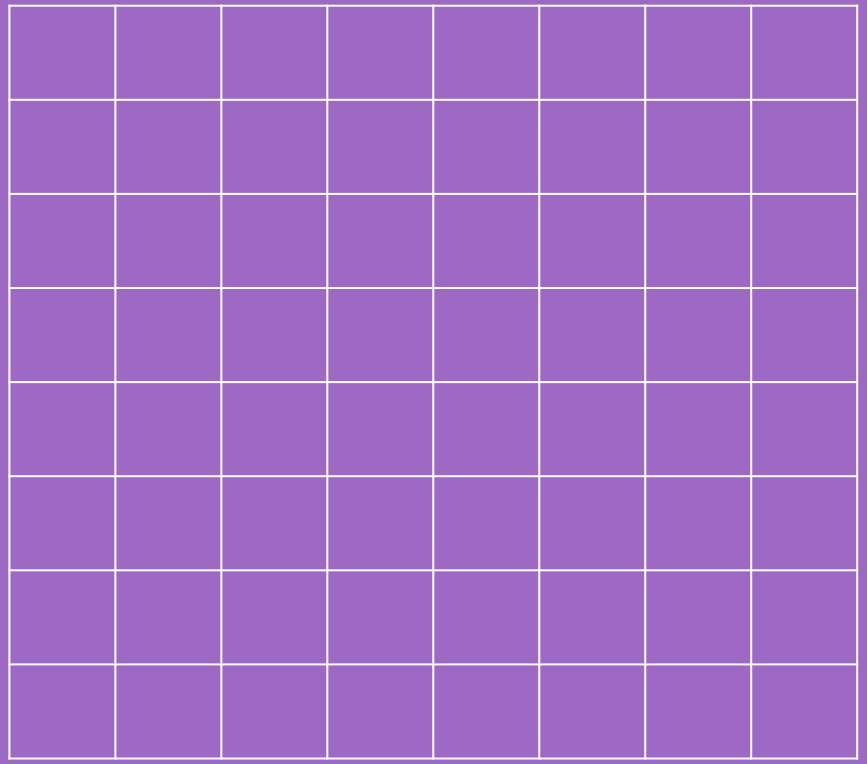
Argentina programa 4.0

Control de flujo: orden en que las llamadas a las instrucciones son ejecutadas o evaluadas.

Estructuras:

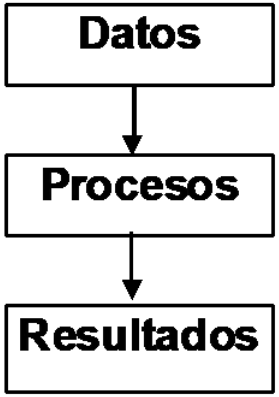
- Secuenciales
- Condicionales
- Repetitivas

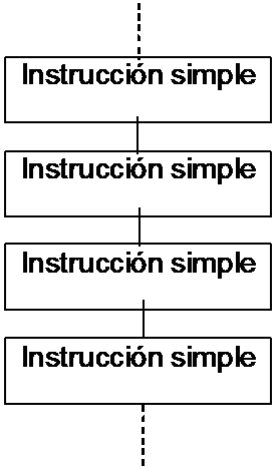
ESTRUCTURA SECUENCIAL

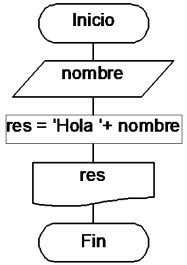


Estructura secuencial

Argentina programa 4.0

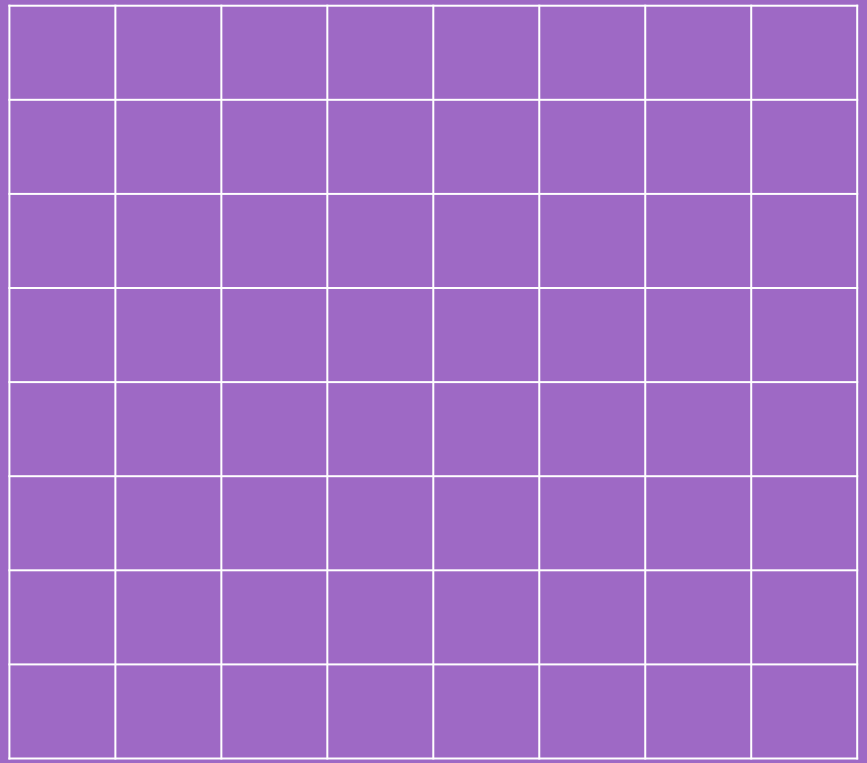







```java
/*1. Ingresar un nombre por teclado y mostrarlo por pantalla*/

package clase1;
import java.util.Scanner;

public class Ejercicio1_1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        // Solicita cargar un nombre
        System.out.print("Ingrese un nombre: ");
        // Entrada de una cadena
        String nombre = sc.nextLine();
        // Muestra el saludo con el nombre
        System.out.println("Hola " + nombre);
    }
}
```

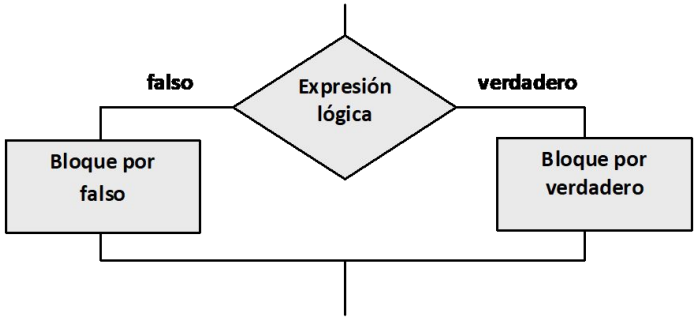
ESTRUCTURA CONDICIONAL



Estructura condicional

Argentina programa 4.0

- Establece qué instrucciones deben de ejecutarse o no, en función del valor de una condición o expresión lógica.



Sintaxis Java

```
if (expresión lógica)
{
    instrucciones por verdadero
}
else
{
   instrucciones por falso
}
```

Estructura condicional

Argentina programa 4.0

- Expresión lógica: Da como resultado un valor booleano (True o False), se puede combinar con operadores de relación y lógicos.

Tabla de verdad del conector *and*		
p	q	p and q
True	True	True
True	False	False
False	True	False
False	False	False

Tabla de verdad del conector *or*		
p	q	p or q
True	True	True
True	False	True
False	True	True
False	False	False

Tabla de verdad del conector *not*	
p	not p
True	False
False	True

Estructura condicional

Argentina programa 4.0

- Expresión lógica: Da como resultado un valor booleano (True o False), se puede combinar con operadores de relación y lógicos.

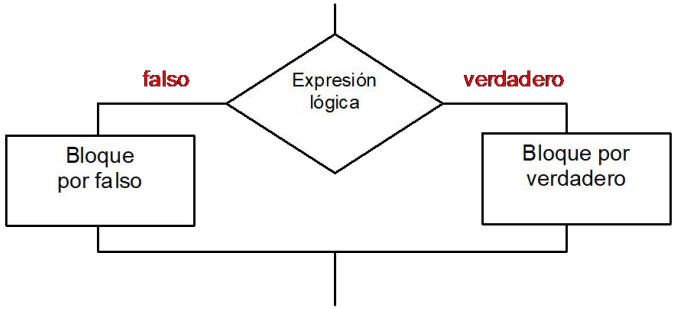
and	`if (sueldo > 20000 && antiguedad >= 10)`
	`  System.out.println("Crédito concedido")`
	`else`
	`  System.out.println("Crédito rechazado")`

or	`if (opcion == 1 || opcion == 3 || opcion == 5)`
	`  System.out.println("Opción correcta")`
	`else`
	`  System.out.println("Opción incorrecta")`

not	`if (!aprobo)`
	`  System.out.println("No aprobó el examen")`
	`else`
	`  System.out.println("Aprobó el examen")`

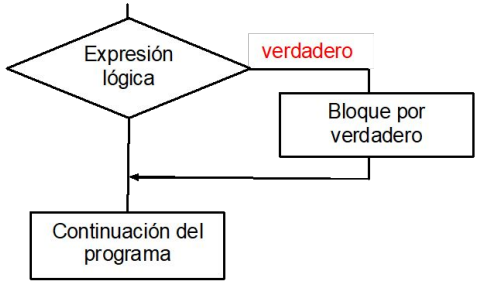
Variantes de la expresión condicional

- Doble




```
if (expresion logica)
{
    instrucciones por verdadero
}
else
{
   instrucciones por falso
}
```

- Simple

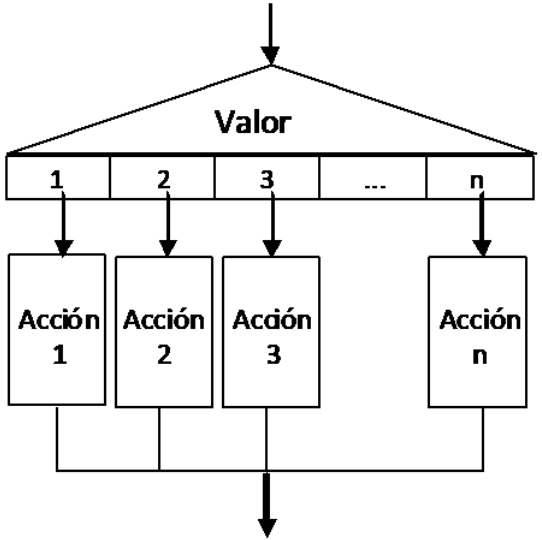



```
if (expresion logica)
{
    instrucciones por verdadero
}
```

Variantes de la expresión condicional

Argentina programa 4.0

- Múltiple



Según Valor Hacer

Valor 1: Acción 1

Valor 2: Acción 2

Valor 3: Acción 3

...

Valor n: Acción n

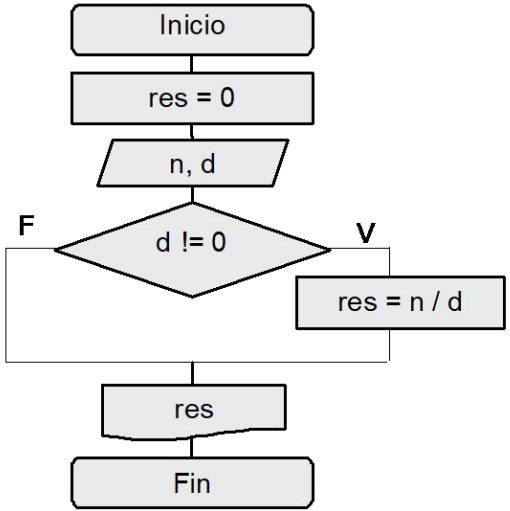
Fin Según

```
switch (expresión_entera)
{
    case (valor1) :
            instrucciones_1;
            [break;]
    case (valor2) :
            instrucciones_2;
            [break;]
    ...
    case (valorN) :
            instrucciones_N;
            [break;]
    default: instrucciones_por_defecto;
}
```

Estructura condicional

Argentina programa 4.0

- Ejemplo 1: Verificar que un divisor sea diferente de 0.

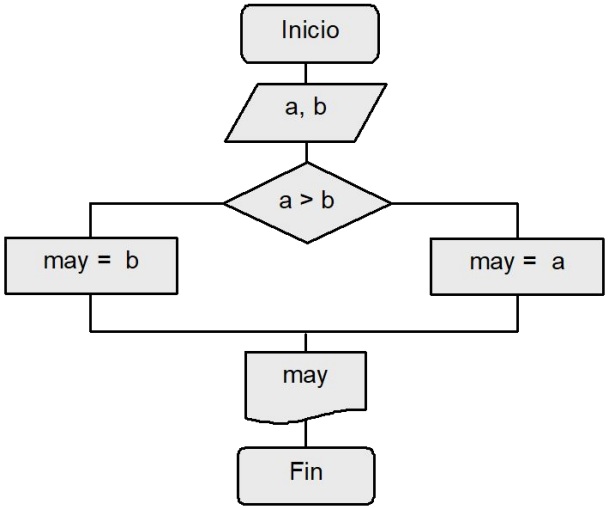



```java
package clase2;
import java.util.Scanner;

/**
 *Verificar que un divisor sea diferente de 0
 */
public class Ejercicio2_2 {

    public static void main(String[] args) {
        float res = 0;
        int num, den;
        Scanner sc = new Scanner(System.in);
        // Ingresar el numerador
        System.out.print("Ingrese el numerador: ");
        num = sc.nextInt();
        // Ingresar el denominador
        System.out.print("Ingrese el denominador: ");
        den = sc.nextInt();
        //proceso
        if (den != 0)
            res = num / den;
        //resultado
        System.out.print("El valor de la división es:"+ res);
    }
}
```

Estructura condicional

Argentina programa 4.0

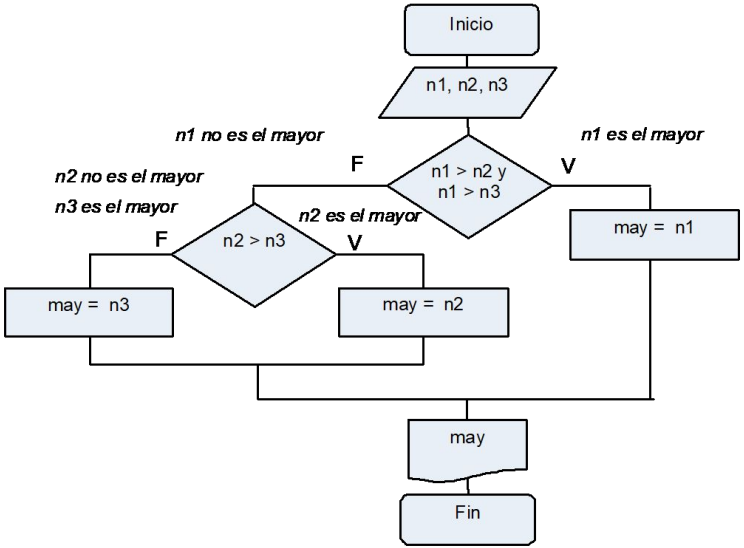
- Ejemplo 2: Cargar dos números por teclado y mostrar el mayor.




```java
package clase2;
import java.util.Scanner;

/**
 * Cargar dos números por teclado y mostrar el mayor.
 */
public class Ejercicio2_2 {

    public static void main(String[] args) {
        int mayor;
        int a, b;
        Scanner sc = new Scanner(System.in);

        System.out.print("Ingrese el primer número: ");
        a = sc.nextInt();
        System.out.print("Ingrese el segundo número: ");
        b = sc.nextInt();
        //proceso
        if (a > b)
            mayor = a;
        else
            mayor = b;
        //resultado
        System.out.print("El valor mayor es:"+ mayor);
    }
}
```

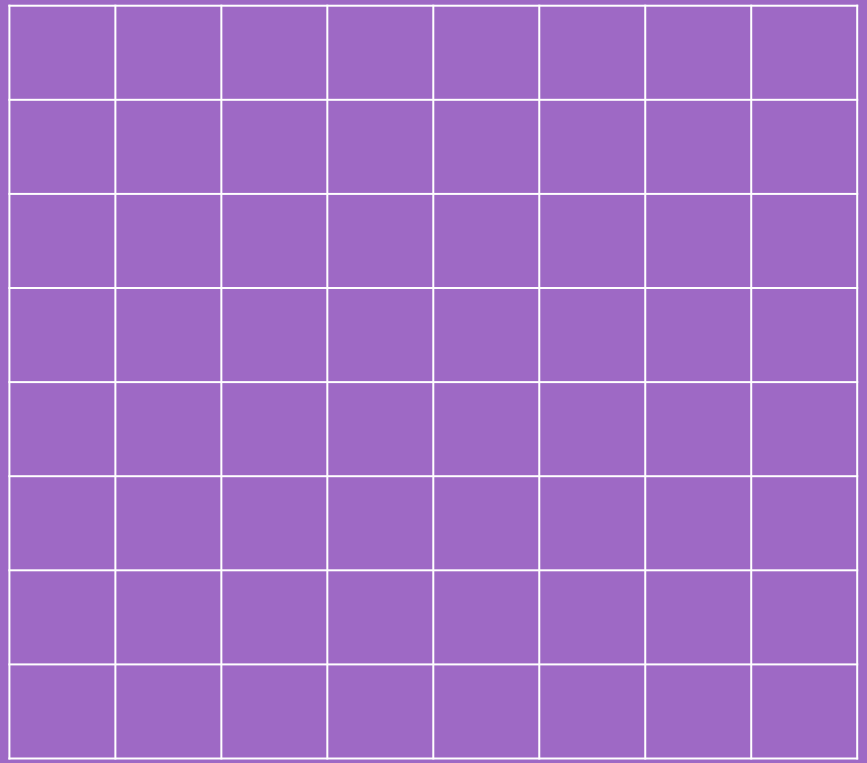
Estructura condicional

- Ejemplo 3: Cargar tres números por teclado y mostrar el mayor.




```java
package clase2;
import java.util.Scanner;

/** * Cargar tres números por teclado y mostrar el mayor. *
public class Ejercicio2_3 {

    public static void main(String[] args) {
        int mayor;
        int num1, num2, num3;
        Scanner sc = new Scanner(System.in);

        System.out.print("Ingrese el primer número: ");
        num1 = sc.nextInt();
        System.out.print("Ingrese el segundo número: ");
        num2 = sc.nextInt();
        System.out.print("Ingrese el tercer número: ");
        num3 = sc.nextInt();
        //proceso
        if (num1 > num2 && num1 > num3)
            mayor = num1;
        else
            if (num2 > num3)
                mayor = num2;
            else
                mayor = num3;
        //resultado
        System.out.print("El valor mayor es: "+ mayor);
    }
}
```

ESTRUCTURA REPETITIVA

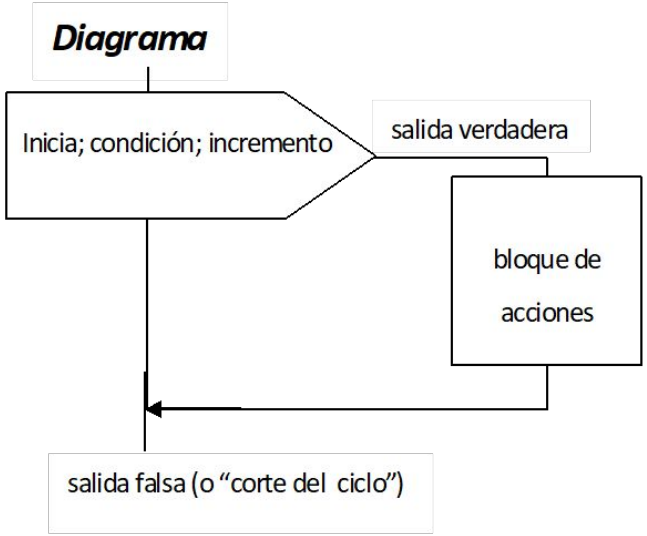


Estructura repetitiva

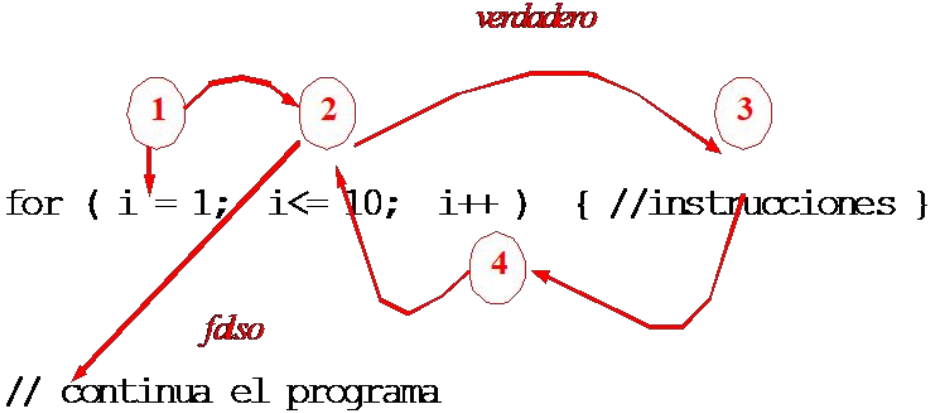
Argentina programa 4.0

- Permite repetir la ejecución de un bloque de instrucciones mientras se cumpla una condición o expresión lógica.
- Es un ciclo, bucle o instrucción compuesta que permite la repetición controlada de la ejecución de cierto conjunto de instrucciones en un programa.
  - La cabecera del ciclo, que incluye una condición de control y/o elementos adicionales en base a los que se determina si el ciclo continúa o se detiene.
  - El bloque de acciones o cuerpo del ciclo, que es el conjunto de instrucciones cuya ejecución se pide repetir.
- En Java
  - Ciclo for
  - Ciclo while
  - Ciclo do while

Ciclo for en Java




```
for(inicia;condicion;cambiaElemento){
   //Una Accion
}
```



Ciclo for en Java

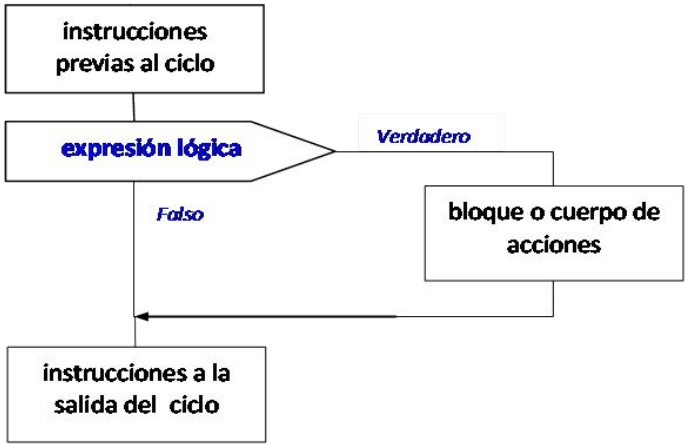
Argentina programa 4.0

4. Mostrar los números de 0 a n en orden creciente, ingresar n por teclado.

```java
package clase2;
import java.util.Scanner;

/** * Mostrar los números de 0 a n en orden creciente, ingresar n por teclado */
public class Ejercicio2_4 {

    public static void main(String[] args) {
        // para el ciclo
        int n, i;
        Scanner sc = new Scanner(System.in);

        System.out.print("Ingrese el valor de n: ");
        n = sc.nextInt();
        //proceso
        for (i = 0; i <= n; i++)
        {
            //resultados parciales
            System.out.println("Valor : "+ i);
        }
    }
}
```

```
Output - Clase2 (run)
run:
Ingrese el valor de n: 5
Valor : 0
Valor : 1
Valor : 2
Valor : 3
Valor : 4
Valor : 5
BUILD SUCCESSFUL (total time: 2 seconds)
```

Ciclo while en Java




```
while(condicion){
     //Una Accion
     //En algun momento se tiene
     // modificar la condicion
   }
```

Ciclo while en Java

Argentina programa 4.0

5. Mostrar los números de 0 a n en orden creciente, ingresar n por teclado.

```java
package clase2;
import java.util.Scanner;

/** * Mostrar los números de 0 a n en orden creciente, ingresar n por teclado */
public class Ejercicio2_5 {

    public static void main(String[] args) {
        // para el ciclo
        int n, i;
        Scanner sc = new Scanner(System.in);

        System.out.print("Ingrese el valor de n: ");
        n = sc.nextInt();
        //proceso
        // inicializar la variable de condicion del ciclo
        i = 0;
        // recorrer hasta que i valga mayor que n
        while (i <= n)
        {
            //resultados parciales
            System.out.println("Valor : "+ i);
            // incrementar el valor de i
            i++;
        }
    }
}
```

```
Output - Clase2 (run)
run:
Ingrese el valor de n: 5
Valor : 0
Valor : 1
Valor : 2
Valor : 3
Valor : 4
Valor : 5
BUILD SUCCESSFUL (total time: 2 seconds)
```

Ciclo while en Java

Argentina programa 4.0

6. Ingresar n números, calcular y mostrar su suma

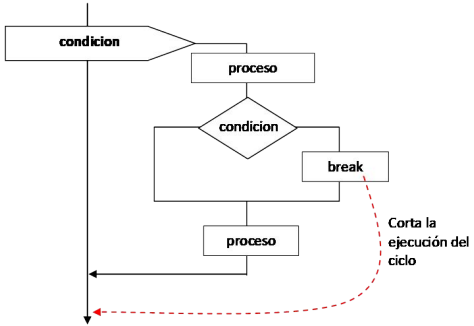
```java
package clase2;
import java.util.Scanner;

/** * Ingresar n números, calcular y mostrar su suma*/
public class Ejercicio2_6 {
    public static void main(String[] args) {
        // para el ciclo
        int n, i;
        // para los numeros
        int num = 0;
        // para la suma
        int suma = 0;
        Scanner sc = new Scanner(System.in);

        System.out.print("Ingrese el valor de n: ");
        n = sc.nextInt();
        // inicializar la variable de condicion del ciclo
        i = 0;
        while (i < n)
        {
            System.out.print("Ingrese un número: ");
            num = sc.nextInt();
            // acumula eel valor de suma
            suma += num;
            System.out.println("PRUEBA:  Valor de i: "+i+" num: "+num+ " sum: "+ suma);
            // incrementar el valor de i
            i++;
        }
        // resultado final
        System.out.println("la suma es: "+ suma);
    }
}
```

```
Output - Clase2 (run) #2
run:
Ingrese el valor de n: 3
Ingrese un número: 6
PRUEBA:  Valor de i: 0 num: 6 sum: 6
Ingrese un número: 2
PRUEBA:  Valor de i: 1 num: 2 sum: 8
Ingrese un número: 1
PRUEBA:  Valor de i: 2 num: 1 sum: 9
la suma es: 9
```

Control de ciclo en Java

Argentina programa 4.0

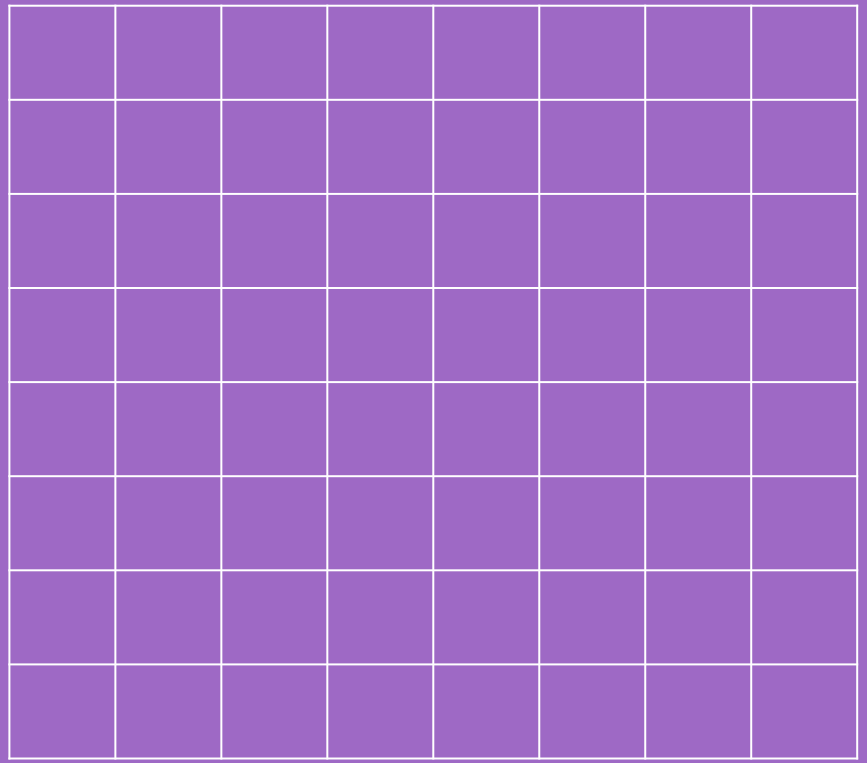
7. Ingresar n números, calcular y mostrar su suma, si se ingresa un número negativo, cortar el ciclo.




```
Output - Clase2 (run) #2
run:
Ingrese el valor de n: 3
Ingrese un número: 4
PRUEBA:  Valor de i: 0 num: 4 sum: 4
Ingrese un número: 2
PRUEBA:  Valor de i: 1 num: 2 sum: 6
Ingrese un número: -4
la suma es: 6
```

```java
import java.util.Scanner;

/** * Ingresar n números, calcular y mostrar su suma, si se ingresa un
 * número negativo, cortar el ciclo*/
public class Ejercicio2_7 {

    public static void main(String[] args) {
        int n, i, num;
        int suma = 0;
        Scanner sc = new Scanner(System.in);

        System.out.print("Ingrese el valor de n: ");
        n = sc.nextInt();
        i = 0;
        while (i < n)
        {
            System.out.print("Ingrese un número: ");
            num = sc.nextInt();
            //valida que el numero sea negativo
            if (num < 0)
                break;
            // acumula el valor de suma
            suma += num;
            System.out.println("PRUEBA:  Valor de i: "+i+" num: "+num+ " sum: "+ suma);
            // incrementar el valor de i
            i++;
        }
        // resultado final
        System.out.println("la suma es: "+ suma);
    }
}
```

PRUEBA DE EJECUCION



Prueba de ejecución

Argentina programa 4.0

La prueba de nuestro programa es una de las etapas más importante en el desarrollo del programa, ya que nos permite saber:

- Si el programa hace lo que debería hacer.
- Si no hace lo que debería hacer, nos permitirá detectar errores como ser:
- Si algún paso o instrucción no está en el orden correcto.
- Si falta algo.
- Si algo está de más.
- Si los pasos o instrucciones que se repiten lo hacen más o menos veces de lo debido.
- Si las instrucciones están en un orden apropiado.
- Otros errores que pueden presentarse.
- Elegir los datos apropiados para la prueba.

Prueba de ejecución

```
1   public class Principal {
2
3       public static void main(String args[])
4       {
5           int c, s;
6
7           c = 0;
8           s = 0;
9           while (c < 5)
10          {
11              c++;
12              System.out.println("valor parcial de c: "+c);
13              s = s + c;
14              System.out.println("valor parcial de s: "+s);
15          }
16          System.out.println("\n La sumatoria de los valores es: "+s);
17      }
18  }
```

```
run:
valor parcial de c: 1
valor parcial de s: 1
valor parcial de c: 2
valor parcial de s: 3
valor parcial de c: 3
valor parcial de s: 6
valor parcial de c: 4
valor parcial de s: 10
valor parcial de c: 5
valor parcial de s: 15

 La sumatoria de los valores es: 15
BUILD SUCCESSFUL (total time: 0 seconds)
```

Pasos	c = c + 1	s = s + 1	Observación
0	0	0	Inicialización
1	c = 0 +1 = 1	s = 0 + 1 = 1	En el ciclo
2	c = 1 + 1 = 2	s = 1 + 2 = 3	En el ciclo
3	c = 2 + 1 = 3	s = 3 + 3 = 6	En el ciclo
4	c = 3 + 1 = 4	s = 6 + 4 = 10	En el ciclo
5	c = 4 + 1 = 5	s = 10 + 5 = 15	En el ciclo
	5	15	Fuera el ciclo

Otra forma de prueba, que es bastante efectiva, es colocar mensajes auxiliares en el código, para ir controlando los resultados parciales, y así poder saber si se está ejecutando lo deseado

Prueba de ejecución

Argentina programa 4.0

Cargar una nota y mostrar si el alumno reprobó con nota menor a 6, si aprobó con nota mayor e igual a 6 o promocionó con nota mayor a 8.

```java
import java.util.Scanner;

public class Principal {

    public static void main(String args[])
    {
        int nota;
        String res = "";
        Scanner miEscaner = new Scanner(System.in);

        System.out.print("\nIngrese una nota: ");
        nota = miEscaner.nextInt();
        if (nota < 6)
            res = "REPROBADO";
        else
            if (nota > 8)
                res = "PROMOCIONADO";
            else
                res = "APROBADO";

        System.out.println("\nEl alumno está : "+res);
    }
}
```

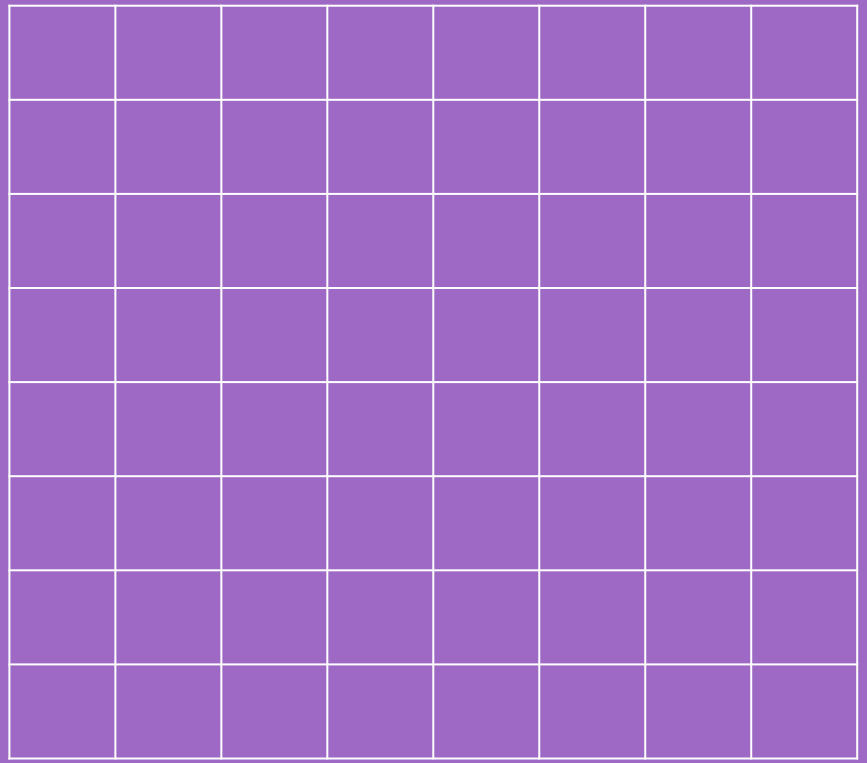
3 datos (valores) para probar (hay 3 casos posibles)

```
run:

Ingrese una nota: 4

El alumno está : REPROBADO
BUILD SUCCESSFUL (total time: 3 seconds)
```

```
run:

Ingrese una nota: 9

El alumno está : PROMOCIONADO
BUILD SUCCESSFUL (total time: 2 seconds)
```

```
run:

Ingrese una nota: 7

El alumno está : APROBADO
BUILD SUCCESSFUL (total time: 3 seconds)
```

EJERCICIOS



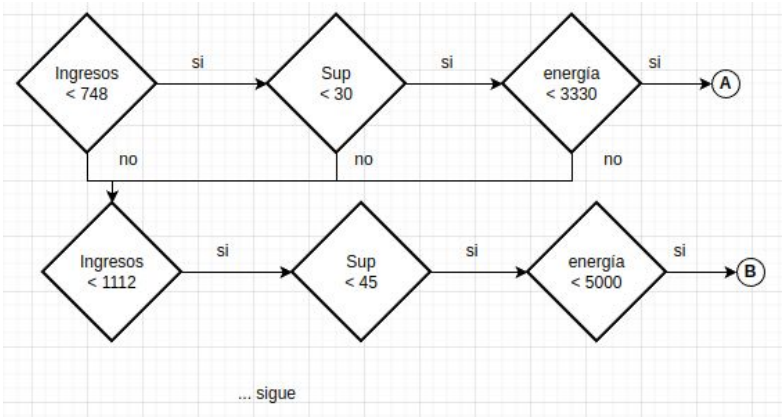
Ejercicio: Categoría Monotributo

Problema: Determinar qué categoría del monotributo corresponde una determinada persona
Entrada: Ingresos, superficie, energía eléctrica
Salida: Categoría

Categ.	Ingresos brutos	Sup. Afectada hasta:	Energía eléctrica consumida anualmente, hasta:
A	$748.382,07	30 m2	3330 Kw
B	$1.112.459,83	45 m2	5000 Kw
C	$1.557.443,75	60 m2	6700 Kw
D	$1.934.273,04	85 m2	10000 Kw
E	$2.277.684,56	110 m2	13000 Kw
F	$2.847.105,70	150 m2	16500 Kw
G	$3.416.526,83	200 m2	20000 Kw
H	$4.229.985,60	200 m2	20000 Kw

Ejemplo	Ingresos	Sup.	Energía	Cat?
Docente X	$500.000,00	0	330 Kw	?
Carpintero X	$1000000	30 m2	10000 Kw	?
Vendedor X	$1.112.460	0	0	?

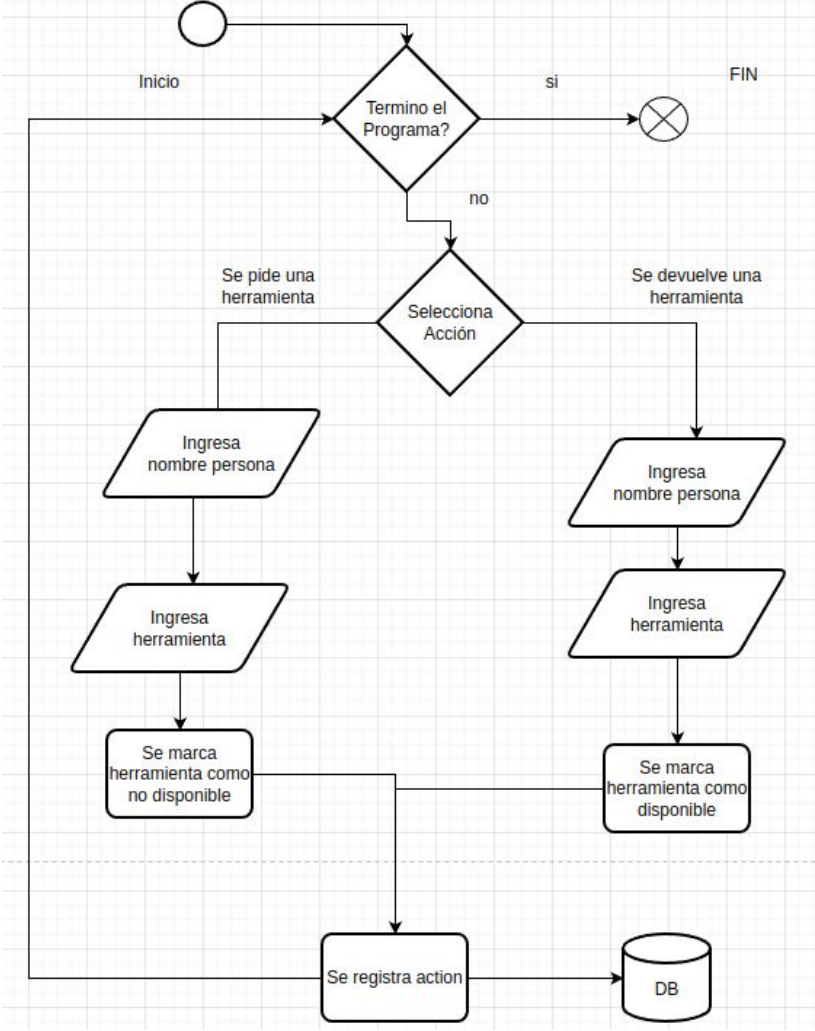
Determine a qué categoría pertenece cada ejemplo.
¿Se anima a escribir el procedimiento?



Ejercicio: Herramientas de un taller

Problema: Llevar contabilidad de quien usa las herramientas de un taller

- Préstamo de herramienta
  - Entrada: Una persona se lleva una herramienta
  - Salida: eso queda registrado y la herramienta no está disponible para el siguiente
- Devolución:
  - Entrada: Una devuelve una herramienta que que pidió prestada
  - Salida: La acción queda registrada y la herramienta está disponible para el siguiente que venga



Argentina
programa
4.0

Gracias!