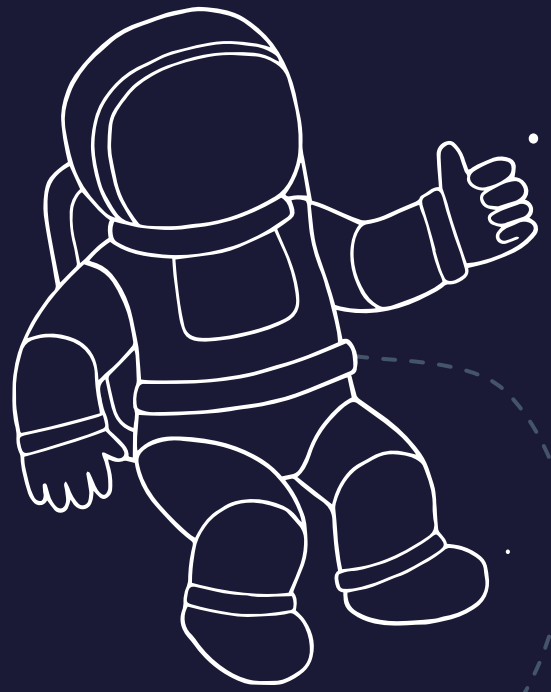
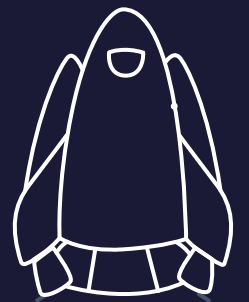


Programa académico CAMPUS

MODULO JAVA
Sesión 3: Operadores y estructuras de decisión

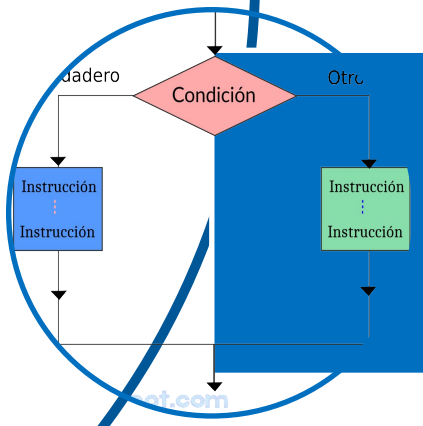


Trainer Carlos H. Rueda C.


Java™

TEMARIO

For



While

ESTRUCTURAS REPETITIVAS

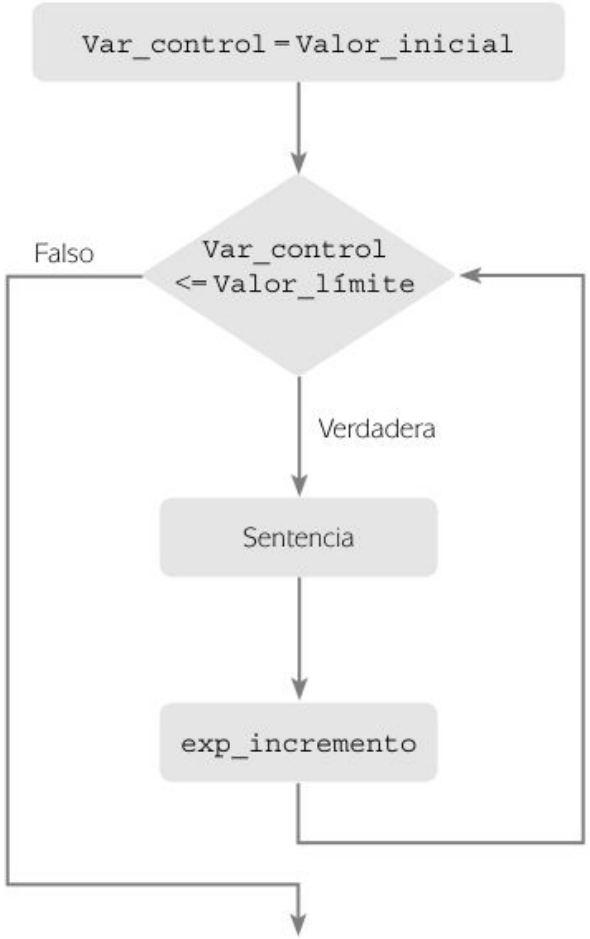
Las estructuras que repiten una secuencia de instrucciones un número determinado de veces se denominan bucles y se denomina iteración al hecho de repetir la ejecución de una secuencia de acciones.

CICLOS FOR

Java también proporciona la declaración de repetición **for**, que especifica los detalles de un ciclo controlado por el contador en una sola línea de código.

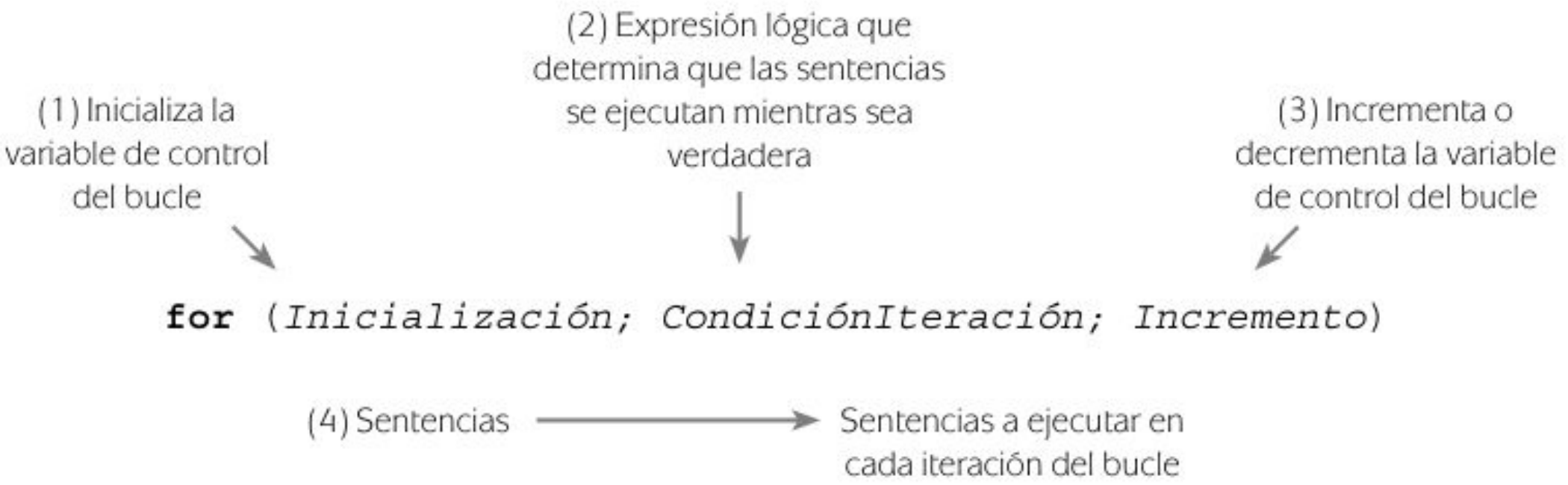
CICLOS FOR

Diagrama de flujo:



CICLOS FOR

Sintaxis



CICLOS FOR

Calcular 15 veces el valor de la función ex – x y escribir los resultados.

Ejemplo

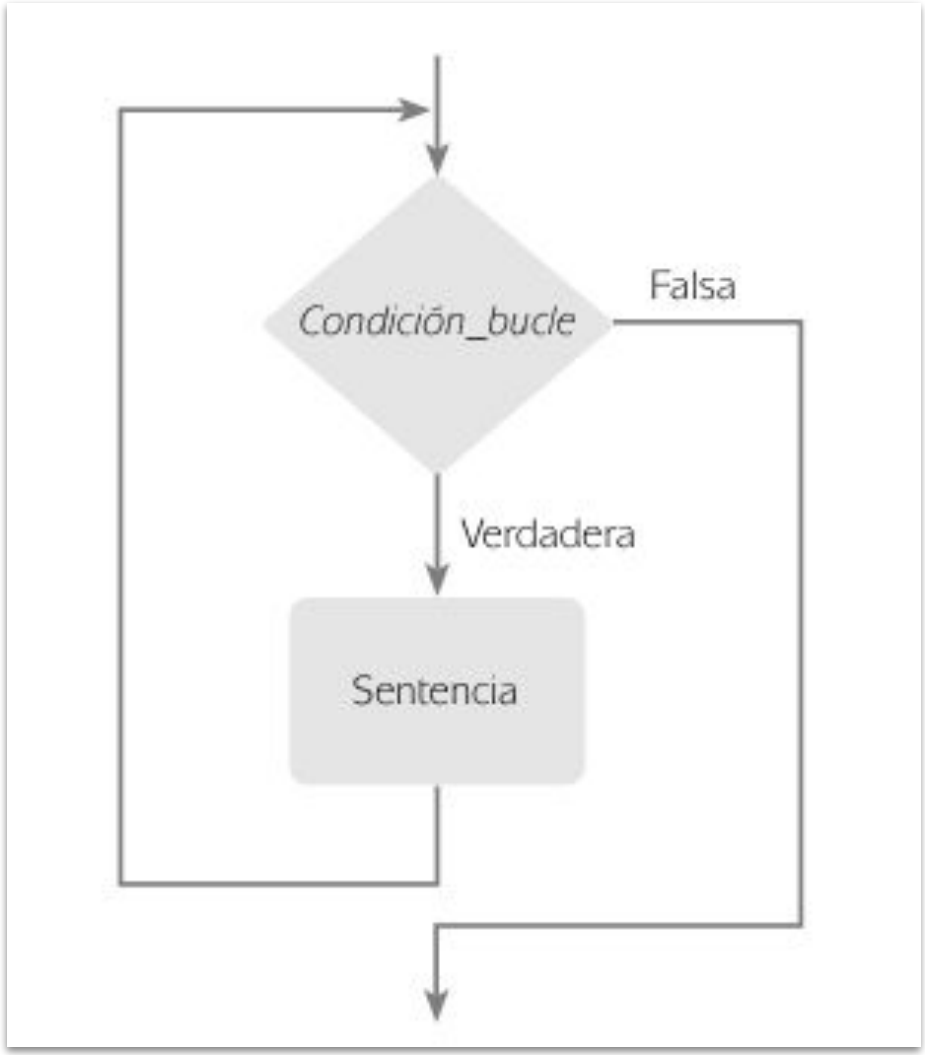
```
public static void main(String []a)
{
   final int VECES = 15;
   Scanner entrada = new Scanner(System.in);
   for (int i = 1; i <= VECES; i++)
   {
      double x, f;
      System.out.print("Valor de x: ");
      x = entrada.nextDouble();
      f = Math.exp(2*x) - x;
      System.out.println("f(" + x + ") = " + f);
   }
}
```

CICLOS WHILE

Un bucle **while** tiene una condición, una expresión lógica que controla la secuencia de repetición; su posición es delante del cuerpo del bucle y significa que while es un bucle *pretest*, de modo que cuando éste se ejecuta, se evalúa la condición antes de ejecutarse el cuerpo del bucle

CICLOS WHILE

Diagrama de flujo



CICLOS WHILE

Sintaxis

1. **while** (*condición_bucle*)
   sentencia; → Cuerpo

2. **while** (*condición_bucle*)
   {
   sentencia-1;
   sentencia-2;
   .
   .
   .
   sentencia-n;
   } → Cuerpo

while	es una palabra reservada de Java
condición_bucle	es una expresión lógica
sentencia	es una sentencia simple o compuesta

CICLOS WHILE

Ejemplo

```
// cuenta hasta 10
int x = 0;
while (x < 10)
  System.out.println("X: " + x++);
```

CICLOS WHILE

Ejemplo

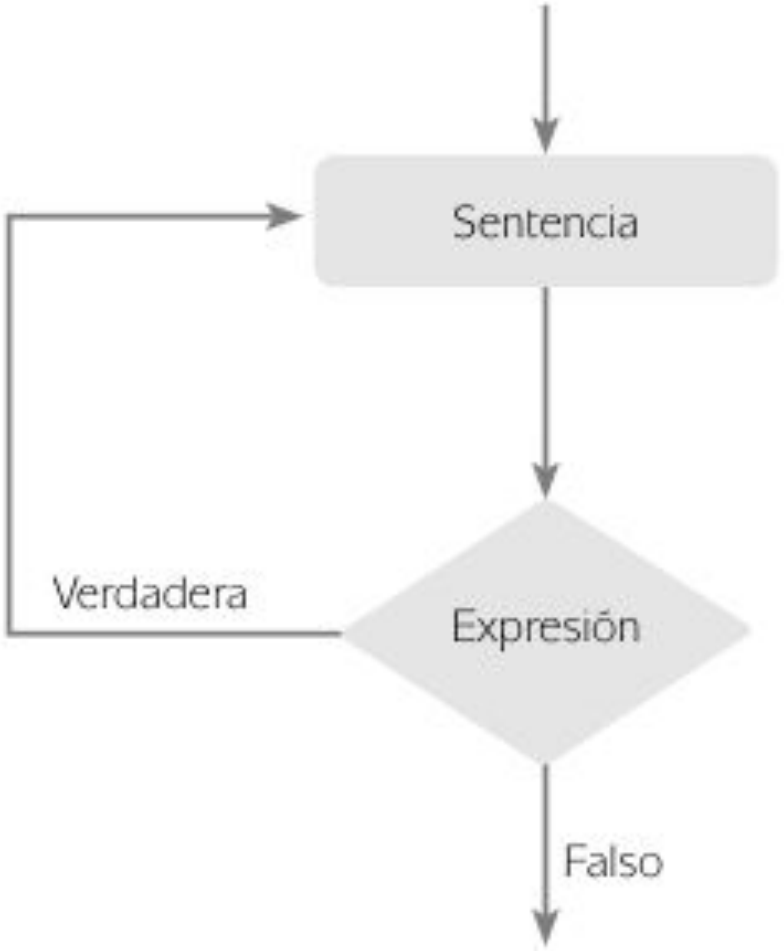
```
// visualizar n asteriscos
contador = 0; ──────────────────────► inicialización
while (contador < n) ──────────────► prueba/condición
{
   System.out.print(" * ");
   contador++; ─────────────────────► actualización (incrementa en 1 contador)
} // fin de while
```

CICLOS DO - WHILE

La sentencia **do-while** se utiliza para especificar un bucle condicional que se ejecuta al menos una vez; cuando se desea realizar una acción determinada al menos una o varias veces, se recomienda este bucle.

CICLOS DO - WHILE

Diagrama de flujo



CICLOS DO - WHILE

Sintaxis

Acción (sentencia) a ejecutar al menos una vez

Expresión lógica que determina si la acción se repite

1. **do** *sentencia* **while** (*expresión*)

2.
```
do
   sentencia
while (expresión)
```

CICLOS DO - WHILE

Ejemplo

```
public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    char digito;
    do {
        System.out.println("Introduzca un digito (0-9): ");
        digito = scan.nextLine().charAt(0);
    } while ((digito < '0') || (digito > '9'));
}
```

CICLOS DO - WHILE

Ejemplo

Escribir un programa que visualice el factorial de un entero comprendido entre 2 y 20 usando un ciclo ***do - while***

```
public static void main(String[] args) {
    long n, m, fact;
    Scanner entrada = new Scanner(System.in);
    do {
        System.out.println("\nFactorial de número n, entre 2 y 20: ");
        n = entrada.nextLong();
    } while ((n < 2) || (n > 20));

    for (m = n, fact = 1; n > 1; fact *= n--);

    System.out.println(m + "! = " + fact);
}
```

Transferencia de control: sentencias break y continue

Sentencia break

La sentencia break, ya estudiada antes, normalmente realiza dos acciones:

- La salida inmediata de un bucle.
- Saltar el resto de la sentencia *switch*.

el flujo de control del programa continúa en la siguiente sentencia después del bucle

Transferencia de control: sentencias break y continue

Ejemplo

```
for (int i = 1; i <= 25; i++)
{
   d = leerDistancia(i)
   if (d == 0) // salida de bucle
      break;
   System.out.println ("Distancia: ", d);
}
```

Transferencia de control: sentencias break y continue

Sentencia continue

La sentencia continue se utiliza en los tres tipos de bucles; cuando se ejecuta en un bucle, se saltan las sentencias restantes y se prosigue con la siguiente iteración.

Transferencia de control: sentencias break y continue

Ejercicio

Dado un entero `n`, devuelve *una sucesión de cadenas separadas por coma* `answer` *(indexada desde 1) donde*:

- `i == "FizzBuzz"` si `i` es divisible por `3` y `5`.
- `i == "Fizz"` si `i` es divisible por `3`.
- `i == "Buzz"` si `i` es divisible por `5`.
- `i == i` (como cadena) si ninguna de las condiciones anteriores es verdadera.

Restricciones:

- `1 <= n <= 104`

Example 1:

Input: n = 3
Output: “1”, “2”, “Fizz”

Example 2:

Input: n = 5
Output: “1”, “2”, “Fizz”, “4”,“Buzz”

Example 3:

Input: n = 15
Output:
“1”, “2”, “Fizz”, “4”, “Buzz”, “Fizz”, “7”, “8”, “Fizz”, “Buzz”, “11”, “Fizz”, “13”, “14”, “FizzBuzz”

Transferencia de control: sentencias break y continue

Ejercicio

Realice el programa que responda al siguiente requerimiento.

```
Voy a contar las calificaciones aprobadas.
Teclea las calificaciones (termina con -1)
7
4
8
10
-1
Obtuviste 3 calificación(es) aprobatoria(s) y 1 calificación(es)
reprobatoria(s).
Tu promedio fue: 7.25
```

Transferencia de control: sentencias break y continue

Ejercicio

Un apostador asiduo tiene la costumbre de apostar si atina al número que resulta al tirar un dado. Si lo logra gana $5.00; en caso contrario, pierde $1.00. Comienza con $7.00 y el juego termina cuando gana $6.00 o pierde todo su capital.

```
¿Qué número saldrá en el dado? 5
El dado arrojó un 4. Acaba de perder $1.00
Su saldo es $6.00

¿Qué número saldrá en el dado? 5
```

Transferencia de control: sentencias break y continue

Ejercicio

Juguemos a menor-mayor. Usted tiene que apostar un peso a alguna de las siguientes opciones:

Menor	7	Mayor
2-6	7	8-12

Después tira dos dados. Si le apostó a menor y la suma de los dados está en el rango de 2 a 6 gana un peso. Algo similar sucede con el caso de mayor. Si le apostó a 7 y la suma de los dados coincide con 7, entonces gana 5 pesos. Arranca con 5 pesos y termina su participación cuando tiene 10 pesos o pierde todo su capital.

Construya un programa en Java que realice este juego.

Transferencia de control: sentencias break y continue

Ejercicio

En una empresa de computadoras, los salarios de los empleados se aumentarán según su contrato actual:

Contrato	Aumento %
0 a 9 000 dólares	20
9 001 a 15 000 dólares	10
15 001 a 20 000 dólares	5
más de 20 000 dólares	0

Escribir un programa que solicite el salario actual de cada empleado y que, además, calcule y visualice el nuevo salario.

Transferencia de control: sentencias break y continue

Ejercicio

La constante pi (3.141592) se utiliza en matemáticas; un método sencillo de calcular su valor es:

$$Pi = 2 * \frac{2}{1} * \frac{2}{3} * \frac{4}{3} * \frac{4}{5} * \frac{6}{5} * \frac{6}{7} * \frac{8}{7} * \frac{8}{9} \qquad \frac{2n}{2n-1} \quad \frac{2n}{2n-1}$$

Escribir un programa que efectúe este cálculo con un número de términos especificados por el usuario.

Transferencia de control: sentencias break y continue

Ejercicio

Escribir un programa que encuentre los tres primeros números perfectos pares y los tres primeros números perfectos impares.

Un número perfecto es un entero positivo que es igual a la suma de todos los enteros positivos (excluido él mismo) que son sus divisores. El primer número perfecto es 6, ya que sus divisores son 1, 2, 3 y 1 + 2 + 3 = 6.

Transferencia de control: sentencias break y continue

Ejercicio

Escribir un programa que encuentre los tres primeros números perfectos pares y los tres primeros números perfectos impares.

Un número perfecto es un entero positivo que es igual a la suma de todos los enteros positivos (excluido él mismo) que son sus divisores. El primer número perfecto es 6, ya que sus divisores son 1, 2, 3 y 1 + 2 + 3 = 6.

Transferencia de control: sentencias break y continue

Ejercicio

Calcular todos los números de tres cifras tales que la suma de los cubos de las cifras es igual al valor del número.

Transferencia de control: sentencias break y continue

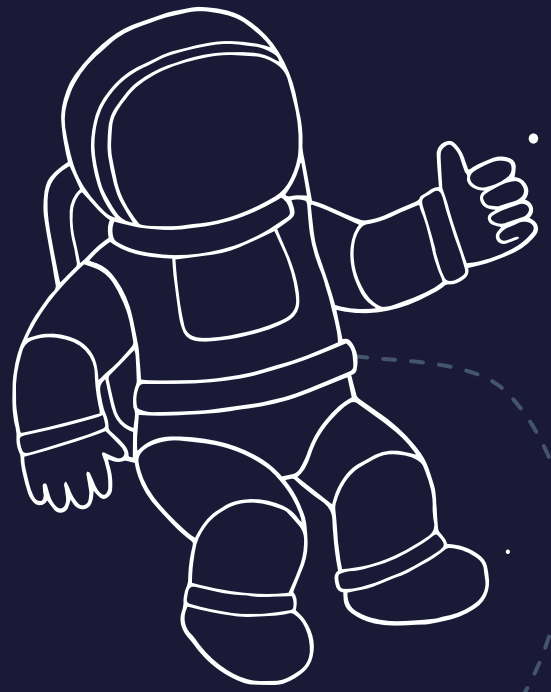
Ejercicio

Escribir un programa que encuentre el primer número primo introducido por medio del teclado.

Transferencia de control: sentencias break y continue

Ejercicio

Una estación climática proporciona un par de temperaturas diarias (máxima, mínima) (no es posible que alguna o ambas temperaturas sea 9 grados). La pareja fin de temperaturas es 0,0. Se pide determinar el número de días, cuyas temperaturas se han proporcionado, las medias máxima y mínima, el número de errores —temperaturas de 9°— y el porcentaje que representaban.



Programa académico CAMPUS

Ciclo 2