

Desarrollo de Algoritmos

Estructuras de Control Repetitivas

Facultad de Informática
Universidad Nacional del Comahue

2021

Temario

- 1 Introducción
- 2 Estructura Repetitiva REPETIR HASTA
- 3 Estructura Repetitiva MIENTRAS
- 4 Estructura Repetitiva PARA

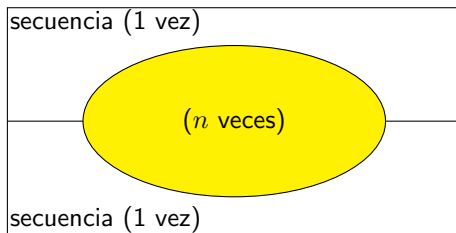
Temario

- 1 Introducción
- 2 Estructura Repetitiva REPETIR HASTA
- 3 Estructura Repetitiva MIENTRAS
- 4 Estructura Repetitiva PARA

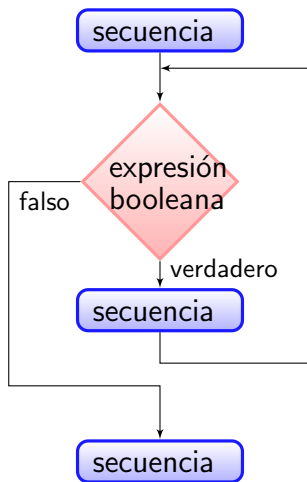
Repetitivas

Definición: CONJUNTO DE DATOS DE ESTRUCTURA REPETITIVA

Una **estructura repetitiva** es aquella en la que un conjunto de acciones se ejecutan un número determinado de veces y dependen de un **valor predefinido** o el **cumplimiento de una determinada condición**. Cada vez que se ejecuta el conjunto de acciones a repetir, diremos que se ejecuta una **iteración** del bucle.



Estructura Repetitiva MIENTRAS



Estructura Repetitiva MIENTRAS

ALGORITMO....

secuencia

MIENTRAS (expresión booleana) **HACER**

cuerpo de la

estr. rep. Se ejecuta

si exp.bool es true

FIN MIENTRAS

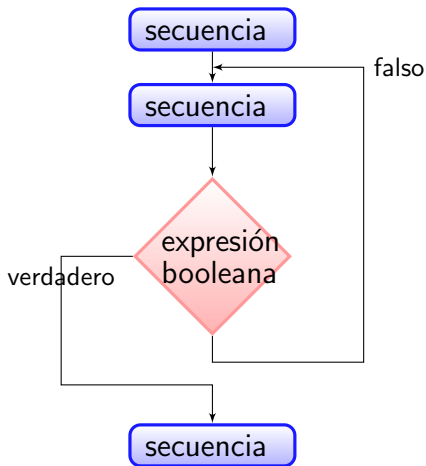
secuencia

FIN ALGORITMO....

JAVA: while

```
1      while (condición)
2          {
3              //acciones del bucle
4          }
```

Estructura Repetitiva REPETIR HASTA



Estructura Repetitiva REPETIR HASTA

ALGORITMO....

secuencia

REPETIR

Cuerpo de la
estr. rep. Se ejecuta
si exp. booleana es false

HASTA (expresion booleana)

secuencia

FIN ALGORITMO....

JAVA: do while

```
1      do
2          {
3              //acciones del bucle
4          } while (condición)
```


Estructura Repetitiva PARA

ALGORITMO....

secuencia

PARA varIter ← limInf **HASTA** limSup **PASO** varInc **HACER**

Cuerpo de la
Estructura
Repetitiva

FIN PARA

secuencia

FIN ALGORITMO....

JAVA: for

```
1      for (i=0; i<n; i++)  
2          {  
3              //acciones del bucle  
4          }
```

Definición: CICLO DEFINIDO

Un ciclo se dice definido cuando conocemos la cantidad de veces a iterar. Es un valor constante o una variable entera (por ej. cantAlumnos).

Definición: CICLO INDEFINIDO

Un ciclo se dice indefinido cuando no conocemos la cantidad de veces a iterar. El usuario indica con un valor determinado cuando desea parar la iteración. Ej. leer valores hasta que el usuario ingrese el valor -1, o un punto ('.'), etc.

Repetitivas

Variables Iteradoras y Acumuladoras

Definición: VARIABLES ITERADORA

Una variable iteradora está limitada al contexto de un bucle, ya su valor se incrementa en cada iteración por una cantidad fija.

Definición: VARIABLE ACUMULADORA

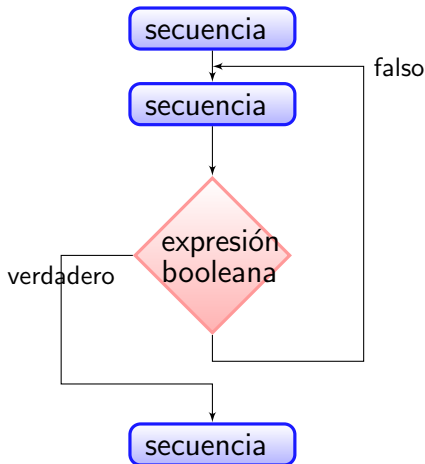
Una variable acumuladora es aquella que verifica que en determinados puntos del flujo del programa su valor se actualiza para reflejar el estado actual de una serie de operaciones parciales.

Temario

- 1 Introducción
- 2 Estructura Repetitiva REPETIR HASTA
- 3 Estructura Repetitiva MIENTRAS
- 4 Estructura Repetitiva PARA

Estructura Repetitiva REPETIR HASTA

Repite una o más veces. Repite mientras la expresión booleana sea falsa.



Estructura Repetitiva REPETIR HASTA

- 1 Ejecuta el bloque de sentencias a repetir.
- 2 Evalúa la expresión booleana.
- 3 Si la expresión booleana es falsa vuelve al paso 1.
- 4 Si la expresión booleana es verdadera termina la estructura.

ALGORITMO....

secuencia

REPETIR

Cuerpo de la
estr. rep. Se ejecuta
si exp. booleana es false

HASTA (expresion booleana)

secuencia

FIN ALGORITMO....

Estructura Repetitiva REPETIR HASTA

La estructura java no es exáctamente igual al REPETIR HASTA de pseudocódigo.

- 1 Ejecuta el bloque de sentencias a repetir.
- 2 Evalúa la expresión booleana.
- 3 Si la expresión booleana es verdadera vuelve al paso 1.
- 4 Si la expresión booleana es falsa termina la estructura.

JAVA: do while

```
1      do
2      {
3          //acciones del bucle
4      } while (condición)
```

Repetir Hasta. Ejemplo: Relación Maestro-Detalle.

Una relación maestro detalle es una relación de tipo uno a muchos.

Ejemplos de una relación maestro-detalle son: un conjunto de productos de una factura o tickets de compra, un informe de gastos con un conjunto de líneas de gastos, un departamento con una lista de empleados que pertenecen a él.



Pseudocódigo

```
1  ALGORITMO supermercado() RETORNA vacío
2      ENTERO cantidad
3      REAL precioUnitario , acumulador
4      LOGICO continuar
5      acumulador ← 0.0
6      continuar ← VERDADERO
7      REPETIR
8          ESCRIBIR(" Ingrese_Cantidad")
9          LEER(cantidad)
10         ESCRIBIR(" Ingrese_Producto_Unitario")
11         LEER(precioUnitario)
12         ESCRIBIR(" Total_Producto:_" , cantidad* precioUnitario)
13         acumulador ← acumulador + cantidad* precioUnitario;
14         ESCRIBIR(" Desea_Ingresar_otro_producto?")
15         LEER(continuar)
16     HASTA (continuar=FALSO)
17     ESCRIBIR(" Total_Ticket:_" , acumulador)
18  FIN ALGORITMO supermercado
```



```
1 public static void main(String[] args) {
2     // ...
3     int i, cantidad;
4     double precioUnitario, acumulador=0.0;
5     boolean continuar= true;
6     System.out.println(" Supermercado ");
7     System.out.println(" ");
8     do {
9         System.out.print(" Ingrese_Cantidad: ");
10        cantidad = TecladoLn.readLineInt();
11        System.out.print(" Ingrese_Producto_Unitario: ");
12        precioUnitario = TecladoLn.readLineDouble();
13        System.out.println(" Total_Producto: " +
14            cantidad* precioUnitario);
15        acumulador = acumulador + cantidad* precioUnitario;
16        System.out.println(" Desea_ingresar_otro_producto? ");
17        continuar = TecladoLn.readLineBoolean();
18    } while (continuar);
19    System.out.println(" _____ ");
20    System.out.println(" Total_Ticket: "+acumulador);
21 }
```

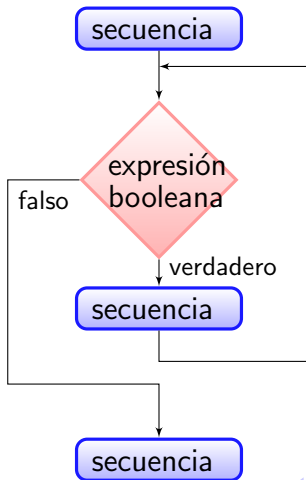


Temario

- 1 Introducción
- 2 Estructura Repetitiva REPETIR HASTA
- 3 Estructura Repetitiva MIENTRAS**
- 4 Estructura Repetitiva PARA

Estructura Repetitiva MIENTRAS

Repite cero o más veces. Repite siempre que la expresión booleana sea verdadera. Cuando es falsa termina de iterar. Si la primera evaluación es falsa no ejecuta nunca el bloque de sentencias.



Estructura Repetitiva MIENTRAS

- 1 Evalúa la expresión booleana.
- 2 Si la expresión booleana es verdadera ejecuta el bloque de sentencias de la estructura. Luego, volver a paso 1.
- 3 Si la expresión booleana es falsa termina la estructura repetitiva.

ALGORITMO....

secuencia

MIENTRAS (expresión booleana) **HACER**

cuerpo de la

estr. rep. Se ejecuta

si exp.bool es true

FIN MIENTRAS

secuencia

FIN ALGORITMO....

Estructura Repetitiva MIENTRAS

- 1 Evalúa la expresión booleana.
- 2 Si la expresión booleana es verdadera ejecuta el bloque de sentencias de la estructura. Luego, volver a paso 1.
- 3 Si la expresión booleana es falsa termina la estructura repetitiva.

JAVA: while

```
1      while (condición)
2          {
3              //acciones del bucle
4          }
```

Estructura Repetitiva MIENTRAS

Ejemplo descomponer un entero en sus cifras

Pseudocódigo

```
1  ALGORITMO principal () RETORNA Ø
2      ENTERO numero
3      ESCRIBIR(" Ingrese un entero mayor a cero")
4      LEER(numero)
5      descomponer(numero)
6  FIN ALGORITMO principal
7  MODULO descomponer(ENTERO num) RETORNA Ø
8      MIENTRAS (num>0 ) HACER
9          ESCRIBIR(num MOD 10)
10         num ← num /10
11     FIN MIENTRAS
12 FIN MODULO descomponer
```

Estructura Repetitiva MIENTRAS

Ejemplo descomponer un entero en sus cifras



```
1 public class Binario {
2     public static void main(String[] args) {
3         int numero;
4         System.out.println(" Ingrese un numero");
5         numero = TecladoLn.readLineInt();
6         descomponer(numero);
7     }
8     public static void descomponer(int num){
9         while (num > 0){
10             System.out.println(num%10);
11             num = num /10;
12         }
13     }
14 }
```


Estructura Repetitiva MIENTRAS

Ejemplo contar los dígitos que componen a un entero mayor a cero

Pseudocódigo

```
1  MODULO contar(ENTERO num) RETORNA ENTERO
2      ENTERO contador
3      contador  $\leftarrow$  0
4      MIENTRAS (num > 0) HACER
5          contador  $\leftarrow$  contador + 1
6          num  $\leftarrow$  num / 10
7      FIN MIENTRAS
8      RETORNA contador
9  FIN MODULO contador
```

Estructura Repetitiva MIENTRAS

Ejemplo contar los dígitos que componen a un entero mayor a cero

JAVA

```
1 public static int contar(int num){
2     int contador= 0;
3     while (num > 0){
4         contador++;
5         num = num /10;
6     }
7     return contador;
8 }
```

Estructura Repetitiva MIENTRAS

Ejemplo sumar los dígitos que componen a un entero mayor a cero

Pseudocódigo

```
1  MODULO sumarDigitos(ENTERO num) RETORNA ENTERO
2      ENTERO acumulador
3      acumulador  $\leftarrow$  0
4      MIENTRAS (num>0 ) HACER
5          acumulador  $\leftarrow$  acumulador + (num MOD 10)
6          num  $\leftarrow$  num / 10
7      FIN MIENTRAS
8      RETORNAR acumulador
9  FIN MODULO sumarDigitos
```

Estructura Repetitiva MIENTRAS

Ejemplo sumar los dígitos que componen a un entero mayor a cero



```
1 public static int sumarDigitos(int num){
2     int acumulador= 0;
3     while (num > 0){
4         acumulador = acumulador + (num % 10);
5         num = num /10;
6     }
7     return acumulador;
8 }
```

Estructura Repetitiva MIENTRAS

Ejemplo verificar si los dígitos que componen a un entero mayor a cero son 0 ó 1

Pseudocódigo

```
1  MODULO verificarBinario(ENTERO num) RETORNA LOGICO
2      LOGICO resultado
3      resultado ← true
4      MIENTRAS ( resultado AND num > 0 ) HACER
5          SI ((num MOD 10) <> 0 AND (num MOD 10) <> 1) ENTONCES
6              resultado ← false
7          FIN SI
8          num ← num / 10
9      FIN MIENTRAS
10     RETORNAR resultado
11 FIN MODULO verificarBinario
```

Estructura Repetitiva MIENTRAS

Ejemplo verificar si los dígitos que componen a un entero mayor a cero son 0 ó 1



```
1      public static boolean verificarBinario(int num) {  
2          boolean resultado = true;  
3          while (resultado && (num > 0)) {  
4              if (((num % 10) != 0) && ((num % 10) != 1)) {  
5                  resultado = false;  
6              }  
7              num = num / 10;  
8          }  
9          return resultado;  
10     }
```

Temario

- 1 Introducción
- 2 Estructura Repetitiva REPETIR HASTA
- 3 Estructura Repetitiva MIENTRAS
- 4 Estructura Repetitiva PARA**

```
for (inicio; condicion; incremento)
{ //acciones del bucle }
```

JAVA: Cuando utilizarlo y Procedimiento de utilizacion

Debemos utilizar un bucle PARA, cuando estamos frente al tratamiento de **ciclos definidos**.

- 1 Primero se inicializa la variable iteradora
- 2 Se verifica que la condición ó, en el caso de pseudocódigo, que el valor inicial no supere el límite final en el caso de pseudocódigo.
- 3 Se ejecuta el bloque de instrucciones a repetir.
- 4 Se incrementa la variable iteradora, de acuerdo a lo que indica PASO en pseudocódigo ó en el caso de java lo que indique la tercer componente del for.
- 5 Volver al paso 2.

Repetitivas

Ejemplo Convertir un numero de binario a decimal

Pseudocódigo

```
1  MODULO conversionDecimal (ENTERO num) RETORNA REAL
2      ENTERO i, cantidad
3      REAL acum
4      acum  $\leftarrow$  0.0
5      cantidad  $\leftarrow$  contar(num)
6      PARA i  $\leftarrow$  0 HASTA cantidad-1 PASO 1 HACER
7          acum  $\leftarrow$  acum + potencia (2 * (num MOD 10), i)
8          num  $\leftarrow$  num /10
9      FIN MIENTRAS
10     RETORNAR acum
11  FIN MODULO conversionDecimal
```

Estructura Repetitiva Para

Ejemplo Convertir un numero de binario a decimal



```
1      public static double conversionDecimal(int num) {
2          int i, cantidad;
3          double acum;
4          acum = 0.0;
5          cantidad = contar(num);
6          for (i= 1; i <= cantidad; i++) {
7              acum = acum + Math.pow(2 * (num % 10), i-1);
8              num = num / 10;
9          }
10         return acum;
11     }
```



Cátedra de Desarrollo de Algoritmos

Apunte sobre Modularización

Facultad de Informática, Universidad del Comahue, 2021.



Cátedra de Desarrollo de Algoritmos

Apunte sobre Estructura de Control Repetitiva.

Facultad de Informática, Universidad del Comahue, 2021.