



UNIVERSIDAD  
DE LOS ANDES  
MERIDA VENEZUELA

# Estructuras de repetición

## *“Repita mientras”*

Prof. Judith Barrios Albornoz

Departamento de Computación

Escuela de Ingeniería de Sistemas

Facultad de Ingeniería

Universidad de Los Andes

Semestre A\_2013

# Programación estructurada

Enfoque disciplinado que permite escribir programas estructurados, utilizando las estructuras de control:

- Secuencial (asignación, lectura, escritura)
- Decisión o selección (simple, doble, múltiple)
- **Repetición** (*Repita mientras,* ←

*Hacer-Mientras, Repita hasta,  
Repita para*)

# Estructuras de Repetición

- Estructura de **repetición** o lazo

Permite que un conjunto de sentencias (**una o varias**) de un programa sea ejecutado por la computadora en forma repetida

- **Tipos de estructuras de repetición**

- Repetición **condicional** o controlada por un **centinela**:
  - **Repita mientras**, *Repita hasta* y *Hacer-Mientras*
- Repetición **no condicional** o controlada por un **contador**:
  - *Repita para*

- Conceptos Básicos

- **Contador**
- **Acumulador**
- **Centinela**



# Contador

- Variable cuyo **valor** se **incrementa o decrementa** en una cantidad constante cada vez que se produce un determinado suceso o acción
- Se debe realizar primero una operación de **iniciación** y posteriormente, se realizan los correspondientes **incrementos** o **decrementos**



# Contador

- Operaciones sobre un **Contador**
  - **Iniciación**: Nombre del contador = valor inicial
  - **Incremento/decremento**:

*Nombre del contador = Nombre del contador + valor constante*

*Nombre del contador = Nombre del contador - valor constante*

- **Ejemplo**

***Iniciación***             $\text{conta1} = 0$

***Incremento***

$\text{conta1} = \text{conta1} + 1$

***Decremento***

$\text{conta1} = \text{conta1} - 2$

# Acumulador

- Variable cuyo **valor se incrementa o decrementa** en una **cantidad variable** cada vez que se produce un determinado suceso o acción
- Se debe realizar primero una **operación de iniciación** y posteriormente, se realizan los correspondientes **incrementos o decrementos**

# Operaciones sobre el acumulador

- Iniciación-> Nombre del acumulador = valor **inicial**
- Acumulación:

NombreDelAcumulador = NombreDelAcumulador + valor

NombreDelAcumulador = NombreDelAcumulador \* valor

NombreDelAcumulador = NombreDelAcumulador - valor

NombreDelAcumulador = NombreDelAcumulador / valor

## Ejemplo

Iniciación	Acumulación ( <i>incrementos</i> )	Acumulación ( <i>decrementos</i> )
totalNota1 = 0 totalNota2 = 1	totalNota1 = totalNota1 + nota totalNota2 = totalNota2 * nota	totalNota1 = totalNota1 - nota totalNota2 = totalNota2 / nota



# Centinela

- **Valor** dado a una variable o conjunto de variables que permite detectar **cuándo** se desea **terminar la repetición de** las acciones que constituyen el cuerpo de una estructura de repetición o lazo

- Ejemplo

**Iniciar**

bandera = cierto

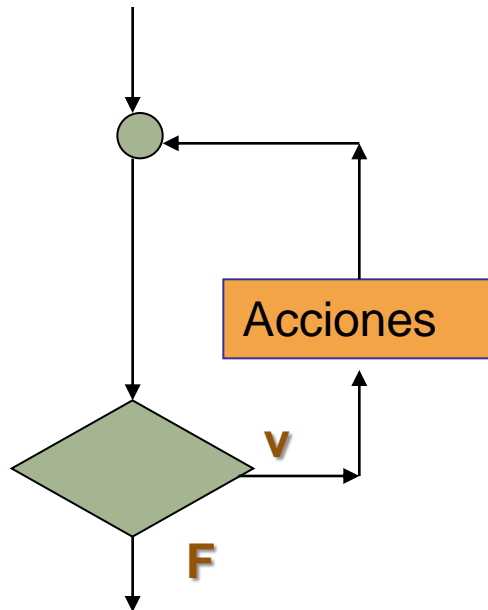
**Cuando queremos que el lazo termine su ejecución**

bandera = falso



# Repita mientras

## Diagrama de flujo



## Pseudocódigo en español

**Repita mientras** (<condición>)

$S_1$

....

$S_n$

**fin\_RM**

## Código en C

**while** (<condición>)

{

$S_1$

....

$S_n$

}



# Repita mientras

- Las sentencias (una o más) del cuerpo del lazo se **ejecutan mientras la condición** (expresión lógica) es **cierta**
  - Cuando la condición es falsa, termina la ejecución del lazo
- Se pregunta al principio por la condición,
  - **si la primera vez la condición es falsa** el lazo se ejecutará **cero veces** – no se ejecutará **nunca**
  - **si la primera vez la condición es cierta** se ejecuta **1 o más veces**

## ATENCION:

- Si la condición **nunca** se hace **falsa**, el programa entra en un **“lazo infinito”**
  - las sentencias del cuerpo del lazo **se ejecutarán indefinidamente**



# *Repita mientras*

- Dado que lo primero que hace la estructura es evaluar la condición
  - Hay que **INICIAR** – dar un valor inicial a **las variables que intervienen en la condición**
  - **antes** de **ejecutar el lazo** por primera vez
- Para garantizar que en algún momento la condición se hace falsa y el lazo pueda terminar su ejecución
  - Hay que estar pendientes de **MODIFICAR dentro del cuerpo del lazo los valores de las variables que intervienen en la condición**

# Repita mientras

## Ejemplo 1

Pseudocódigo en Español	Código en C/C++
<pre>i = 0 Repita mientras (i &lt; MAX)     i = i + 1     Escribir i frm</pre>	<pre>i = 0;      /* inicio contador*/ while(i &lt; MAX) {   i++;          /* incremento*/     printf ("i=%d\n", i); };</pre>

NOTA: Cuando hay mas de una sentencia asociada al **while se escribe { }** en C/C++

# Repita mientras

## Ejemplo 2

Repetición infinita

Pseudocódigo en Español	Código en C
<pre>pot = 0 Repita mientras (pot &lt; 500)     pot = 2*pot     frm Escribir “potencia=”, pot</pre>	<pre>pot = 0; /* inicio acumulador */ while(pot &lt; 500)     pot *=2; /*incremento*/ printf (“potencia= %d\n”, pot);</pre>

**NOTA:** Cuando hay una sola sentencia asociada al **while** **NO** se escribe { } en C/C++

# Repita mientras

## Ejemplo 2

Pseudocódigo en Español	Código en C
<pre>pot = 1 Repita mientras (pot &lt; 500)     pot = 2*pot     frm Escribir “potencia=”, pot</pre>	<pre>pot = 1;                /* inicio acumulador */ while(pot &lt; 500)     pot *=2;             /* incremento */ printf (“potencia= %d\n”, pot);</pre>

**NOTA:** Cuando hay una sola sentencia asociada al **while** **NO** se escribe { } en C/C++

# *Repita mientras*

## Ejemplo 3

Pseudocódigo en Español	Código en C
<pre>ban = cierto <b>Repita mientras</b> (ban)     leer a, b, c     Si(a &lt; 0) entonces         ban = falso     fsi <b>frm</b></pre>	<pre>ban = 1; /* inicio centinela*/ <b>while</b>(ban) {   scanf ("%d%d%d", &amp;a&amp;b&amp;c);     if(a &lt; 0)         ban = 0;    /* cambio de valor*/ <b>};</b></pre>

# Repita mientras

## Ejemplo 4

Nunca entra en el lazo de repetición

Pseudocódigo en Español	Código en C
<pre> a = 1 b = 45 <b>Repita mientras</b> (a ≥ b)     leer a <b>fin</b> </pre>	<pre> a = 1, b = 45; <b>while</b>(a ≥ b)     scanf ("%d", &amp;a); </pre>
<pre> Leer nota <b>Repita mientras</b>(nota &gt; 15)     cont=cont+1     Escribir “Estudiante eximido” <b>fin</b> </pre>	<pre> Scanf ("%d", &amp;nota) <b>while</b>(nota &gt; 15) {     cont++;     printf(“Estudiante eximido\n”); } </pre>



# Repita mientras

## Ejemplo 5

```
#include <stdio.h>
void main( )
{
    int i = 0, suma = 0;
    while(i<5)
    {
        suma += i;
        i++;
        printf ("suma = %d\n", suma);
    }
}
```

Iteración repetición	i	suma
(1)	0	0
(2)	1	0
(3)	2	1
(4)	3	3
(5)	4	6
(6)	5	10

1 a N

0 a N -1

Corrida en frío

# Repita mientras

## Ejemplo 5

```
#include <stdio.h>
void main ()
{
    int i = 1;
    while ( i < 3 )
    { printf ( "i es menor que 3\n");
      i++;
    }
    printf ( " Se termino el lazo\n");
}
```

Corrida en frío

iteración	i
(1)	1
(2)	2
(3)	3



# *Repita mientras*

## Ejercicio 1

Calcular la **suma** de **n** números introducidos por teclado

### **Análisis E-P-S**

**Entradas**: total de números ( $n \in \mathbb{Z}^+$ ),  
números ( $a_1, a_2, a_3, \dots, a_n \in \mathbb{R}$ )

**Proceso**: Calcular  $\text{suma} = a_1 + a_2 + a_3 + \dots + a_n$

**Salidas**:  $\text{suma} \in \mathbb{R}$

# Diseño - algoritmo

sumaNumeros		
{pre:}		{pos: car $\neq$ 'S', $n \in \mathbb{N}$ }
1	n = 0	<ul style="list-style-type: none"> <li>● <b>n</b>: Natural. <b>Contador</b>. Número de valores introducidos por el usuario</li> <li>● <b>suma</b>: Real. <b>Acumulador</b>. Suma total de los n números introducidos por el usuario</li> <li>● <b>car</b>: Caracter. <b>Centinela</b> para controlar el lazo</li> <li>● <b>num</b>: Real. Número suministrado por el usuario para ser sumado.</li> </ul>
2	suma = 0.0	
3	car = 'S'	
4	repita mientras (car = 'S' o car = 's') Escribir "Tienes algún número que sumar? (si -> S o s)" Leer car Si(car = 'S' o car = 's') entonces Escribir "Dame un número " Leer num n = n + 1 suma = suma + num fsi frm	
5	Si(n > 0) entonces Escribir "La suma de los ", n, " numeros dados es= ", suma sino Escribir "No hubo números que sumar" fsi	
1	num = 1, 2, 3, 4, 5 -> car $\neq$ 'S', n = 5, suma = 15	Caso exitoso
2	num = indefinido -> car $\neq$ 'S', n = 0, suma = 0	Caso exitoso



# Codificación del ejercicio 1

```
#include <stdio.h>
int main ( )
{ unsigned int  n = 0;
  float suma=0.0, num;
  char car='S';
  while (car == 'S' || car == 's ')
  {   printf ("Tienes algun numero que sumar? (si -> S o s)\n");
      scanf (% c, &car);
      if(car == 'S' || car == 's ')
      {   printf ("Dame un numero\n");
          scanf (% d, &num);
          n++, suma+=num;
      };
  };
  if(n > 0)
      printf ("La suma de los %d numeros es=\n", suma) ;
  else
      printf ( "No hubo numeros que sumar\n");
}
```

Operador  
coma

Operador coma: indica que debe realizar la sentencia anterior seguida de la siguiente

# Ejercicios

**Para cada uno de los siguientes problemas realizar el análisis E-P-S, diseño, implementación y pruebas**

1. Hallar el resultado de la siguiente sucesión:

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{N}$$

hasta que **1/N** converja hacia **e/100** (igual o menor que un número **e** cualquiera – leer dividido entre **100**)

2. Calcular la sumatoria de los 100 primeros números naturales
3. Calcular independientemente la suma de los números pares e impares comprendidos entre 1 y n



## Ejercicios

4. Escribir todos los enteros positivos menores que 100 omitiendo aquellos divisibles por 7
5. En Caracas se hizo una encuesta de los medios de transporte urbano más comunes. A cada persona se le preguntó si el autobús, el metro o el carro era el medio más usado para ir al trabajo.

Se permitió más de una respuesta. El resultado de la encuesta fue el siguiente:



## Ejercicios continuación del 5

- $n_1$  personas escogieron el autobús
- $n_2$  personas escogieron el metro
- $n_3$  personas escogieron el carro
- $n_4$  personas escogieron el autobús y el metro
- $n_5$  personas escogieron el autobús y el carro
- $n_6$  personas escogieron el metro y el carro
- $n_7$  personas escogieron los tres medios de transporte

- ¿Cuántas personas respondieron la encuesta ?
- ¿Cuántas personas escogieron sólo un medio de transporte ?
- ¿Cuántas personas escogieron dos medios de transporte ?
- ¿Cuántas personas escogieron los tres medios de transporte ?





## Ejercicios

6. Leer valores que representan **años** e indique si son o no años **bisiestos**. El programa seguirá leyendo años hasta un **máximo de 10** o **hasta** que haya leído **3 años bisiestos**

Recuerde, una vez más, la regla:

"Un año es bisiesto si es divisible por 400, o bien si es divisible por 4 pero no por 100"

Por ejemplo, el año 2000 es bisiesto (es divisible por 400), el año 1992 es bisiesto (es divisible por 4 y no por 100) y, el año 2100 no es bisiesto (es divisible por 4 y también por 100)



## Ejercicios

7. Imprimir la tabla de multiplicar de un número dado. EL programa debe pedir el número del 1 al 9

Introduzca un número: 7

**La tabla de multiplicar del 7 es:**

$$7 \times 1 = 7$$

$$7 \times 2 = 14$$

...

$$7 \times 10 = 70$$