



UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

Estructuras de Repetición

hacer-mientras y repita hasta

Prof. Judith Barrios Albornoz

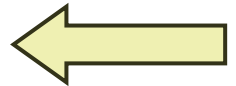
Departamento de Computación
Escuela de Ingeniería de Sistemas
Facultad de Ingeniería
Universidad de Los Andes

Semestre A_2013

Recordando que en Programación estructurada

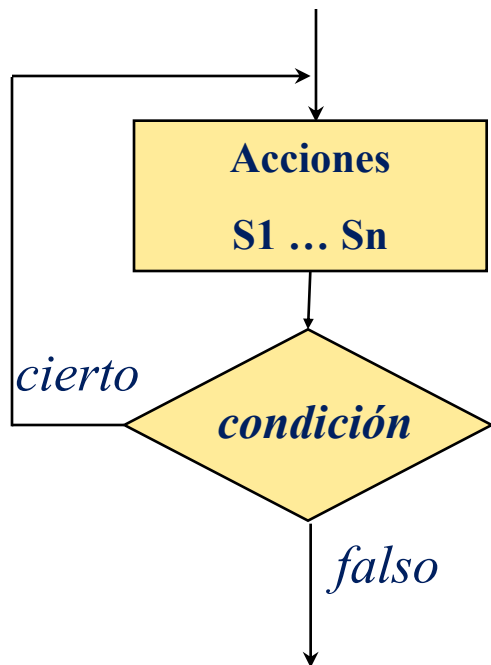
Enfoque disciplinado para escribir programas que utiliza las estructuras de control:

- Secuencial (asignación, lectura, escritura)
- Decisión o selección (simple, doble, múltiple)
- **Repetición** (Repita mientras, Hacer-
Mientras, Repita hasta, Repita para)



Hacer mientras

Diagrama de flujo



Pseudocódigo en español

Hacer

S_1
...
 S_n
mientras (*<condición>*)

Código en C/C++

do
{
 S_1
...
 S_n
}while (*<condición>*);

Hacer mientras

- La estructura “**Hacer-Mientras**” **es** una sentencia “Repita **mientras**” con la diferencia que evalúa la condición **al final** del lazo y no al principio
- Las sentencias (una o más) del cuerpo del lazo **se ejecutan *mientras que la condición* (expresión lógica) es **cierta****
- **El lazo se ejecuta una o más veces** – siempre se evalúa la condición al final de las sentencias
- Atención:
 - **Si la *condición nunca se hace falsa*, el programa entra en un lazo *infinito*, es decir, las sentencias del cuerpo del lazo se ejecutarán indefinidamente**

Ejemplo 1

promedioDeNotas

{pre: numEst $\in \mathbb{N}$ }


{pos: numEst $\in \mathbb{N}$ }

1	contEst = 1	● contEst : Natural.
2	Escribir “Introduzca el numero total de estudiantes”	Contador del número actual de estudiantes.
3	Leer numEst	● numEst : Natural. Número total de estudiantes introducido por el usuario
4	repita mientras (contEst \leq numEst) hacer Escribir “Dame las dos notas del estudiante entre 0 y 20” Leer nota1, nota2 mientras (0 > nota1 > 20 o 0 > nota2 > 20) prom = (nota1 + nota2) / 2 Escribir “Promedio del estudiante= “, prom contEst = contEst + 1 frm	● nota1, nota2 : Real. Notas del estudiante para calcular el promedio del mismo. ● prom : Real. Promedio de las dos notas suministradas por el usuario para el estudiante.
1	numEst = 1, nota1 = 10.0, nota2 = 11.0 -> contEst = 1, prom = 10.5	Caso exitoso
2	numEst = 0 -> contEst = 1	Caso exitoso

Hacer mientras

- Las **variables** que intervienen en la condición pueden **INICIARSE** antes de la primera ejecución del lazo o durante la ejecución – dentro del lazo
- la condición se evalúa al final de la estructura
- Hay que **MODIFICAR** los valores de las variables que intervienen en la condición
 - dentro del cuerpo del lazo - para garantizar que en algún momento ésta se haga falsa y el lazo pueda terminar su ejecución, así se garantiza que el lazo no es infinito

Ejemplo 2

Pseudocódigo en Español	Código en C
hacer Escribir “ a es mayor” mientras ($a \geq b$)	do { printf (“a es mayor\n”); } while ($a \geq b$) <div>  </div>
conta=0 hacer conta=conta+1 leer nota Escribir conta, nota mientras (conta \leq 15)	conta=0; do { conta++; scanf ("%d", ¬a); printf (“%d, %d”, conta, nota\n); } while (conta<= 15)

Nota: Siempre se escribe {} en C/C++

Ejemplo 3

Pseudocódigo en Español	Código en C
<pre>suma=0.0 Leer x hacer suma = suma + x leer x mientras(x > 0.0)</pre>	<pre>suma = 0.0; scanf("%d", &x); do { suma += x; scanf("%d", &x); } while(x > 0.0)</pre>

Ejemplo 3

Pseudocódigo en Español	Código en C
<pre>suma=0.0 hacer leer x suma = suma + x mientras(x > 0.0)</pre>	<pre>suma = 0.0; do { scanf("%d", &x); suma += x; } while(x > 0.0)</pre>

Ejemplo 4

```
#include <stdio.h>
```

```
void main ()
```

```
{
```

```
    int i = 1;
```

```
    do
```

```
    {
```

```
        printf ("%d\n", i);
```

```
        i++;
```

```
    } while ( i <= 3 );
```

```
    printf ("Escribe los números 1, 2 y 3\n");
```

```
}
```

Iteración <i>ejecución</i>	i/i++	i<=3
(1)	1/2	2<=3 si
(2)	2/3	3<=3 si
(3)	3/4	4<=3 no

Corrida en frío

Ejemplo 5

Imprimir los números del 1 al 10

Ejemplo 5

Imprimir los números del 1 al 10

```
#include <stdio.h>

void main ()
{
    int num = 0;
    do
    {
        num ++;
        printf ("%d\n", num);
    } while (num < 10);
}
```

Ejercicio 1

Dados **n** números enteros que se introducirán por teclado, calcular la suma de los números pares, cuántos números pares fueron introducidos y la media aritmética de los números impares dados

Análisis E-P-S

Entrada: n números enteros, $n \in \mathbb{N}$.

Proceso: Para cada número se debe

- determinar si es par (número mod 2 = 0)
- Si es par, se incrementa un contador de pares (**CP**) y se acumula su valor en el acumulador de pares (**AP**)
- Si es impar, se incrementa un contador de impares (**CI**) y se acumula su valor en el acumulador de impares (**AI**)

Calcular la media de impares (**MI** = **AI/CI**)

Salida: suma de los números pares (**AP** $\in \mathbb{Z}$), total de números pares (**CP** $\in \mathbb{N}$) y media aritmética de los números impares (**MI** $\in \mathbb{R}$)

Diseño - algoritmo

paresImpares

{pre: $n \in \mathbb{N}$ }

{pos: $cp, n \in \mathbb{N}, ap \in \mathbb{Z}, mi \in \mathbb{R}$ }

```

1  cp = ci = ap = ai = cn = 0
2  Escribir "Introduzca el numero total de valores"
3  Leer n
4  Si ( n > 0 ) entonces
    hacer
      Escribir "Dame un valor entero"
      Leer num
      Si ( num mod 2 = 0 ) entonces
        cp = cp + 1
        ap = ap + num
      sino
        ci = ci + 1
        ai = ai + num
      fsi
      cn = cn + 1
    mientras(cn < n)
      Si ( ci > 0 ) entonces
        mi = ai / ci
        Escribir "Pares= ", cp, " Suma =", ap, " media de impares=", mi
      sino
        Escribir "Pares= ", cp, " Suma =", ap, " No hubo impares"
      fsi
    sino
      Escribir "No hay valores"
    fsi
  
```

- **cp, ci, cn:** Natural. Contadores del número de valores pares, impares y del número de valores dados, respectivamente.
- **ap, ai:** Entero. Acumulador para la suma de los valores enteros introducidos por el usuario
- **n:** Natural. Número total de valores dados por el usuario.
- **num:** Entero. Valor introducido por el usuario.
- **mi:** Real. Promedio de los valores impares.

Diseño - algoritmo

paresImpares

1	$n = 0 \rightarrow$ No hay valores	Caso exitoso
2	$n = 4$, num = 3,2,5,7 \rightarrow cp = 1, ap = 2, mi = 5.0	Caso exitoso
3	$n = 5$, num = 4,2,8,10,-40 \rightarrow cp = 5, ap = -16, No hubo impares	Caso exitoso

OJO:

Cambiar el ejercicio para que valide que la cantidad de números a leer sea positiva $n > 0$; en caso de que no lo sea que la pida repetidas veces hasta que sea positiva

Codificación en C - ejercicio 1

```
#include <stdio.h>

void main ()
{
    unsigned int  n, cp=0, ci=0, cn=0;
    int          num, ap=0, ai=0;
    float        mi;
    printf ("Introduzca el numero total de valores\n");
    scanf ("%d, &n);
    if(n > 0 )
    {
        do
        {
            printf (" Dame un valor entero\n");
            scanf ("%d, &num);
            if(num%2 == 0)
            {
                cp++;
                ap +=num;
            }
            else
            {
                ci++;
                ai += num;
            }
            cn++;
        }
        while ( cn <n);
        if(ci > 0 )
        {
            mi = float(ai) / float(ci);
            printf ("Total de pares= %d suma de pares= %d media de impares= %d\n ", cp, ap, mi);
        }
        else
            printf ("Total de pares= %d suma de pares= %d " No hubo impares\n", cp, ap);
    }
    else printf ("No hay valores\n");
}
```

Conversión de
tipo (*casting*)

Ejercicio 2

Calcular independientemente la **suma de los números pares e impares** comprendidos entre **1 y n**

Análisis E-P-S

Entrada: Valor de $n \in \mathbb{N}$.

Proceso: Para todos los números comprendidos entre 1 y n:

Si número es par $sp = sp + \text{número}$

Si número es impar $si = si + \text{número}$

Salida: Suma de los números pares ($sp \in \mathbb{N}$) y suma de los números impares ($si \in \mathbb{N}$).

Diseño – Algoritmo

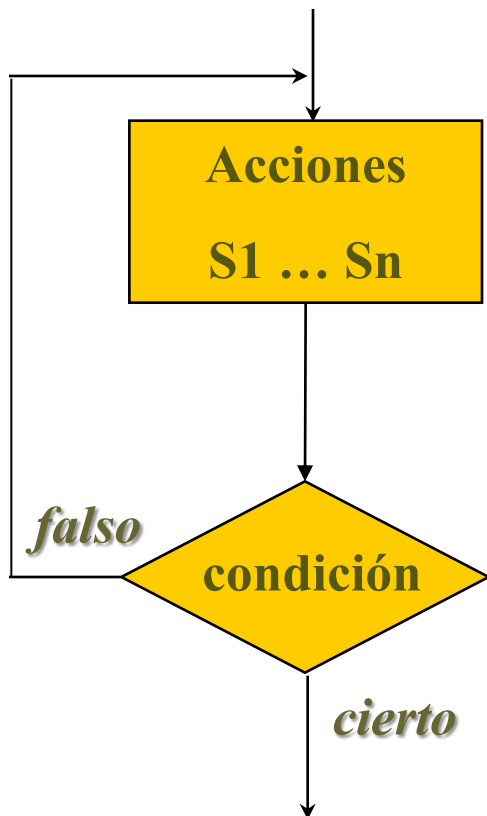
sumaParesImpares		
{pre: $n \in \mathbb{N}$ }		{pos: $n, sp, si \in \mathbb{N}$ }
1	hacer Escribir “Introduzca el valor de $n \geq 1$ ” Leer n mientras ($n < 1$) 2 num = 0, sp = 0, si = 0 3 hacer num = num + 1 si (num mod 2 = 0) entonces sp = sp + num sino si = si + num fsi mientras (num < n) 4 Escribir “Suma de pares entre 1 y “, n, “ es “, sp 5 Escribir “Suma de impares es “, si	● n: Natural. Número máximo para calcular la suma de pares e impares. ● num: Naturales. Números entre 1 y n. ● sp, si: Natural. Suma de los números pares e impares, respectivamente.
1	n = 1 -> n = 1, sp = 0, si = 1	Caso exitoso
2	n = 6 -> n = 6, sp = 12, si = 9	Caso exitoso

Codificación en C

```
#include <stdio.h>
void main ( )
{   unsigned int n, num = 0, nPar = 0, nImpar = 0;
    do                                /* validación del valor de n */
    {   printf ("Introduzca un valor entero mayor que cero\n");
        scanf ("%d",&n);
    } while (n < 1);
    do
    {   num ++;
        if (num % 2 ==0)   sp += num;
        else               si += num;
    } while (num < n);
    printf ("Suma de pares entre 1 y %d es %d\n", n, sp);
    printf (" Suma de impares es %d\n", si);
}
```

Estructura *Repita hasta*

● Diagrama de flujo



Pseudocódigo en español

Repita

S_1

...

S_n

Hasta (<condición>)

Código en C

no existe

Repita hasta

- La estructura “***Repita Hasta***” es una sentencia de repetición que evalúa la condición al final del lazo
- Las sentencias (una o más) del cuerpo del **lazo se ejecutan hasta que la condición** (expresión lógica) **es cierta**

ATENCIÓN: Las sentencias se ejecutan **repetidas** veces **mientras que la condición es falsa...** la repetición se **termina** cuando la condición se hace **cierta**

- Como se pregunta al final por la condición, el lazo se ejecuta **una o más veces**
- **Si** la **condición nunca se hace cierta**, el programa entra en un lazo **infinito**, es decir, las sentencias del cuerpo del lazo se ejecutarán indefinidamente

Repita hasta

- Las variables que intervienen en la condición pueden **INICIARSE** antes de la primera ejecución del lazo o **durante** ésta
 - la evaluación de la condición se hace al final de la estructura

- Hay que **MODIFICAR** dentro del cuerpo del lazo los valores de las variables que intervienen en la condición
 - para garantizar que en algún momento ésta se haga falsa y el lazo pueda terminar su ejecución
 - así garantizar que el lazo no es infinito

Repita hasta

Pseudocódigo en Español	Código en C/C++
<pre>suma=0.0 repita leer x suma = suma + x hasta(x ≤ 0.0)</pre>	<p>Resaltar el modo de representar la <u>equivalencia con el hacer mientras</u> de C/C++</p>

Repetir la acumulación de los valores de X **solamente para los valores de X mayores que 0**

En el momento en que **X es menor ó igual que 0** se **para (deja) de repetir** la acumulación de valores...

Recordemos el Ejemplo 3 de Repita Mientras

Pseudocódigo en Español	Código en C
<pre>suma=0.0 hacer leer x suma = suma + x mientras(x > 0.0)</pre>	<pre>suma = 0.0; do { scanf("%d", &x); suma += x; } while(x > 0.0)</pre>

Resaltar el modo de
representar la
equivalencia con el
hacer mientras de
C/C++

Diseño –Algoritmo – comparar con el diseño del ejercicio 3 de *Repita Mientras*

sumaParesImpares		
{pre: $n \in \mathbb{N}$ }		{pos: $cp, n \in \mathbb{N}, ap \in \mathbb{Z}, mi \in \mathbb{R}$ }
1	fc _p = c _i = a _p = a _i = c _n = 0	<ul style="list-style-type: none"> ● cp, ci, cn: Natural. Contadores del número de valores pares, impares y del número de valores dados, respectivamente. ● ap, ai: Entero. Acumulador para la suma de los valores enteros introducidos por el usuario ● n: Natural. Número total de valores dados por el usuario. ● num: Entero. Valor introducido por el usuario. ● mi: Real. Promedio de los valores impares.
2	Escribir “Introduzca el numero total de valores”	
3	Leer n	
4	Si ($n > 0$) entonces	
	repita Escribir “Dame un valor entero” Leer num Si ($\text{num} \bmod 2 = 0$) entonces $cp = cp + 1$ $ap = ap + \text{num}$ sino $ci = ci + 1$ $ai = ai + \text{num}$ fsi $cn = cn + 1$ hasta ($cn \geq n$)	
	Si ($ci > 0$) entonces $mi = ai / ci$ Escribir “Pares= “, cp, “ Suma =“, ap, “ media de impares=“, mi sino Escribir “Pares= “, cp, “ Suma =“, ap, “ No hubo impares” fsi sino Escribir “No hay valores” fsi	

Diseño - Algoritmo

sumaParesImpares

1	n = 0 -> No hay valores	Caso exitoso
2	n = 4, num = 3,2,5,7 -> cp = 1, ap = 2, mi = 5.0	Caso exitoso
3	n = 5, num = 4,2,8,10,-40 -> cp = 5, ap = -16, No hubo impares	Caso exitoso

La codificación del *Repita hasta* en *C/C++* debe hacerse con la sentencia hacer-mientras

Ejercicios propuestos para ***Repetición condicional***

Para cada uno de los siguientes problemas realizar el análisis E-P-S, algoritmo y codificación:

1. Calcular la **suma** $1+2+3+ \dots + n$, donde **n** es un valor dado. Validar que $n \geq 1$
2. Para el siguiente par de funciones, **encontrar el valor de N** tal que $f(N) < g(N)$ y $N \geq 0$

$$f(N) = 20 N^2 + 100, \quad g(N) = N^2 + 2N + 17$$

3. Dado el balance de su cuenta bancaria del mes anterior y todas las transacciones (retiro/depósito, monto) realizadas durante el presente mes, calcular el balance actual

Ejercicios propuestos

4. El 01 de Enero de 2012, el tanque de agua que se encuentra en la Av. Don *Tulio Febres Cordero*, contenía 10.000 litros de agua. La zona a la cual suministra agua este tanque usa 650 litros de agua semanalmente y el tanque no recibió agua en ningún momento. *Calcular la cantidad de agua que quedó en el tanque al final de cada semana hasta que no había en el tanque suficiente agua para suplir la zona. Diga cuántas semanas duró el agua del tanque.*
5. Calcular la **suma de los cuadrados de los números naturales** entre 1 y n , donde $n^2 < 1000$

Ejercicios propuestos

6. Determinar en un conjunto de n números naturales:
- ¿ Cuántos son menores que 150 ?
 - ¿ Cuántos son mayores que 500 ?
 - ¿ Cuántos están en el rango entre 250 y 450 ?