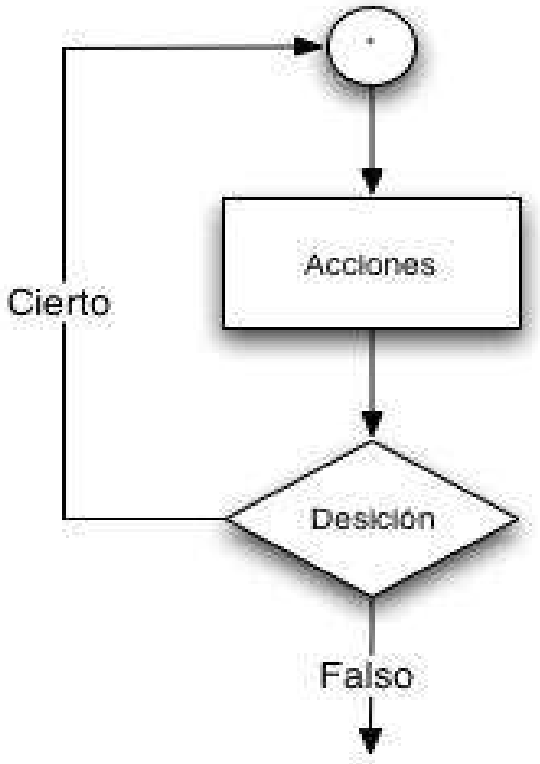
Estructuras de Repetición (Hacer-Mientras)

Material Original: Prof. Flor Narciso
Modificaciones: Prof. Andrés Arcia
Departamento de Computación
Escuela de Ingeniería de Sistemas
Facultad de Ingeniería
Universidad de Los Andes

Estructura de Repetición "Hacer-Mientras"



Pseudocódigo en español

```
Hacer
   S1
   …
   Sn
mientras (<condición>)
```

Código en C

```
do
{
    S1
    …
    Sn
} while (<condición>);
```

Estructura de Repetición “Hacer-Mientras”

La estructura “Hacer-Mientras” es una sentencia del tipo “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.

Como se pregunta al final por la condición, el lazo se ejecuta una o mas veces.

Si la condición nunca se hace falsa, las sentencias del cuerpo del lazo se ejecutarán indefinidamente.

Estructura de Repetición “Hacer-Mientras”

Las variables que intervienen en la condición pueden INICIALIZARSE antes de la primera ejecución del lazo o durante ésta, ya que la evaluación de la condición se hace al final de la estructura.

Hay que estar pendientes de MODIFICAR dentro del cuerpo del lazo los valores de las variables que intervienen en la condición. La modificación de estos valores debe estar en aras de garantizar que en algún momento la condición se haga falsa y el lazo pueda terminar su ejecución.

Estructura de Repetición "Hacer-Mientras": Ejemplos

Pseudocódigo en Español	Código en C
LAZO INFINITO: **hacer** ** Escribir ("A es mayor")** **mientras (A>=B)**	**do** **{** ** printf("A es mayor \n");** **} while (A>=B)**
contador = 1 **exponente = 0** **hacer** ** contador = contador * 2** ** exponente++** **mientras (contador ≤ 2048)** **Si 2^exponente = contador** ** Escribir("resultado correcto")** **Fin_Si**	**int contador = 1** **int exponente = 0** **do {** ** contador *= 2;** ** exponente++;** **} while (contador <= 2048)** **if (pow(2,exponente) == contador)** ** printf("resultado correcto");**

Estructura de Repetición "Hacer-Mientras": Ejemplos

Pseudocódigo en Español	Código en C

`Escribir("Para salir escriba un número negativo")`
`suma = 0.0`
`Leer(x)`
`Hacer`
`  Si (x > 0.0)`
`  suma = suma + x`
`  Leer(x)`
`  Fin_Si`
`Mientras (x > 0.0)`
	`float suma=0.0;`
	`printf("Para salir escriba un número negativo");`
	`scanf("%f",&x);`
	`do`
	`{`
	`  if (x>0.0)`
	`  {`
	`  suma+=x;`
	`  scanf("%f",&x);`
	`  }`
	`} while (x>0.0)`

Estructura de Repetición "Hacer-Mientras": Ejemplo 1

```
#include <stdio.h>

int main ()
{
   int i = 1;
   do
   {
        printf("i = %i\n",i);
        i++;
   } while ( i <= 3 );
   printf("Escribe los numeros 1,
    2 y 3" endl;
}
```

Corrida en frío

Iteración	i
(0)	1
(1)	2
(2)	3
(3)	4

Estructura de Repetición “Hacer-Mientras”: Ejemplo 2

Imprimir los números del 1 al 10

```
#include <stdio.h>

int main ()
{
   int num = 0;

   do
   {
        num ++;
        printf(“%i \n”,num);
   } while (num < 10);
}
```

Estructura de Repetición "Hacer-Mientras": Ejemplo 3

Dados 10 números enteros que se introducirán por teclado, calcular la suma de los números pares, cuántos números pares existen y la media aritmética de los números impares.

Análisis E-P-S

Entradas: Diez números enteros

Proceso: Para cada número se debe

- Determinar si es par (número mod 2 = 0)
- Si es par si 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)

Salidas: suma de los números pares (AP Z), total de números pares (CP Z) y media aritmética de los números impares (MI)

Estructura de Repetición "Hacer-Mientras": Ejemplo 3

Algoritmo

```
0. Inicio
1. conta = 0
2. CP = 0
3. CI = 0
4. AP = 0
5. AI = 0
6. Hacer
    Leer (num)
   Si (num mod 2 = 0) entonces
       CP = CP + 1
       AP = AP + num
   sino
       CI =  CI + 1
       AI = AI + num
   fin_si
    conta =  conta + 1
   mientras (conta < 10)
7. Si (CI ≠ 0) entonces
    media = AI/CI
    Escribir (AP, CP, media)
   sino
     Escribir (AP, CP)
     Escribir("No hay
     impares")
    fin_si
9. Fin
```

Estructura de Repetición "Hacer-Mientras": Ejemplo 3

Codificación

```
#include <stdio.h>
void main ()
{
   int conta = 0, CP = 0, CI = 0;
   int AP = 0, AI = 0, num;
   float media;
   do
   {
    printf("numero ?\n");
    scanf("%i",&num);
    if (num % 2 == 0) {
       // par
      CP++;
      AP += num;
   }
```

```
        else { // impar
               CI++;
               AI += num; }
       conta++;
  } while (conta < 10);
  if (CI != 0) {
  media =(float)AI/(float)CI;
  printf("%i %i %i",AP,CP,media);
  } else {
  printf("%i %i",AP,CP);
  printf("No hay numeros
          impares\n");
   }
}
```

Estructura de Repetición "Hacer-Mientras": Ejemplo 4

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

Análisis E-P-S

Entradas: Valor de n $\in Z^+$.

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

Si número es par nPar = nPar + numero

Si número es impar nImpar = nImpar + numero

Salidas: Suma de los números pares (nPar $\in Z^+$) y suma de los números impares (nImpar $\in Z^+$).

Estructura de Repetición "Hacer-Mientras": Ejemplo 4

Algoritmo

```
0. Inicio
1. hacer
   Escribir ("Introduzca
   el valor de n ≥ 1")
    Leer (n)
   mientras (n >= 1)
2. numero = 0
3. nPar = 0
4. nImpar = 0
```

```
5. hacer
numero = numero + 1
Si (numero mod 2 = 0) entonces
   nPar = nPar + numero
   sino
   nImpar = nImpar + numero
   fin_si
   mientras (numero < n)
6. Escribir ("Suma de numeros
                    pares",
   nPar)
7. Escribir ("Suma de numeros
                impares", nImpar)
8. Fin
```

Estructura de Repetición "Hacer-Mientras": Ejemplo 4

Codificación

```
#include <stdio.h>

void main ()
{
   unsigned int n, numero
   = 0, nPar = 0, nImpar =
   0;

   do    // validación del
   valor de n
   printf("Introduzca un
   valor entero mayor que
   cero\n");
  scanf("%i",&n);
   } while (n >= 1);
```

```
do {
numero ++;
if (numero % 2 ==0)
  nPar += numero;
else
  nImpar += numero;
} while (numero < n);
   printf("Suma de numeros
   pares %i\n",nPar);
   printf("Suma de numeros
   impares %i\n",nImpar);
}
```

Ejercicios

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

Calcular iterativamente la suma 1+2+3+ ...+n, donde n es un valor dado. Validar que $n \geq 1$.

Para el siguiente par de funciones, encontrar el valor de N tal que f(N) > g(N)

$f(N) = 20\ N^2 + 100,\ g(N) = N^3 + 2N + 17$

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

El 1 de Enero de 1999, el tanque de agua “Tulio Febres Cordero” contenía 10.000 litros de agua. La zona a la cual suministra agua este tanque usó 183 litros de agua semanalmente y el tanque no recibió agua en ningún momento. Calcular la cantidad de agua que queda en el tanque al final de cada semana hasta que no quede en el tanque suficiente agua para suplir la zona.

Calcular la suma de los cuadrados de los cien primeros números naturales.

Ejercicios

Determinar en un conjunto de n números naturales:

- ¿ Cuántos son menores que 15 ?
- ¿ Cuántos son mayores que 50 ?
- ¿ Cuántos están en el rango entre 25 y 45 ?

Hacer un programa que determine la secuencia de números partiendo de un número cualquiera *n*, y parar cuando la secuencia finalice en 1. Los números intermedios se calculan de la siguiente manera:

- Si es impar, siguiente = anterior * 3 + 1
- Si es par, siguiente = anterior / 2

Ejercicios

Correr en frío el siguiente segmento de programa y determinar su función:

```
#define true 1
#define false 0
char accept;
float x;
float low, high;

do {
    printf(”Introduzca un valor entre (%f y %f)”,low, high);
    scanf(“%f”,&x);
    if (low <= x && x <= high)
      accept = true;
    else
      accept = false;
  } while (!accept);
```