LICENCIATURA EN INFORMATICA

PROGRAMADOR UNIVERSITARIO

INGENIERÍA EN INFORMÁTICA

PROGRAMACIÓN

Unidad N° 3 – Parte 3. Estructuras de Control Iterativas. Sintaxis y semántica. Diferencias entre las distintas construcciones

2022

Lic. Mariela A. Velázquez

Repasemos lo visto

Operador ternario: ?

Resul = (num1>num2)? num1:num2

Conversiones:

Si i es entero

i= 34,78 → i = 34

Si j es real

j = 34 → j=34.00

Relacionales
Lógicos
Asignación
Aritméticos

PRIORIDADES DE OPERADORES:
(), ….., = += *=

char – 1 byte

float – 4 byte

int – 2 a 4 byte

double – 8 byte

Operadores

Tipos de datos simples

Sentencia if-else

```
if(expresión)
{
   proposición 1;
}
else{  proposición 2;
                }
```

Calificadores enteros:
short < int < long

Calificadores reales:
Float < double < long double

Estructuras de Selección

```
switch(exp)
{  case const1: sent1;
                Break;
                …
   Default: sentn;
}
```

Estructura Iterativa: MIENTRAS()

Son utilizados para repetir acciones bajo un condición

Su notación es:

```
MIENTRAS(condición)
        acciones
Fin_mientras
```

--> Condición: La condición puede ser simple o compuesta. En este último caso estará "unida" por una conjunción (y) o una disyunción (o).

Material sacado del canal de Youtube de la cátedra de Elementos de Computación y Lógica. (https://youtu.be/fp4qdLKi0Ms)

Estructura Iterativa: MIENTRAS()

Veamos un ejemplo de su funcionamiento

Analizando un sector de pseudocódigo

Estructura Iterativa: MIENTRAS()

```
...
LEER(edad)
MIENTRAS( edad > 0)
    LEER(dni)
    SI (edad >= 18) ENTONCES
        ESCRIBIR(dni)
    Fin_si
    LEER(edad)
Fin_mientras
...
```

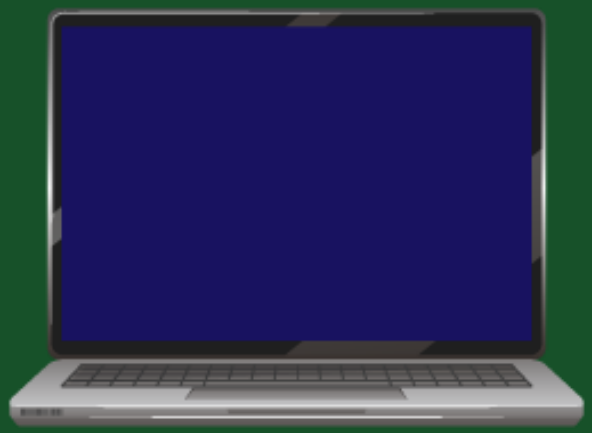
Lee la edad de la primera persona ingresada

Controla si edad es mayor a 0. Como edad = 5 por lo tanto ingresa al bucle.

Lee dni.

Controla si edad es mayor a 18. Como edad = 5 entonces no ingresa a la selección

Lee la próxima edad



edad	dni
5	53.321.125
20	

Estructura Iterativa: MIENTRAS()

```
...
LEER(edad)
MIENTRAS( edad > 0)
    LEER(dni)
    SI (edad > 18) ENTONCES
        ESCRIBIR(dni)
    Fin_si
    LEER(edad)
Fin_mientras
...
```

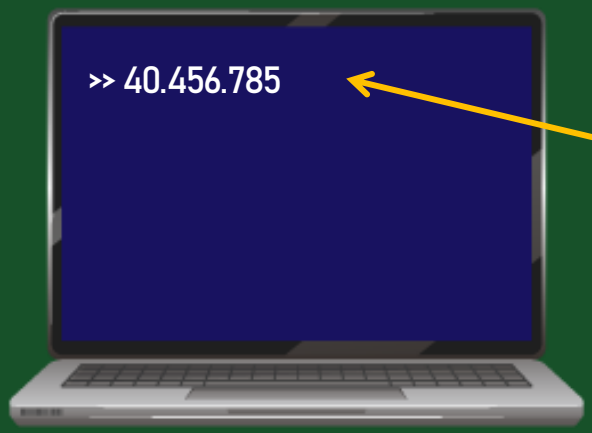
Controla si edad es mayor a 0. Como edad = 20 por lo tanto ingresa al bucle.

Lee dni

Controla si edad es mayor a 18. Como edad = 20 entonces ingresa a la selección.

Escribe el dni

Lee la próxima edad



edad	dni
5	53.321.125
20	40.456.785
18	

Estructura Iterativa: MIENTRAS()

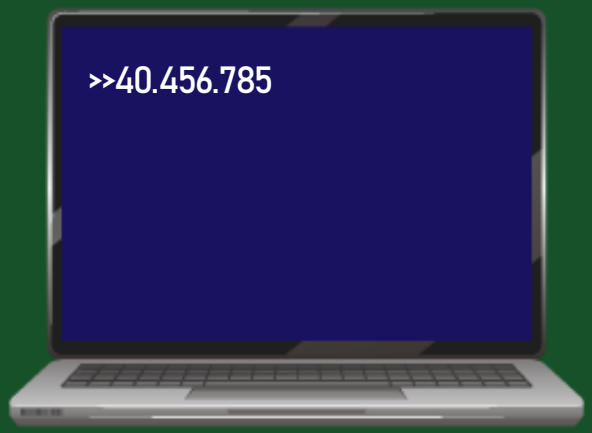
```
...
LEER(edad)
MIENTRAS( edad > 0)
    LEER(dni)
    SI (edad > 18) ENTONCES
        ESCRIBIR(dni)
    Fin_si
    LEER(edad)
Fin_mientras
...
```

Controla si edad es mayor a 0. Como edad = 18 por lo tanto ingresa al bucle.

Lee dni

Controla si edad es mayor a 18. Como edad = 18, no mayor, entonces no ingresa a la selección.

Lee la próxima edad



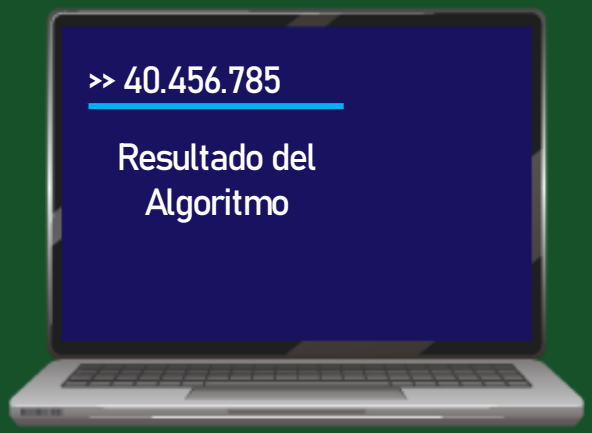
edad	dni
5	53.321.125
20	40.456.785
18	43.128.875
0	

Estructura Iterativa: MIENTRAS()

```
...
LEER(edad)
MIENTRAS( edad > 0)
    LEER(dni)
    SI (edad > 18) ENTONCES
        ESCRIBIR(dni)
    Fin_si
    LEER(edad)
Fin_mientras
...
```

Controla si edad es mayor a 0. Como edad = 0 y no mayor, por lo tanto no cumple con la condición.

Sale de la estructura Mientras



edad	dni
5	53.321.125
20	40.456.785
18	43.128.875
0	

Estructura Iterativa: MIENTRAS()

A tener en cuenta:

- Es conveniente usar esta estructura cuando NO sabemos con anterioridad la cantidad de veces que se repetirá una serie de acciones.
- Se debe tener en cuenta que para evitar un bucle infinito (esto implica que no se puede salir de la estructura iterativa) siempre se debe leer un nuevo dato o incrementar/decrementar dentro de la estructura, ya que ese dato forma parte de la condición.

  Por ejemplo:

```
LEER(cantidad)
MIENTRAS(cantidad > 0)
        acciones
        cantidad ← cantidad - 1
Fin_mientras
```

Si no existiera esta línea, y cantidad fuera igual a 4, no saldría de la estructura, debido a que ese dato jamás es modificado

Sentencia While

La sentencia while ejecuta una sentencia, simple o compuesta, cero o más veces, dependiendo de una condición.

Su sintaxis es:

while (condición) sentencia ;

Donde:

condición es cualquier expresión numérica, relacional o lógica.

sentencia es una sentencia simple o compuesta.

Sentencia While

while (condición) sentencia ;

La ejecución de la sentencia while sucede así:

Se evalúa la condición . Si el resultado de la evaluación es 0 (falso), la sentencia no se ejecuta y se pasa el control a la siguiente sentencia en el programa. Si el resultado de la evaluación es distinto de 0 (verdadero), se ejecuta la sentencia y el proceso descrito se repite desde el Inicio.

Ejemplo

Ya conocemos la sintaxis de la estructura **while()** en C

Entonces estamos listos para implementar el algoritmo visto en el ejemplo

Ejemplo

Algotimo MayorDeEdad
ENTRADA: edad, dni: enteros
SALIDA: dni: entero

A1. LEER(edad)

A2. MIENTRAS(edad > 0)

LEER(dni)

SI (edad >= 18) ENTONCES

ESCRIBIR(dni)

Fin_si

LEER(edad)

Fin_mientras

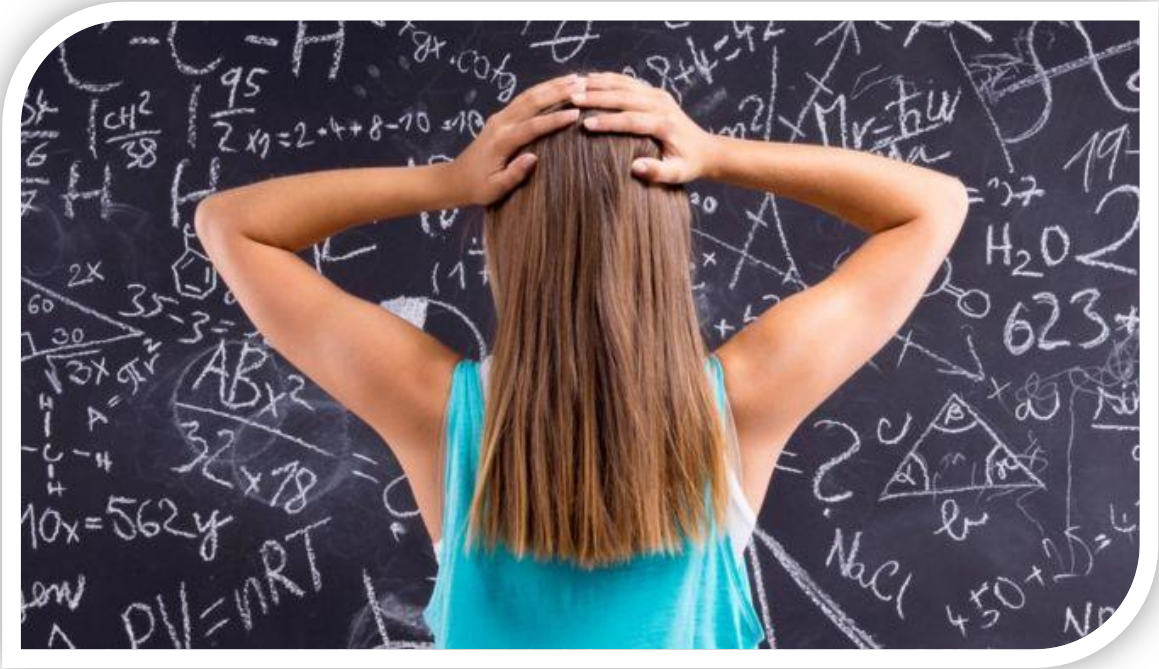
A3. PARAR

```c
#include <stdio.h>

int main()
{
    int edad, dni;

    printf("Ingrese la edad: ");
    scanf("%d", &edad);

    while(edad >0)
    {
        printf("Ingrese el dni: ");
        scanf("%d", &dni);
        if(edad >=18)
        {
            printf("La persona con DNI %d es mayor de edad", dni);
        }

        printf("\n Ingrese la proxima edad: ");
        scanf("%d", &edad);
    }

    return 0;
}
```

Ejercicio

Diseñar un algoritmo que muestre la tabla de multiplicar de un número determinado.

Además se solicita que muestre por pantalla hasta que número se desea ver la tabla. Implemente en C.



Sentencia do-while

En el lazo do-while tiene la comprobación relacional al final, en vez de tenerla al inicio como es en la estructura while. La sintaxis es:

```
do {
        sentencia;
}while ( condición );
```

La sentencia se ejecuta y después se evalúa la condición. Si es verdadera la proposición se evalúa de nuevo. Cuando la expresión se hace falsa el ciclo termina.

Sentencia do-while

- Retomar el ejemplo anteriormente realizado usando la estructura do-while.

- ¿Qué diferencias nota entre while y do-while?.

Estructura Iterativa: HACER()

Son utilizados para repetir un número predeterminado o fijo de veces

Su notación es: HACER 10 VECES (i = 1, ..., 10)

acciones

Fin_hacer

Variable i:

--> Indica el número de iteraciones.
--> Se debe declarar como variable auxiliar.
--> En el diseño de algoritmo es opcional su uso.
--> En muchas ocasiones se usa dentro del bucle.

Material sacado del canal de Youtube de la cátedra de Elementos de Computación y Lógica. (https://youtu.be/mWnoklYptZE)

Estructura Iterativa: HACER()

Veamos un ejemplo de su funcionamiento

Analizando un sector de pseudocódigo

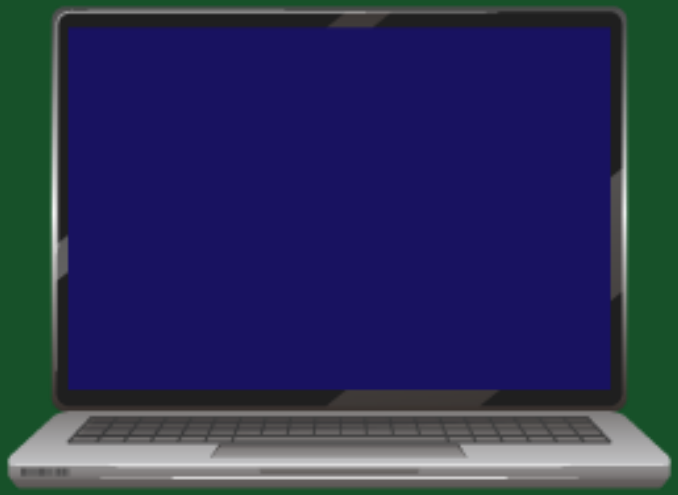
Estructura Iterativa: HACER()

```
...
HACER 4 VECES (i= 1,...,4)
    LEER(dni, edad)
    SI (edad > 18) ENTONCES
        ESCRIBIR(dni)
    Fin_si
Fin_hacer
...
```

Ingresa con i=1

Lee los 2 datos solicitados

Controla si edad es mayor a 18. Como edad = 5 entonces no ingresa a la selección



i	dni	edad
1	53.321.125	5

Estructura Iterativa: HACER()

```
...
HACER 4 VECES (i= 1,...,4)
    LEER(dni, edad)
    SI (edad > 18) ENTONCES
        ESCRIBIR(dni)
    Fin_si
Fin_hacer
...
```

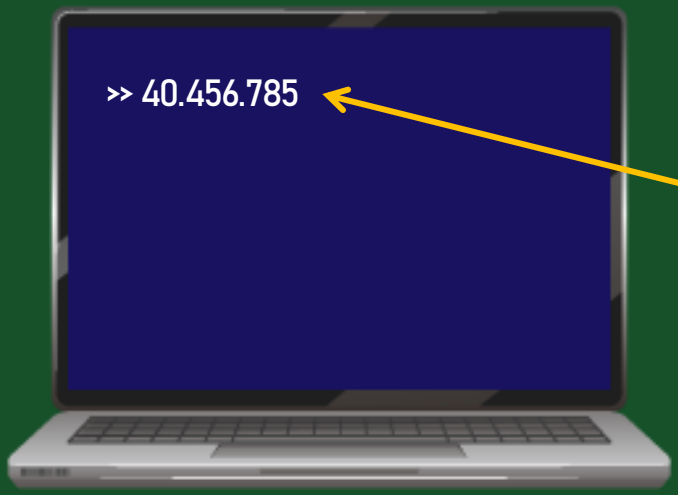
Ingresa con i=2

Lee los 2 datos solicitados

Controla si edad es mayor a 18. Como edad = 20 entonces ingresa a la selección

Escribe dni

Sale de la selección



i	dni	edad
1	53.321.125	5
2	40.456.785	20

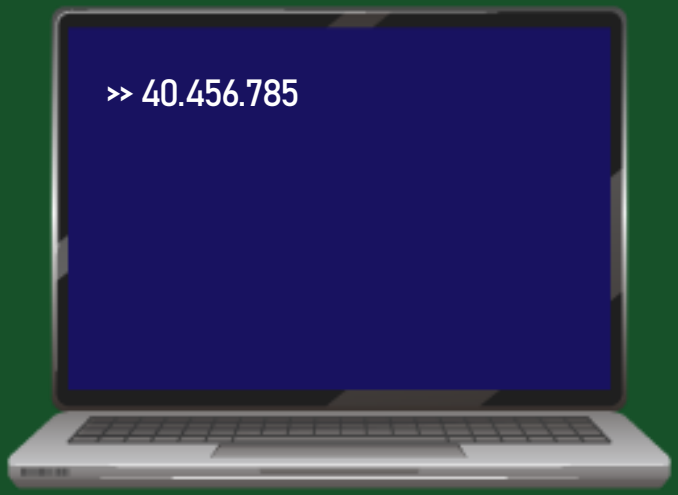
Estructura Iterativa: HACER()

```
...
HACER 4 VECES (i= 1,...,4)
    LEER(dni, edad)
    SI (edad > 18) ENTONCES
        ESCRIBIR(dni)
    Fin_si
Fin_hacer
...
```

Ingresa con i=3

Lee los 2 datos solicitados

Controla si edad es mayor a 18. Como edad = 18, no mayor, entonces no ingresa a la selección



i	dni	edad
1	53.321.125	5
2	40.456.785	20
3	43.128.875	18

Estructura Iterativa: HACER()

```
...
HACER 4 VECES (i= 1,...,4)
    LEER(dni, edad)
    SI (edad > 18) ENTONCES
        ESCRIBIR(dni)
    Fin_si
Fin_hacer
...
```

Ingresa con i=4

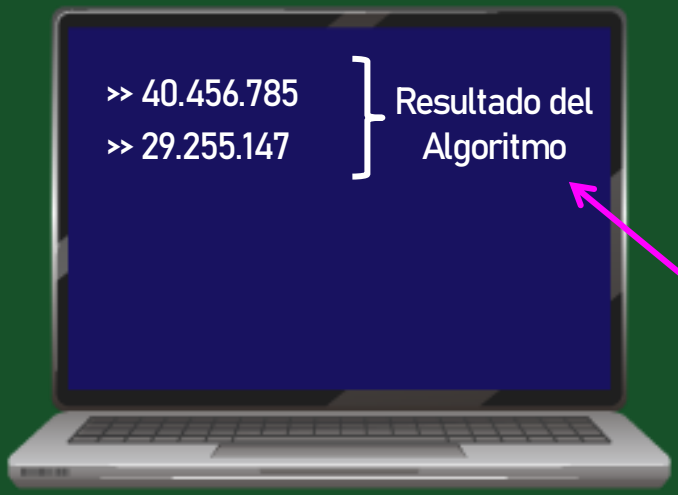
Lee los 2 datos solicitados

Controla si edad es mayor a 18.
Como edad = 37 ingresa a la selección

Escribe dni

Sale de la selección

Sale del Hacer



i	dni	edad
1	53.321.125	5
2	40.456.785	20
3	43.128.875	18
4	29.255.147	37

Estructura Iterativa: HACER()

A tener en cuenta:

- Es conveniente usar esta estructura cuando sabemos con anterioridad la cantidad de veces que se repetirá una serie de acciones.
- Podemos solicitar al usuario final que ingrese la cantidad de veces que desea que se ejecute la estructura.

  Por ejemplo:

```
LEER(cantidad)
HACER cantidad VECES (i=1,.., cantidad)
        acciones_a_repetir
Fin_Hacer
```

Sentencia For

A ver Marta, pase y escriba 500 veces no debo tirar papeles en clases…

```
#include <stdio.h>

int main()
{
     int cont;

      for(cont=1;  cont<=500;  cont++)
     {
          printf("no debo tirar papeles en clases");
      }
      return 0;
}
```

desmotivaciones.es

Cosas de programadores.

Sentencia for

La sentencia for permite ejecutar una sentencia simple o compuesta, repetidamente un número de veces conocido.

Su sintaxis es la siguiente:

for (expresión-comienzo; condición; progresión-condición)

Sentencia;

Expresión-comienzo: representan variables de control que serán iniciadas con los valores de las expresiones.

Condición: es una expresión booleana que si se omite, se supone verdadera.

Progresión-condición: es una o más expresiones separadas por comas cuyos valores evolucionan en el sentido de que se cumpla la condición para finalizar la ejecución de la sentencia for.

Sentencia: es una sentencia simple o compuesta.

Ejemplo

Ya conocemos la sintaxis de la estructura **for()** en C

Entonces estamos listos para implementar el algoritmo visto en el ejemplo

Ejemplo

Algotimo MayorDeEdad
ENTRADA: edad, dni: enteros
SALIDA: dni: entero

A1. HACER 4 VECES (i= 1,…,4)

    LEER(dni, edad)

    SI (edad >= 18) ENTONCES

        ESCRIBIR(dni)

    Fin_si

Fin_hacer

A2. PARAR

```c
#include <stdio.h>

int main()
{
    int edad, dni;

    for (int i = 0; i < 4; i++)
    {
        printf("\n Ingrese la edad: ");
        scanf("%d", &edad);

        printf("\n Ingrese el dni: ");
        scanf("%d", &dni);
        if(edad >=18)
        {
            printf("La persona con DNI %d es mayor de edad", dni);
        }
    }

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    printf("Hasta la próxima clase!!\n");

    return 0;
}
```