```
int main(int argc, char** argv)
{
    Mat img = imread(argc > 1 ? argv[1] : "lena.jpg",1); // open the image
    if(img.empty()) // check if we succeeded
    {
        cout << "can not load image\n";
        return 0;
    }
    help();

    Size s=img.size();
    int w=s.width, h=s.height;
    int ro0=3; //radius of the blind spot
    int R=120;  //number of rings

    //Creation of the four different objects that implement the four log-polar transformatio
    //Off-line computation
                ter(w/2,h/2);
                                                    NEAREST)
```

UAEM | Universidad Autónoma del Estado de México

Centro Universitario UAEM Zumpango

Ingeniería en Computación

Programación estructurada

CENTRO UNIVERSITARIO UAEM ZUMPANGO

Agosto 2015

Identificación de la Unidad de Aprendizaje (UA)

Nombre UA:

Programación esctructurada (L41012)

Total de horas a la semana: **6** Créditos: **9**

Carácter de la UA: **Obligatoria** Modalidad: **Presencial**

UA Antecedente: **Ninguna**

UA Consecuente: **Programación avanzada**

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Propósito de la unidad de aprendizaje

Aplicar el paradigma de la programación estructurada para representar en términos de pseudocódigo, la solución de problemas reales automatizables, mostrando en ella el dominio de variables simples, vectores, matrices, registros y modularidad.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Contenido

1. Identificar las fases de la metodología de programación estructurada para la solución de problemas.
2. Aplicar la programación estructurada en la solución de problemas utilizando diagramas de flujo y pseudocódigo.
3. Utilizar arreglos unidimensionales y bidimensionales para el almacenamiento de datos en la solución de problemas.
4. Usar las técnicas de programación modular en el desarrollo de programas informáticos.
5. Utilizar los registros para almacenar y manipular información en el desarrollo de programas informáticos.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Introducción

El Ingeniero en Computación es el profesional que posee los conocimientos y habilidades en el desarrollo de sistemas computacionales, diseño y mantenimiento de hardware, comunicaciones y redes de computadoras así como en la administración de recursos computacionales.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Introducción

Una de las principales características del Ingeniero en Computación es la habilidad que tiene para desarrollar aplicaciones que apoyan a la solución de problemas. Esta habilidad se adquiere durante la formación del profesionista e inicia desde los primeros semestres y va madurando continuamente.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Introducción

La programación evoluciona constantemente, sin embargo, la programación estructurada en otros paradigmas de programación es el conocimiento base y permite la adquisición de habilidades de programación que se requieren en la construcción de aplicaciones de mediana y alta complejidad.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Introducción

Esta unidad de aprendizaje tiene la finalidad de introducir al alumno al ámbito de la programación en sus elementos básicos y proporcionarle los conocimientos necesarios y suficientes para que utilice estructuras de datos básicas en la programación, registros y programación modular, para la solución de problemas informáticos

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Unidad de competencia II

Aplicar la programación estructurada en la solución de problemas utilizando lenguaje informal y diagramas de flujo

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Conocimientos

- Caracterizar las diferentes estructuras de datos
- Caracterizar las diferentes estructuras de control de flujo
- Conocer las instrucciones básicas
- Describir las carácterísticas de un pseudocódigo
- Describir los pasos para realizar una prueba de escritorio

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Conocimientos

- Elaborar programas que utilizan estructuras de control de flujo
- Solucionar problemas utilizando estructuras de control de flujo
- Utilizar instrucciones básicas en la elaboración de programas
- Diseñar programas utilizando diagramas de flujo
- Diseñar programas utilizando pseudocódigo
- Realizar la prueba de escritorio de un programa
- Disen ▫ar creativamente las diferentes soluciones de un programa informático

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Para representar programas o algoritmos, en este curso se utilizarán dos herramientas conceptuales:

- Pseucodódigo
- Diagramas de flujo

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Pseudocódigo

Joyanes [1] dice “ El pseudocógido es un lenguaje de especificación (descripción de algoritmos).”

“La ventaja del pseudocódigo es que en su uso, en la planificación de un programa, el programador se puede concentrar en la lógica y en las estructuras de control y no preocuparse de las reglas de un lenguaje específico.”

Pseudocódigo

Cada programador puede tener su propio pseudocódigo, por ejemplo:

Inglés	Español	Lenguaje C
start	inicio	{
end	fin	}
read	leer	scanf
write	escribir	printf
If – then - else	si – entonces - sino	If - else
Repeat - until	Repetir - hasta	Do - while
While	Mientras	while

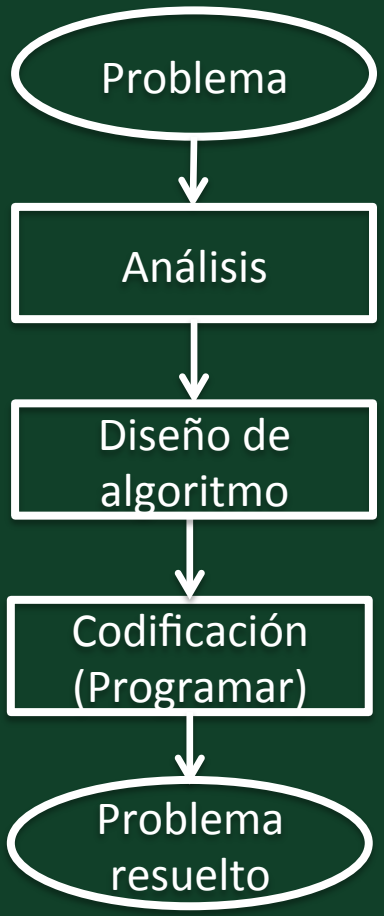
M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Diagramas de flujo

Otra vez Joyanes[1] nos dice: ”Un diagrama de flujo (flowchart) es un diagrama qiue utiliza símbolos (cajas) estándar … y que tienen los pasos del algoritmo escritos en esas cajas unidas por flechas, denominadas líneas de flujo, que indican la secuencia en que se debe ejecutar”

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Relación entre problemas y algoritmo



M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Diagramas de flujo

En pocas palabras un diagrama de flujo es la representación gráfica de un algoritmo.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Símbolo	Paso o instrucción del algoritmo	Símbolo	Paso o instrucción del algoritmo
	Inicio , Fin		Toma de decisión
	Operación, Proceso		Líneas de flujo
	Entrada / salida de datos		Ciclo o bucle
	Función, Subproceso		Conectores

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Diagramas de flujo
Reglas

- Deben tener un inicio y un fin.
- Se debe realizar de abajo hacia arriba y de izquierda a derecha.
- Las líneas de flujo deben ser horizontales y verticales.
- El diagrama no tendra más de 10 símbolos por hoja.
- Los “adelantos” serán por la derecha y los “regresos” por la izquierda.
- Los símbolos, a excepción de las tomas de decisión, solo tienen una entrada y una salida.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Ejemplos pseudocódigo, diagrama de flujo

De cada ejemplo realiza el pseudocódigo, el diagrama de flujo y códificalo en lenguaje C.

Adquiere desde el teclado 2 números, realiza la suma, la resta, la multiplicación y muestra los resultados en la pantalla.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Ejemplos

Desplegar en pantalla tu nombre(s) y apellidos, guardar el programa como nombre.c

Desplegar en pantalla los números de 1 al 10 con un salto de línea entre ellos, guardar el programa con el nombre numeros.c

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Ejemplos entrada salida de datos

En una base de datos de personas, se específica el género de la misma con las letras ‘M’ en caso de las mujeres y ‘H’ en caso de los hombres.

Adquiere desde teclado tu edad, peso, estatura y el carácter de tu género, despliega todos los valores capturados en pantalla.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Ejemplos pseudocódigo, diagrama de flujo

Adquiere desde el teclado el precio de un artículo, obten el IVA y muestralo en la pantalla (IVA = precio * 1.15).

Adquiere desde el teclado el radio de un círculo, calcula el área y el perímetro, muestra los resultados en pantalla.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Ejemplos pseudocódigo, diagrama de flujo

El volumen de una piramide rectangular se calcula por medio de la fórmula

$$\text{Volumen} = \frac{\text{área base}}{3} \times \text{altura}$$

Adquiere desde el teclado el lado y la altura, calcula en volumen y muestralo en pantalla.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Ejemplos pseudocódigo, diagrama de flujo

La ecuación de la pendiente de una recta es:

$$m = \frac{y_1 - y_2}{x_1 - x_2}$$

Asigna a las coordenadas valores y calcula la pendiente de la recta.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estucturas de control del programa

Los programas de computadora desarrollados bajo el paradigma de Programación estructurada tienen como característica que están formados por tres tipos de estructuras de control:

- Secuenciales
- Selectivas
- Iteración

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura de control **secuencial**

Tiene las siguientes características

- Las instrucciones se ejecutan en el orden escrito
- Cada una se ejecuta después de la anterior
- No hay saltos

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura de control **selectiva**

Tiene las siguientes características

- Permiten elegir si ejecutan una instrucción u otra
- Usan una condición boolena para realizar la elección

Nota:

Esto le proporciona a la computadora la capacidad para tomar decisiones. Los saltos se realizan hacia adelante.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura de control **selectiva en lenguaje C**

Una condición booleana es una expresión que puede tener como resultado únicamente los valores verdadero o falso.

Por lo que puede usar operadores relacionales (>, <, ==) o lógicos (||,&&)

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura de control **selectiva en lenguaje C**

Las estructuras de control selectivas en este lenguaje de programación son:

- if (si)
- if-else (si – de lo contrario)
- switch (conmutación)

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura de control selectiva en lenguaje C

if (si)

Se le conoce como “toma de decisión”, o estructura condicional simple.

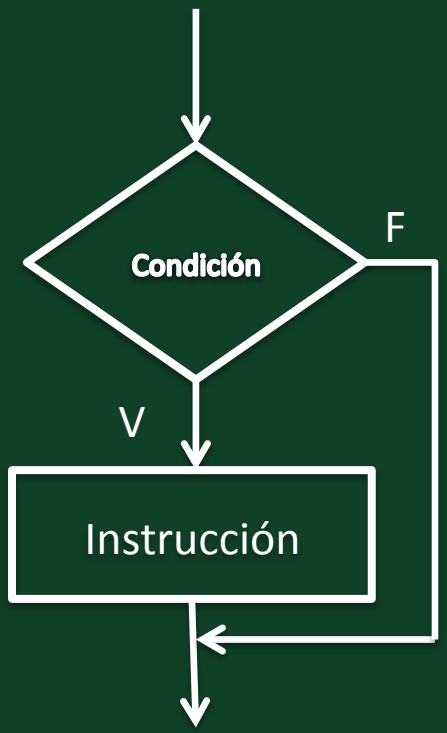
Evalúa una condición que si es verdadera realiza las instrucciones definidas para ella.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura selectiva en lenguaje C: if (si)

Pseudocódigo

```
Si (condición es verdadera)
Inicio
      instrucción 1
      instrucción 2
            :
      instrucción N
Fin
```



NOTA: Siempre hacia "adelante"

Ejemplo en Lenguaje C

```
If (edad >= 18)
{
    printf("Hola");
    printf("Eres mayor");
    mayor=1;
}
```


M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura de control selectiva en lenguaje C

if-else (si – de lo contrario)

Se le conoce como estructura condicional doble.

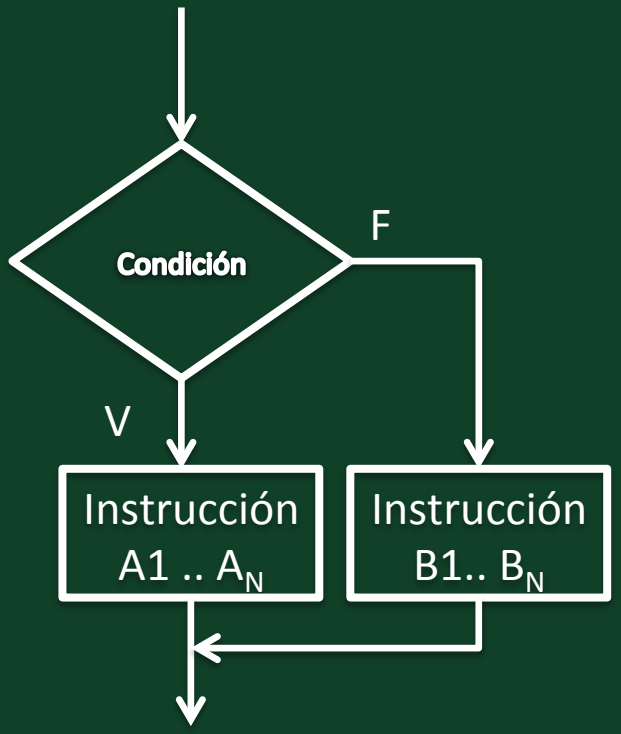
Permiten elegir entre dos alternativas posibles en función del cumplimiento o no de una determinada condición.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura selectiva en lenguaje C: if - else

Pseudocódigo

Si (condición es verdadera)
Inicio
    instrucción A_1
        :
    instrucción A_N
Fin
De lo contrario
Inicio
    instrucción B_1
        :
    instrucción B_N
Fin



NOTA: Siempre hacia "adelante"

Ejemplo en Lenguaje C

```
If (edad >= 18)
{
    printf("Hola");
    printf("Eres mayor");
    mayor=1;
}
else
{
    printf("Hola");
    printf("Eres puberto")
    mayor=0;
}
```


M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura selectiva en lenguaje C

switch (conmutación)

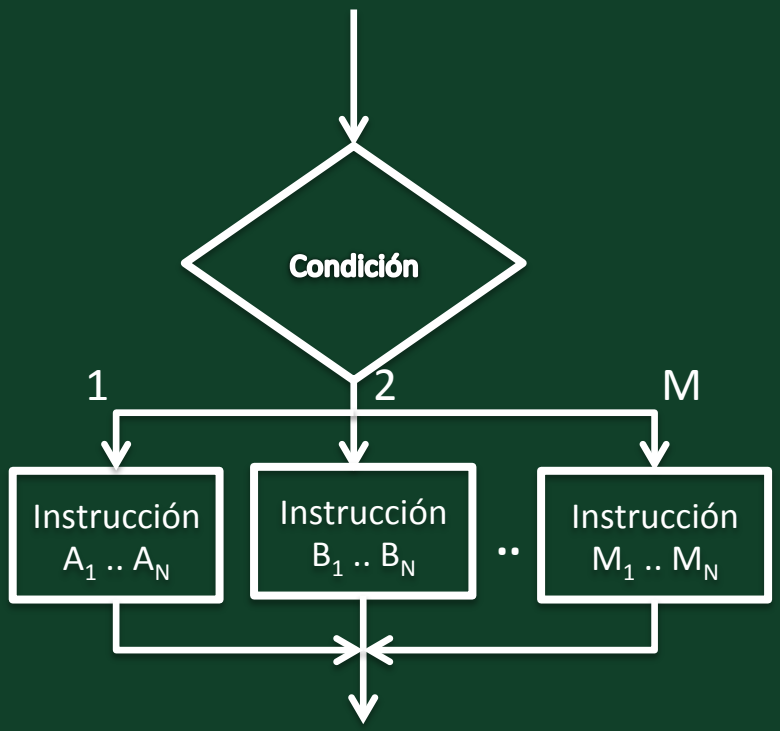
Las estructuras condicionales múltiples son tomas de decisión especializadas que permiten comparar una variable contra distintos posibles resultados, ejecutando para cada uno una serie de instrucciones.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura selectiva en lenguaje C: switch

Pseudocódigo

Si el valor es igual a
Valor 1: Inicio
instrucción A_1
:
instrucción A_N
Fin 1
Valor 2: Inicio
instrucción B_1
:
instrucción B_N
Fin 2
:
Valor M: Inicio
instrucción M_1
:
instrucción M_N
Fin M



NOTA: Siempre hacia "adelante"

Ejemplo en Lenguaje C

```
switch(edad)
{
  case 1:
    printf("Hola");
    printf("Tienes 1 año");
  break;
  case 2:
    printf("Hola");
    printf("Tienes 2 años");
  break;
      :
  default:
    printf("No has nacido");
  break;
}
```


M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura iterativa

Tiene las siguientes características

- Permiten ejecutar varias veces una instrucción
- Usan una condición para determinar el número de iteraciones

Notas:

Esto le brinda al programador la facilidad para realizar cálculos repetitivos sin necesidad de escribir muchas veces las mismas instrucciones

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura iterativa en lenguaje C

Las estructuras iterativas en este lenguaje de programación son:

- while (mientras)
- do-while (hacer - mientras)
- for (para)

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura iterativa en lenguaje C

while (mientras)

Esta estructura repite una instrucción o un conjunto de instrucciones siempre que la condición sea verdadera o se cumpla, cuando no se cumple o se vuelve falsa el ciclo (bucle) no se realiza o termina.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura iterativa en lenguaje C

while (mientras)

IMPORTANTE

Siempre se debe de inicializar la variable o variables que se utilizarán en la condición

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura iterativa en lenguaje C

while (mientras)

IMPORTANTE

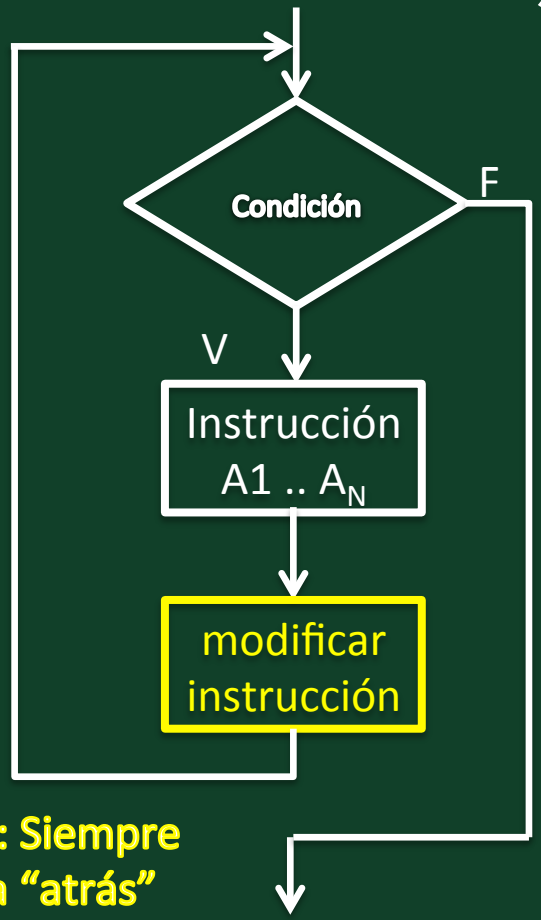
Siempre debe exitir una instrucción o función que modifique el estado de la condición, ya que si no se modifica puede que el ciclo nunca termine (bucle infinito).

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura iterativa en lenguaje C: while

Pseudocódigo

Inicializar variable
Mientras condición
Verdadera
Inicio
  instrucción A_1
  :
  instrucción A_N
  modificar condición
Fin
Falsa continua el flujo



NOTA: Siempre hacia "atrás"

Ejemplo en Lenguaje C

```
numero=1;

while (numero<=10)
{
   printf("%d", numero);
   numero=numero+1;
}


printf("Sigue el flujo");
```


M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Ejemplos

- Desplegar en pantalla tu nombre(s) y apellidos, guardar el programa como nombre.c
- Desplegar en pantalla los números de 1 al 10 con un salto de línea entre ellos, guardar el programa con el nombre numeros.c

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura iterativa en lenguaje C

do - while (hacer - mientras)

Esta estructura realiza una instrucción o un conjunto de instrucciones, verifica si la condición es verdadera o se cumple, cuando no se cumple o se vuelve falsa el ciclo (bucle) termina.

Es decir por lo menos hace una vez la instrucción o instrucciones.

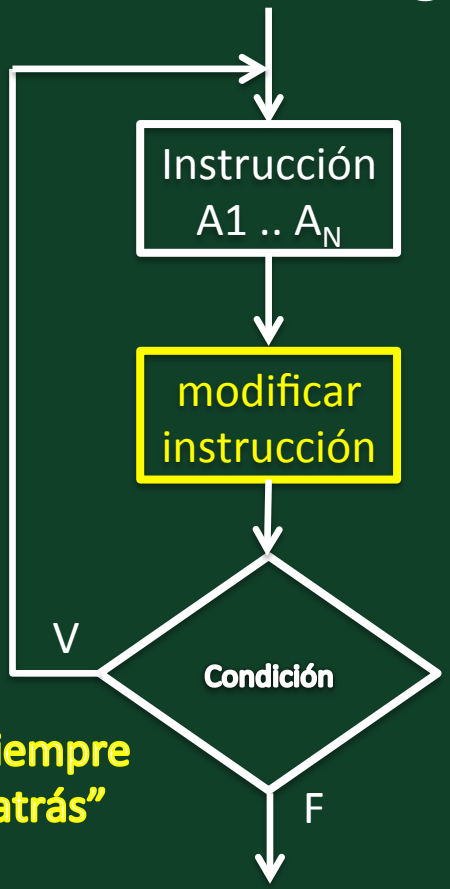
M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura iterativa en lenguaje C: do - while

Pseudocódigo

Inicializar variable
Hacer
Inicio
  instrucción A_1
  :
  instrucción A_N
  modificar condición
Fin
Mientras condición
Verdadera ir a Inicio

Falsa continua el flujo



NOTA: Siempre hacia "atrás"

Ejemplo en Lenguaje C

```
numero=1;
do
{
   printf("%d", numero);
   numero=numero+1;
} while (numero<=10);


printf("Sigue el flujo");
```


M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura iterativa en lenguaje C

for (para)

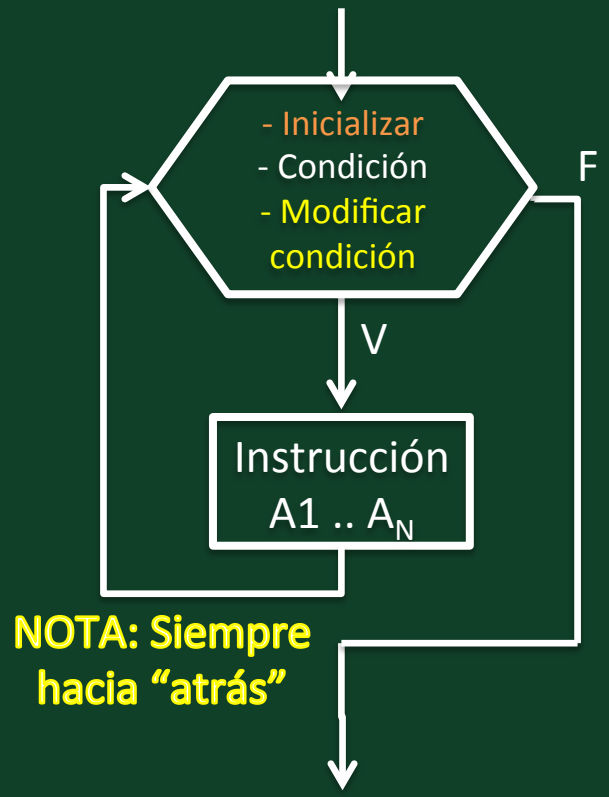
Esta estructura repite una instrucción o un conjunto de instrucciones siempre que la condición sea verdadera o se cumpla, cuando no se cumple o se vuelve falsa el ciclo (bucle) no se realiza o termina.

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Estuctura iterativa en lenguaje C: for

Pseudocódigo

Para condición
Verdadera,
modificar condición
Inicio
  instrucción A_1
  :
  instrucción A_N
Fin
Falsa continua el
flujo



NOTA: Siempre hacia "atrás"

Ejemplo en Lenguaje C

```
for(num=1;num<=10;num++)
{
   printf("%d", num);
}

printf("Sigue el flujo");
```


M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Referencias bibliográficas

Bibliográfia básica

[1]Luis Joyanes Aguilar, Ignacio Zahonero Martinez, “Programación en C, metodología, estructuras de datos y objetos”, Segunda edición, McGraw-Hill Interamericana, 2006, QA76.73.C153 J68 2006

[2]Osvaldo Cairó, “Fundamentos de Programación. Piensa en C”, Pearson Educación, 2006, QA76.73.C15 C357 2006

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

Referencias bibliográficas

Bibliográfia complementaria

[3] García, B., & Giner, J. R.,“Programación Estructurada en C”, Pearson Educación, 2008, QA76.73.C15 G34 2008.

[4] Silvia Edith Albarran Trujillo, Mireya Salgado Gallegos, “Programación Estructurada”, UAEM, 2008, QA76.6 A43

[5] Herbert Schildt,” C++ Soluciones de programación ”, McGraw-Hill/Interamericana, 2009, QA76.73.C153 S325 2009

M.T.I. Carlos Alberto Rojas Hernández
carojash@uaemex.mx

HUMANISMO
QUE TRANSFORMA

www.uaemex.mx