

ESCUELA SUPERIOR POLITECNICA
DE CHIMBORAZO

Facultad de Informática y Electrónica

Escuela de Ingeniería en Sistemas

Fundamentos de Programación

Fundamentos de Programación

Capitulo III

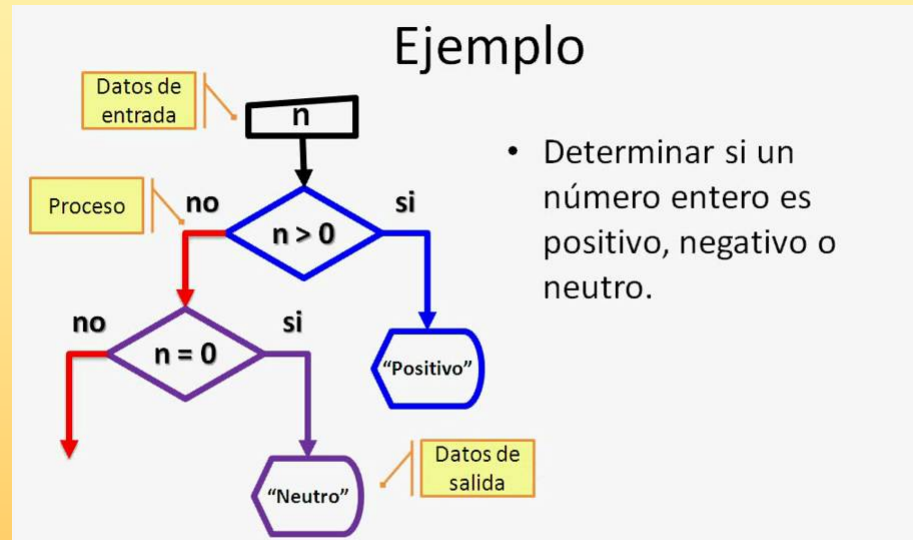
Estructuras de Programación

Propósitos y Objetivos

- Identificar las estructuras de los algoritmos
- Saber cuando a utilizar las estructuras de programación en determinados casos.
- Aplicar las estructuras de programación en casos prácticos.

Estructuras de programación

- Un problema se puede dividir en acciones elementales o instrucciones, usando un número limitado de estructuras de control (básicas) y sus combinaciones que pueden servir para resolver dicho problema.



Reglas para usar programación estructural

- Siempre un programa debe tender a ser general y considerar todos los posibles casos que se pueden presentar.
- Todas las estructuras deben poseer una sola entrada y una sola salida.
- Las estructuras son independientes entre sí indicando que cada estructura puede trabajar con otros datos.
- Entre todas las estructuras se pueden combinar, o una estructura puede contener a otra, o también una estructura puede estar a continuación de otra.

Clasificación de las estructuras de programación

CLASIFICACION ESTRUCTURAS

**ESTRUCTURA
SECUENCIAL**

**ESTRUCTURA DE
BIFULCACION O
CONDICION**

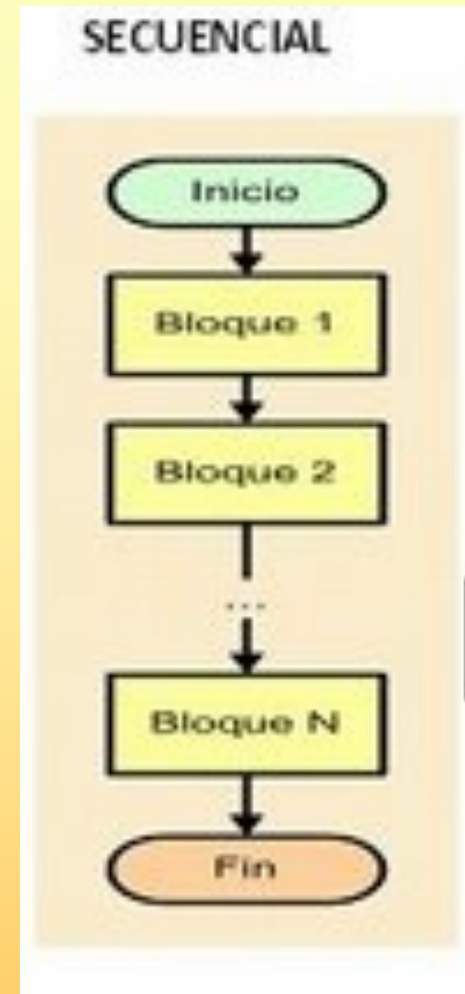
**ESTRUCTURA
ITERACION O
REPETICION**

**B. SIMPLE
B. ANIDADA
B. COMPUESTA
B. MULTIPLE**

**EST. WHILE<CONDICION>DO
EST. REPEAT<CONDICIO>UNTIL
EST. FOT<VI>TO<VF>DO**

Estructura secuencial

- Se caracteriza porque una acción se ejecuta detrás de otra. El flujo del programa coincide con el orden físico en el que se han ido poniendo las instrucciones. Dentro de este tipo podemos encontrar operaciones de inicio/fin, inicialización de variables, operaciones de asignación, cálculo, sumarización, etc.



Estructura secuencial - Ejemplo

Algoritmo Secuencial

{Algoritmo con estructura secuencial}

variables enteras A, B, S, R, P

leer(A, B)

$S \leftarrow A + B$

$R \leftarrow A - B$

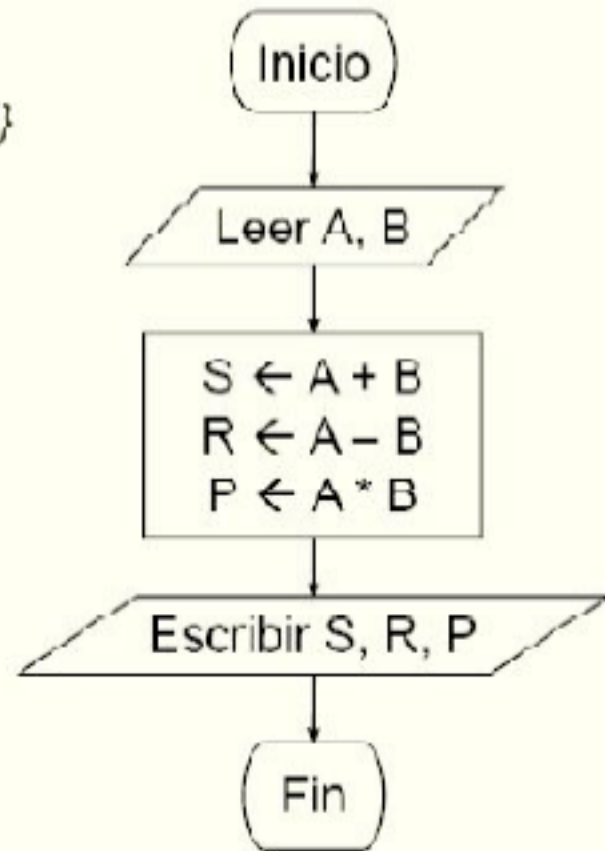
$P \leftarrow A * B$

escribir('Suma = ', S)

escribir('Resta = ', R)

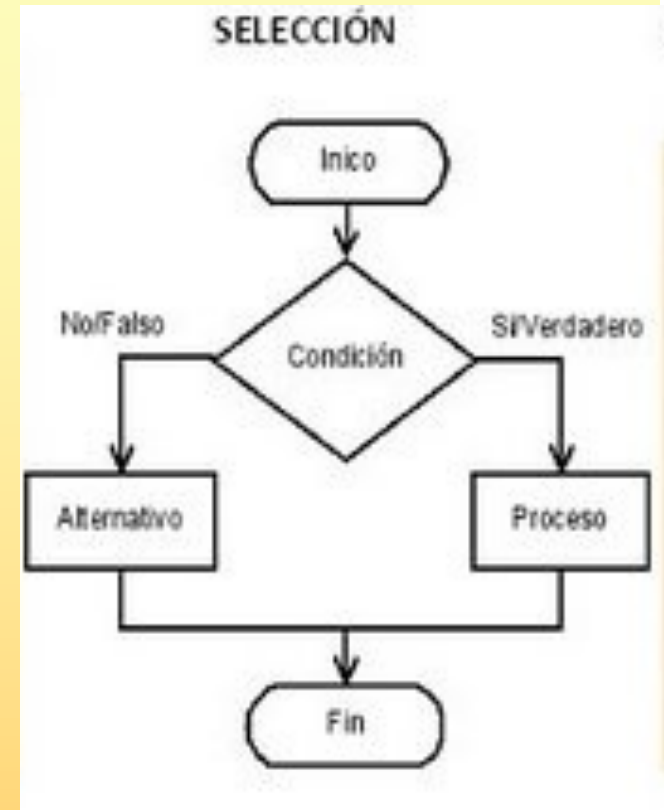
escribir('Producto = ', P)

finalgoritmo



Estructura condicional

- Estas estructuras se identifican porque en la fase de solución del problema existe algún punto en el cual es necesario establecer una pregunta, para decidir si ciertas acciones deben realizarse o no.
- Se evalúa una expresión y, dependiendo del resultado, se decide la siguiente sentencia a ejecutar.



Estructura condicional

- **Simples.-** Lo que se hace es EVALUAR la condición, si la condición es verdadera realiza la acción, en caso contrario termina el programa.

Si <condición> entonces

Acción(es)

Fin-si

Estructura condicional

- **Simples.- Ejemplo**

Pseudocódigo

Inicio

Leer COMPRA

Si $COMPRA > 1000$ entonces

$DESCUENTO = COMPRA * 0.10$

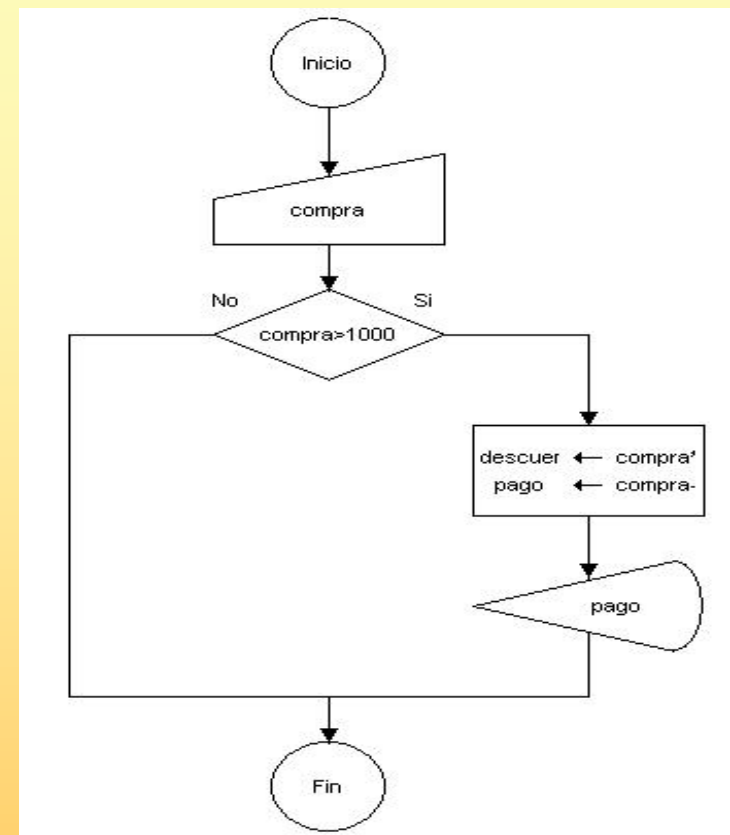
$PAGO = COMPRA - DESCUENTO$

Escribir PAGO

Finsi

Fin

DFD



Estructura condicional

- **Doble.-** Luego de evaluar una condición si esta se cumple, es decir si es verdadera realiza una serie de acciones, y si esta es falsa se realiza otra serie de acciones distinta a la primera.

Si <condición> entonces

Acción(es)

Sino

Acción(es)

Fin si

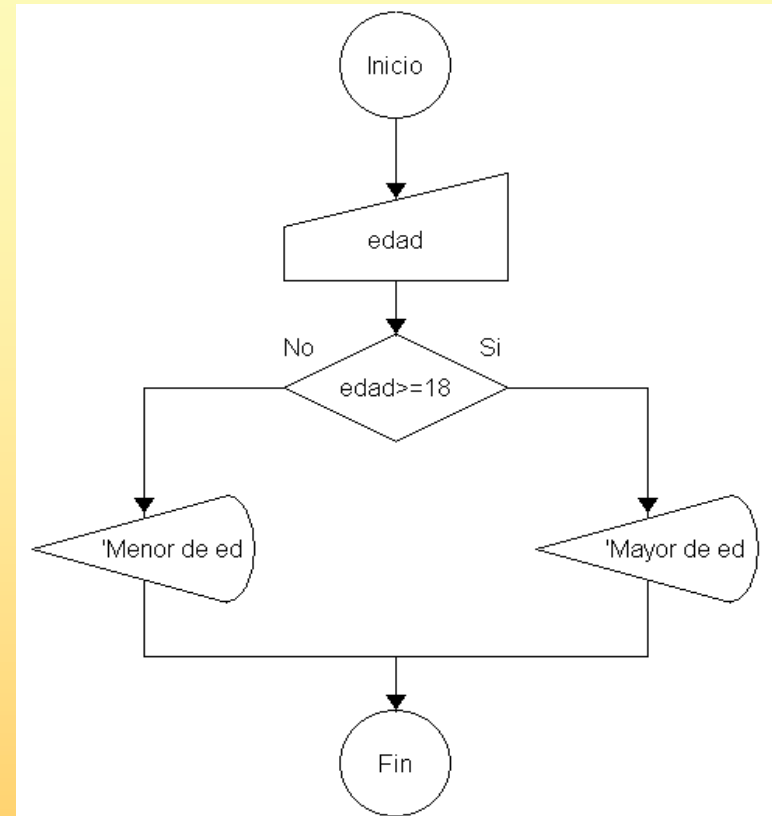
Estructura condicional

- **Doble.- Ejemplo**

Pseudocódigo

```
Inicio  
  Leer EDAD  
  Si EDAD >= 18 entonces  
    Escribir “Mayor de edad”  
  Sino  
    Escribir “Menor de edad”  
  Finsi  
Fin
```

DFD



Estructura condicional

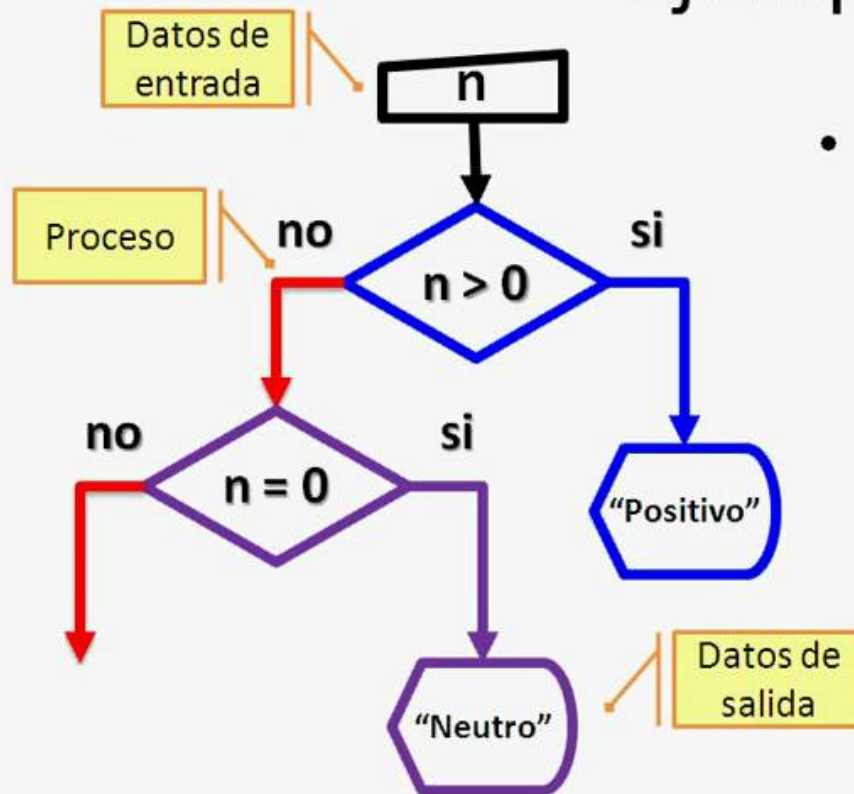
- **Anidada.-** Luego de evaluar una condición si esta se cumple, es decir si es verdadera evalúa otra condición si de forma anidada y ejecutará una serie de acciones.

```
Si <condición> entonces
    Si <condición> entonces
        Acción(es)
    Fin si
Sino
    Si <condición> entonces
        Acción(es)
    Fin si
Fin si
```

Estructura condicional

- Anidada.- Ejemplo

Ejemplo



- Determinar si un número entero es positivo, negativo o neutro.

Estructura condicional

- **Compuesta.-** Luego de evaluar dos o mas condiciones unidas por un operador lógico si esta se cumple, es decir si es verdadera realiza una serie de acciones, y si esta es falsa se realiza otra serie de acciones distinta a la primera.

```
Si <condición> AND <condición> entonces
    Acción(es)
Sino
    Acción(es)
Fin si
```


Estructura condicional

- **Compuesta.- Ejemplo**

Inicio

Leer num1

Leer num2

Leer num3

Si $(num1 > num2)$ Y $(num1 > num3)$ entonces

 Escribir “Mayor es” num1

Sino

 Si $(num2 > num3)$ entonces

 Escribir “Mayor es” num2

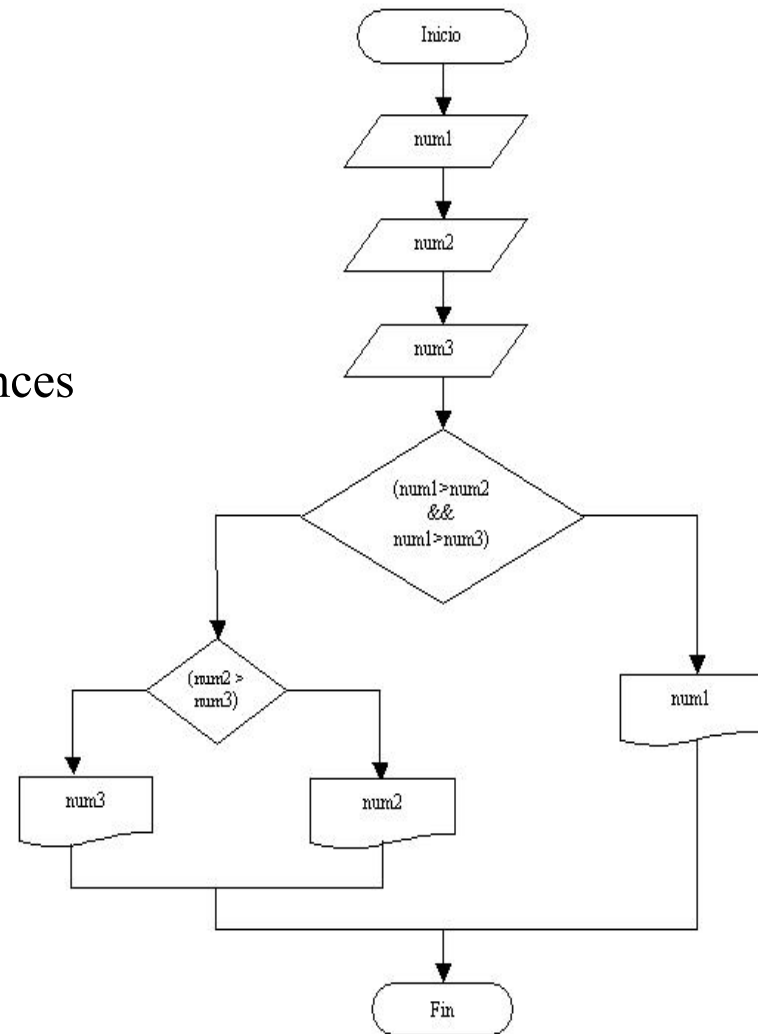
 Si no

 Escribir “Mayor es” num3

 Fin si

Fin si

Fin



Estructura condicional

- **Múltiple.-** Se realiza a partir de anidar estructuras simples y/o dobles, de manera tal que se realicen diferentes acciones con base a varias comparaciones, así habrá tantas opciones como se requieran.

Según <opción> hacer

Opcion1:

Acción(es)

Opcion2:

Acción(es)

Opcion3:

Acción(es)

De otro modo:

Acción(es)

Fin Según

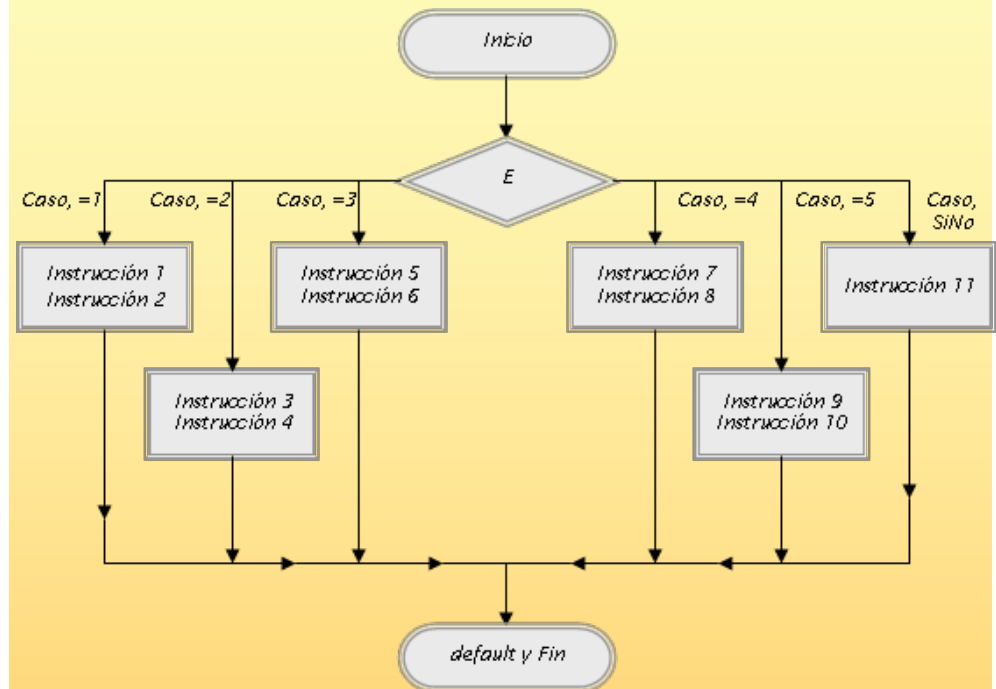
Estructura condicional

- Múltiple.- Ejemplo

Código en C++

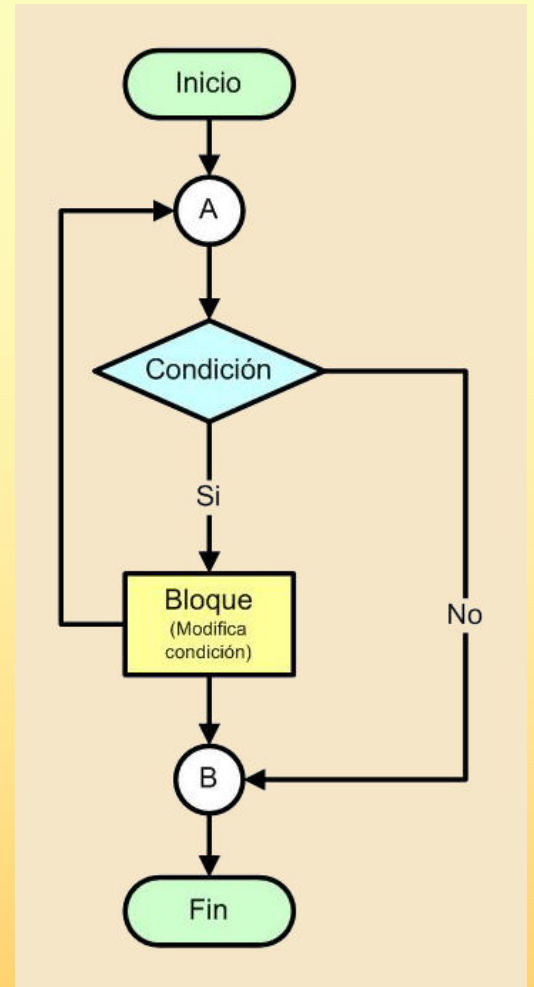
```
void main()
{
    int roll = 3 ;
    switch ( roll )
    {
        case 1 :
            printf("I am Pankaj");
            break;
        case 2 :
            printf("I am Nikhil");
            break;
        case 3 :
            printf("I am John");
            break;
        default :
            printf("No student found");
            break;
    }
}
```

DFD



Estructuras de repetición

- Este tipo de estructura se utilizan para ejecutar acciones repetidamente, esto se hace posible mediante una secuencia de instrucciones que se repiten una y otra vez y así evitamos escribir múltiples veces las mismas instrucciones.



Estructuras de repetición

- **Mientras (While do).**- Permite realizar varias veces un mismo proceso considerando siempre el estado que presenta la condición, la cual es quien controla el proceso. Se la debe utilizar cuando NO sabemos el número de veces que se ha de repetir un ciclo.

Mientras <condición> hacer

Accion1

Accion2

.

.

AccionN

Fin-mientras

Estructuras de repetición

- **Mientras (While do).- Características**
 - Se ejecuta primero la condición y luego el proceso.
 - Se repite el proceso si solo si cumple la condición.
 - Por lo general si el contador empieza en 1 la condición será \leq y si el contador empieza en 0 la condición será $<$
 - El número mínimo de repeticiones del proceso es 0 veces y el máximo es N veces.

Estructuras de repetición

- Mientras (While do).- Ejemplo.**

Proceso sin_titulo

Escribir "Hay alumno";

Leer ALUM;

Mientras ALUM="s" Hacer

Leer CALIF1,CALIF2;

PROM \leftarrow (CALIF1+CALIF2)/2;

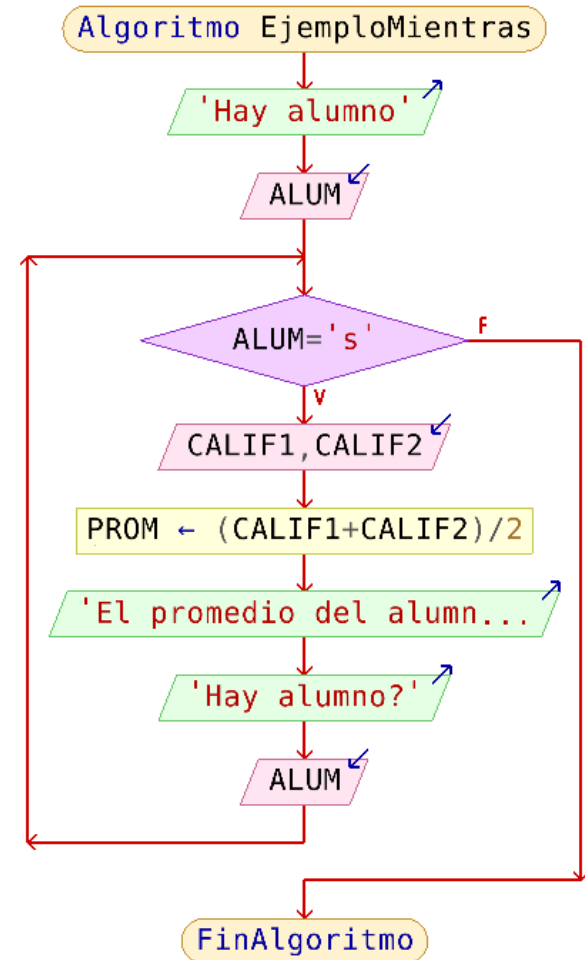
Escribir "El promedio del alumno es ",PROM

Escribir "Hay alumno";

Leer ALUM;

FinMientras

FinProceso



Estructuras de repetición

- **Para (For).**- Permite ejecutar los pasos de la solución del algoritmo un número definido de veces y de modo automático controla el número de iteraciones o pasos a través del cuerpo del ciclo. utiliza un contador en el cual se va acumulando el número de veces que se ha repetido las instrucciones.

Para VC=LI Hasta LS Hacer

Accion1

Accion2

.

AccionN

Fin-para

V.C Variable de control de ciclo

L.I Límite inferior

L.S Límite superior

Estructuras de repetición

- **Para (For).- Características**
 - El valor del incremento puede ser solamente $+n$ o $-n$.
 - Si $VI < VF$ entonces se supone la condición \leq y el incremento de $+1$.
 - Si $VI > VF$ entonces se supone la condición es \geq y el incremento de -1 .
 - El número mínimo de veces que se repite un proceso es 0 veces, mientras que el número máximo N veces.

Estructuras de repetición

- Para (For).- Ejemplo.

Proceso pruebaFor

Para datos<-1 Hasta 5 Con Paso 1 Hacer

Leer num1,num2;

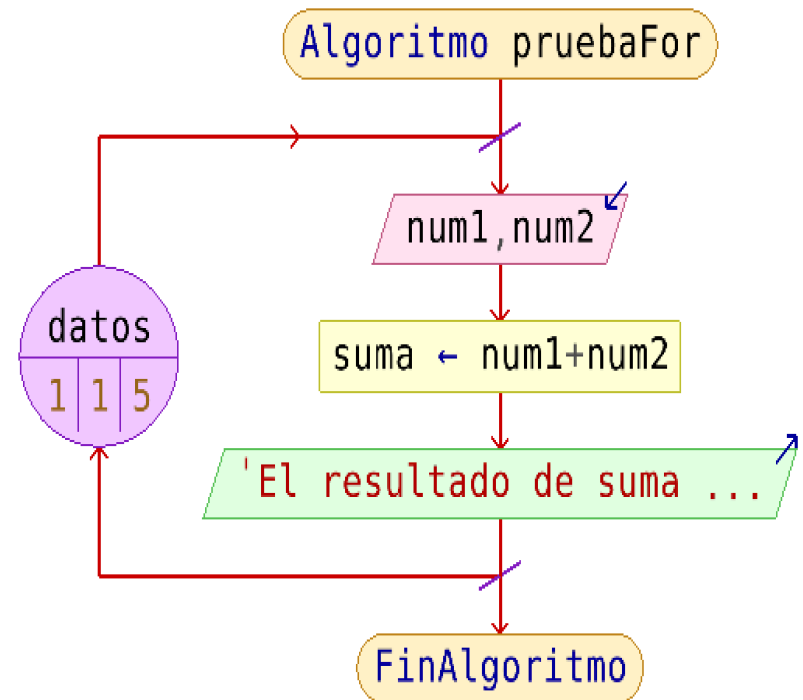
Suma<-num1+num2;

Escribir "el resultado de sumar

",num1," + ",num2," = ",suma;

FinPara

FinProceso



Estructuras de repetición

- **Repetir (Repeat.....until).**- Permite repetir hasta que la condición se cumpla, a diferencia de las estructuras anteriores la condición se escribe al finalizar la estructura.

Repetir

Accion1

Accion2

.

.

AccionN

Hasta que <condición>

Estructuras de repetición

- **Repetir (Repeat...until).**- Características
 - Se ejecuta primero el proceso y luego la condición.
 - Se repite el proceso hasta que la condición sea verdadera o dicho en otras palabras mientras la condición sea falso.
 - Si el contador = 1 la condición podría ser >
 - Si el contador = 0 la condición podría ser >=
 - El número mínimo de veces que se repite el proceso es 1 vez y el número máximo N veces.

Estructuras de repetición

- Repetir (Repeat...until).- Ejemplo.

Proceso sin_titulo

Repetir

Leer Salario;

salarioFin \leftarrow Salario*1.15;

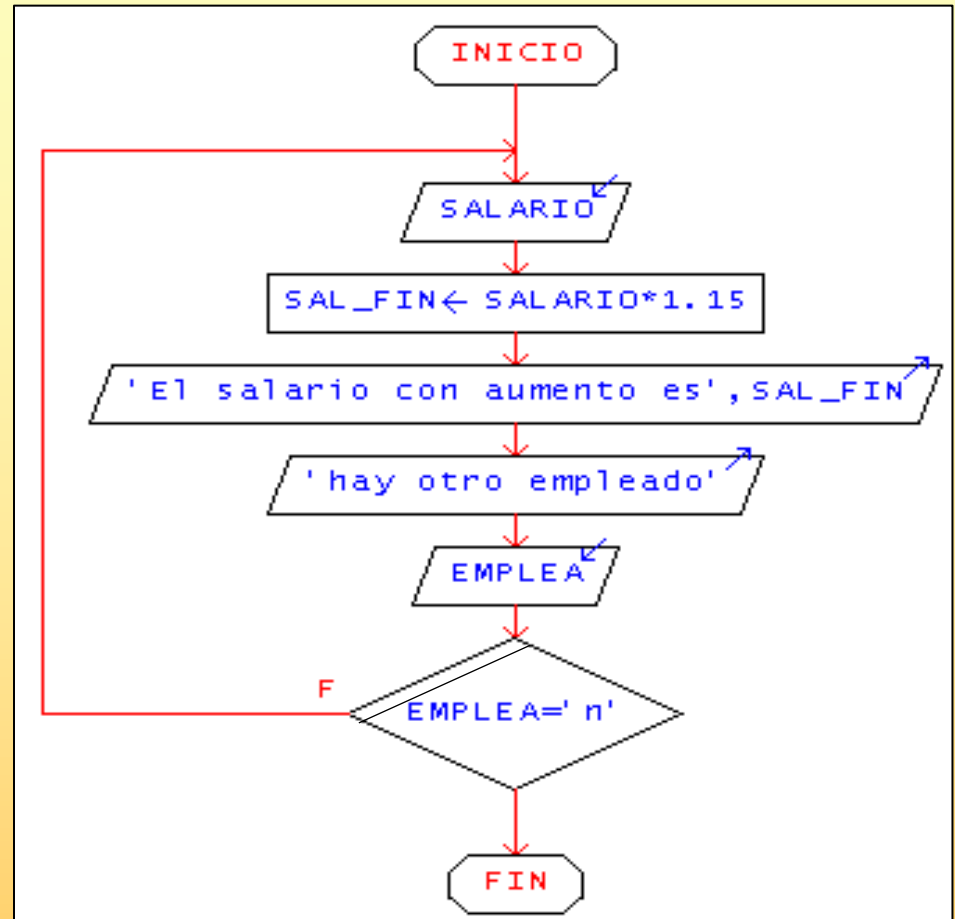
Escribir “El salario con aumento es”,salarioFin;

Escribir ”Hay otro empleado”;

Leer EMPLEA;

Hasta Que EMPLEA=“n”

FinProceso



Gracias
¿Alguna Pregunta?