



**Clase presencial**



## **Sesion3A: Estructuras de Control**

Instructor: David Paúl Porras Córdova

@iscodem



# Objetivo General

- ❑ Identificar las diferentes estructuras de control utilizados en programación.
- ❑ Diferenciar en el uso de las estructuras de control en determinado contexto de aplicación.
- ❑ Conocer el ciclo de vida de las estructuras de control.
- ❑ Mejores prácticas para el uso de estructuras de control.



# Contenido de Agenda

- Estructuras secuenciales
- Tipo de estructuras de control.
  - ▣ Condicionales
    - Simples (IF -ELSE)
    - Compuestas (IF ELSE IF)
  - ▣ Bucles
    - While
    - Do While
    - For
  - ▣ Contadores y acumuladores
- Buenas prácticas de programación

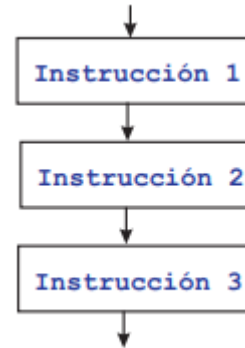


# 1. Estructura secuencial

Es aquella en la que una acción (instrucción) sigue a la otra en el orden en el que están escritas.

Su representación y el diagrama de flujo se muestra a continuación y son ejemplos de algoritmos secuenciales.

```
...  
Instrucción 1  
Instrucción 2  
Instrucción 3  
...
```



## 2. Estructuras de control

Son parte fundamental de cualquier lenguaje. Sin ellas, las instrucciones de un programa sólo podrían ejecutarse en el orden en que están escritas (orden secuencial). Las estructuras de control permiten modificar este orden. Hay dos categorías de estructuras de control.

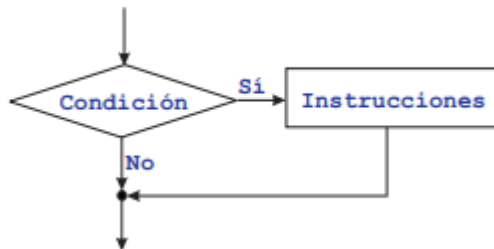
**2.1. Condicionales o bifurcaciones**

**2.2. Bucles o repeticiones**

## 2.1. Estructuras condicionales

### Estructura condicional simple: IF

```
...  
if condición  
    instrucciones  
end  
...
```



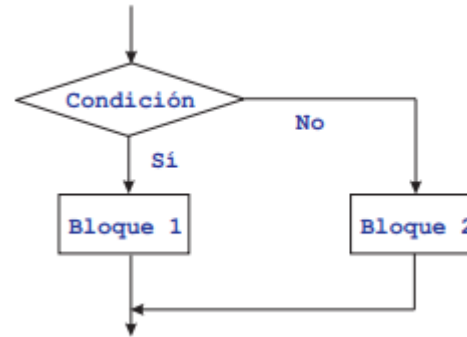
Cálculo del valor de la función  $f(x) = 0$  si  $x \leq 0$ ,  $f(x) = x^2$  si  $x > 0$ .

```
Inicio  
1- LEER x  
2- HACER f=0  
3- Si x>0  
    HACER f=x2  
  Fin Si  
4- IMPRIMIR 'El valor de la funcion es: ', f  
Fin
```

## 2.1. Estructuras condicionales

Estructura condicional doble: IF - ELSE

```
...  
if condición  
    bloque-1  
else  
    bloque-2  
end  
...
```



## 2.1. Estructuras condicionales

Como ejemplo de utilización de este tipo de estructuras se plantea el problema de calcular las raíces de una ecuación de segundo grado

$$ax^2 + bx + c = 0$$

```
Inicio
  1- LEER A,B y C
  2- CALCULAR D=B2-4*A*C
  3- CALCULAR AA=2*A
  4- Si D ≥ 0
      CALCULAR DD=√D
          x1=(-B+DD)/AA
          x2=(-B-DD)/AA
      IMPRIMIR 'La ecuación tiene raíces reales:', x1, x2
  Si no
      CALCULAR DD=√-D
          Re=-B/AA
          Im=DD/A2
      IMPRIMIR 'La ecuación tiene raíces complejas conjugadas:'
      IMPRIMIR 'Parte real:', Re
      IMPRIMIR 'Parte imaginaria:', Im
  Fin Si
Fin
```



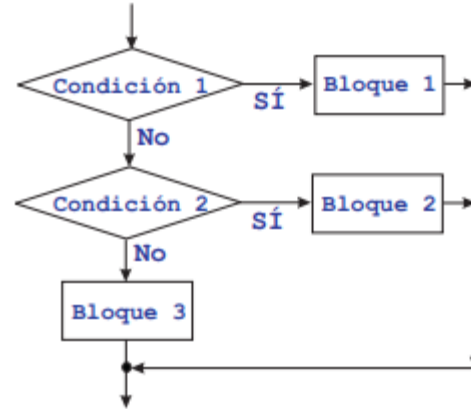
# Laboratorio: 3.1. Estructuras condicionales simples

- Ejercicio 1:
  - ▣ Tiempo: 10 minutos
  - ▣ Pedir el radio de un círculo y calcular su área.  $A=PI*radio^2$ .
  - ▣ Pedir dos números y decir cual es el mayor.
  - ▣ Leer un número e indicar si es positivo o negativo. El proceso se repetirá hasta que se introduzca un 0.
  - ▣ Pedir un número entre 0 y 999, decir si es capicúa.

## 2.1. Estructuras condicionales

Estructura condicional múltiple: IF - ELSEIF - ELSE

```
...  
if condición-1  
    bloque-1  
elseif condición-2  
    bloque-2  
else  
    bloque-3  
end  
...
```



# Laboratorio: 3.1. Estructuras condicionales dobles

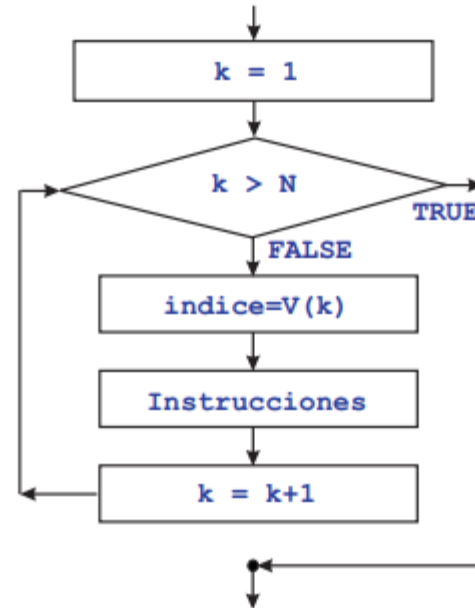
- Ejercicio 3:
  - ▣ Tiempo: 20 minutos
  - ▣ Ingresar por teclado 3 notas de un alumno, calcular el promedio y mostrar en pantalla la situación del estudiante.
    - <10 DESAPROBADO
    - [10 - 15] REGULAR
    - [15 - 20] EXCELENTE
  - ▣ Pedir dos números y decir cual es el mayor.
  - ▣ Pedir tres números y mostrarlos ordenados de mayor a menor.

## 2. Bucles – ciclos - iteraciones

### Estructura de repetición indexada: FOR

```
...  
for indice=V  
    instrucciones  
end  
...
```

- El valor de la variable de control índice puede ser utilizado o no dentro del conjunto de instrucciones que forman parte del cuerpo del FOR, pero no debe ser modificado.
- El conjunto de valores que debe recorrer el índice puede ser vacío ( $N=0$ ). En ese caso, el bloque de instrucciones no se ejecuta ninguna vez.
- Las estructuras FOR e IF pueden “anidarse”, es decir, incluir una dentro de la otra, con la restricción (de sentido común) de que la interior tiene que estar completamente contenida en uno de los bloques de instrucciones de la otra.



## 2. Bucles – ciclos - iteraciones

Estructura de repetición indexada: FOR

Dado un entero, n, calcular

$$\sum_{k=0}^n \left(\frac{1}{2}\right)^k = 1 + \frac{1}{2} + \frac{1}{2^2} + \cdots + \frac{1}{2^n}$$

Inicio

LEER n

HACER suma=1

HACER ter=1

**Para** k= 1, 2, ..., n

HACER ter=ter/2

HACER suma=suma+ter

**Fin Para**

IMPRIMIR 'La suma vale : ', suma

Fin

## Laboratorio: 3.1. Bucle FOR

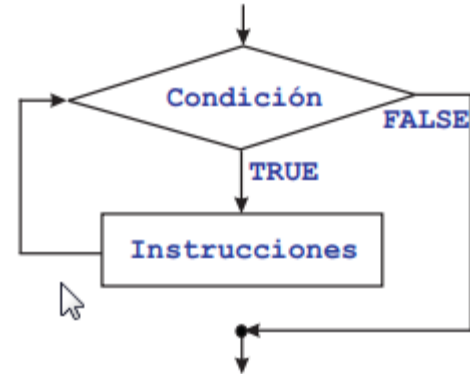
- Ejercicio 4:
  - ▣ Tiempo: 20 minutos
  - ▣ Dado un entero  $n$ , calcular la suma de los  $n$  primeros números impares.
  - ▣ Pedir 15 números y escribir la suma total.
  - ▣ Pedir 10 sueldos. Mostrar su suma y cuantos hay mayores de 1000 soles.

## 2. Bucles – ciclos - iteraciones

### Estructura repetitiva condicional: WHILE

```
...  
while expresión-lógica  
    instrucciones  
end  
...
```

- Al comienzo de cada iteración se evalúa la expresión-lógica.
- Si el resultado es VERDADERO, se ejecuta el conjunto de instrucciones y se vuelve a iterar, es decir,
  - se repite el paso 1.
- Si el resultado es FALSO, se detiene la ejecución del ciclo WHILE y el programa se sigue ejecutando por la instrucción siguiente al END.



## Laboratorio: 3.1. Bucle WHILE

- Ejercicio 4:
  - ▣ Tiempo: 20 minutos
  - ▣ Pedir números hasta que se introduzca uno negativo, y calcular la media.
  - ▣ Pedir 15 números y escribir la suma total.



## Laboratorio: 3.1. Estructuras condicionales dobles

- Ejercicio 5:
  - ▣ Tiempo: 20 minutos
  - ▣ Una empresa requiere un sistema que calcula el total de utilidades que recibirá un trabajador por los años de servicio, considerar lo siguiente por los años de antigüedad.

[ < 1 año ]	5% del salario
[ 1 – 2 años ]	7% del salario
[ 2 – 5 años ]	10% del salario
[ 5 – 10 años ]	15% del salario
[ > 10 años ]	20% del salario

## 2.2.1. Acumuladores y Contadores

- **Acumuladores:** Variables que acumulan cantidades en cada iteración de un bucle.

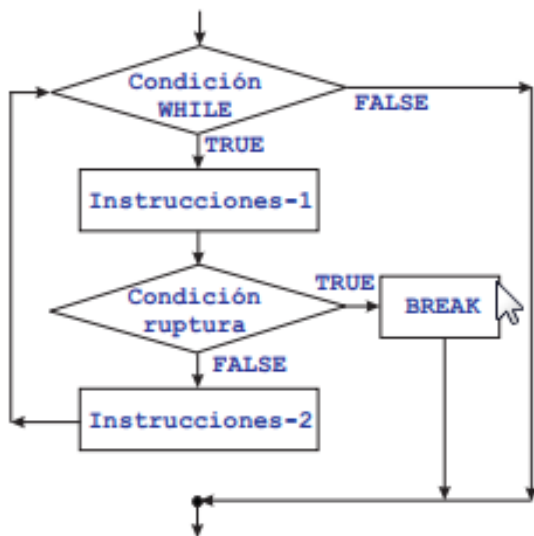
```
num+=1  num-=1  num=num+1
```

- **Contadores:** Secuencias de control y seguimiento para el manejo de iteraciones en un bucle.

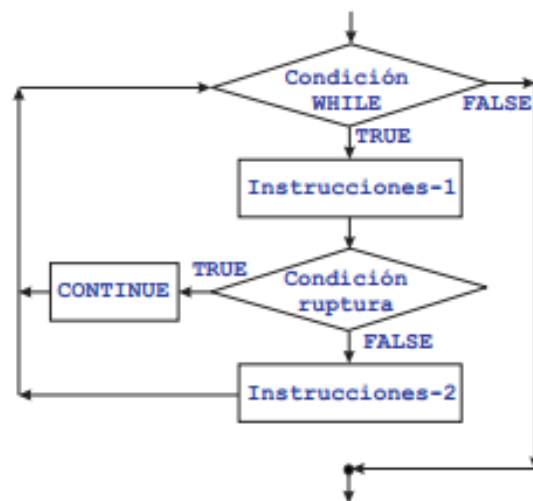
```
num+=valor  num-=i  num=num+valor
```

## 2.3. Ruptura de iteraciones en bucles

Ruptura de ciclos de repetición: BREAK y CONTINUE



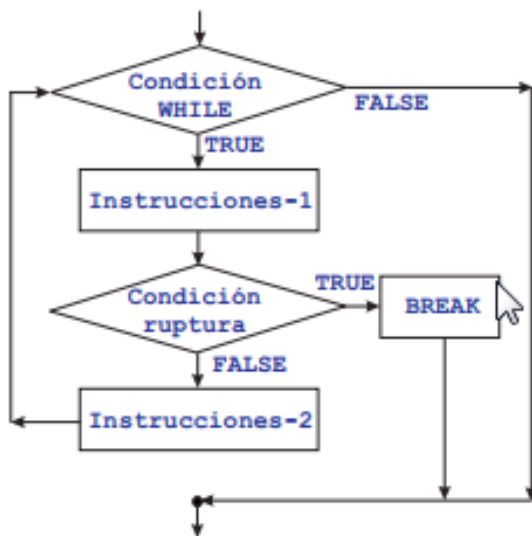
Termina el bucle



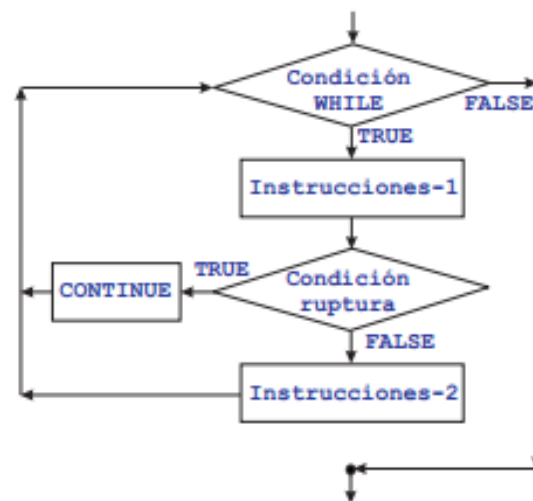
Termina la iteración

## 2.3. Ruptura de iteraciones en bucles

Ruptura de ciclos de repetición: BREAK y CONTINUE



Termina el bucle

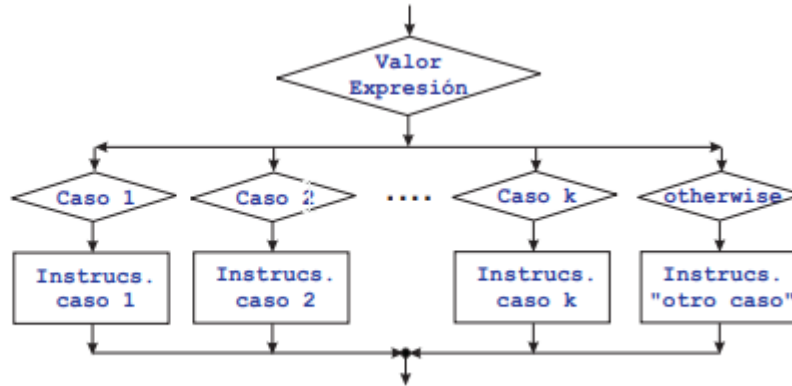


Termina la iteración

## 2.3. Estructura de elección entre varios switch

### Estructura de elección entre varios casos: SWITCH

```
switch expresión
  case valor-1
    instrucciones caso 1
  case valor-2
    instrucciones caso 2
  ...
  case {valores...}
    instrucciones caso k
  otherwise
    instrucciones
end
```



- Al comienzo se evalúa la expresión.
- Si expresión toma el valor (´o valores) especificados junto a la primera cláusula CASE, se ejecuta el conjunto de instrucciones de este caso y después se abandona la estructura SWITCH, continuando por la instrucción siguiente al END.
- Se repite el procedimiento anterior, de forma ordenada, para cada una de las cláusulas CASE que siguen.
- Si la cláusula OTHERWISE está presente y la expresión no ha tomado ninguno de los valores anteriormente especificados, se ejecuta el conjunto de instrucciones correspondiente.

## Laboratorio: 3.1. Bucle WHILE

- Ejercicio 6:
  - ▣ Tiempo: 60 minutos
  - ▣ Una empresa que se dedica a la venta de desinfectantes necesita un programa para gestionar las facturas. En cada factura figura: el código del artículo, la cantidad vendida en litros y el precio por litro.

Se pide de 5 facturas introducidas: Facturación total, cantidad en litros vendidos del artículo 1 y cuantas facturas se emitieron de más de 600 soles.

# Laboratorio: 3.1. Bucle WHILE

- Ejercicio 7:
  - ▣ Tiempo: 60 minutos
  - ▣ Realice un programa que realice las funciones que a continuación se indican. Todas las opciones deben ser presentadas al usuario a través de un menú de opciones.
  - ▣ **Escriba una función** en la que se introduzcan 10 enteros y determine cuales de estos enteros son pares y cuales son impares. Un número entero es un “número perfecto” si sus factores, incluyendo al 1 (pero excluyendo en el número mismo), suman igual que el numero. Ejemplo: 6 es un numero perfecto porque  $6 = 1 + 2 + 3$ .
  - ▣ **Escriba una función** que regrese los primeros 100 números perfectos. Esta función debe tener una función anidada que determine al número perfecto.
  - ▣ **Escriba una función** que tome un valor entero de cuatro dígitos y regrese el número con los dígitos invertidos. Por ejemplo, dado el número 7631, la función deberá regresar 1367.