Material descargable

Fundamentos de la Programación

¿QUÉ ES UNA ESTRUCTURA DE CONTROL?

Las estructuras de control determinan el orden en que deben ejecutarse las instrucciones de un algoritmo. Esto significa que regulan si las instrucciones serán ejecutadas una tras otra (estructuras secuenciales), si se tomarán decisiones sobre la ejecución de alguna acción (estructuras selectivas o de decisión), o si se realizarán repeticiones (estructuras repetitivas).

Esto permite que ciertas instrucciones se ejecuten y otras se omitan según la evaluación de una condición.

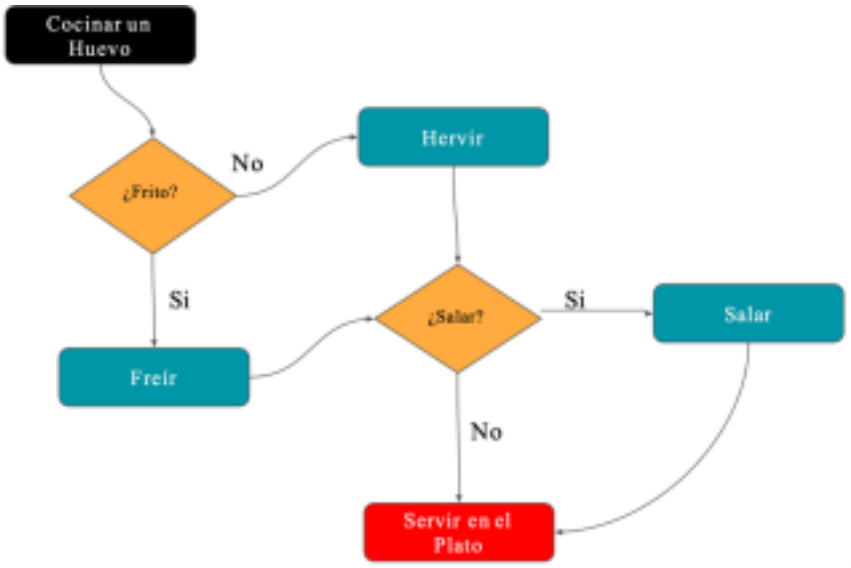
¿QUÉ ES UNA ESTRUCTURA SECUENCIAL?

Una estructura secuencial es aquella en la que una acción (instrucción) sigue a otra de manera secuencial. Las tareas se realizan de forma que la salida de una es la entrada de la siguiente, y así sucesivamente hasta completar todo el proceso. Esta estructura de control es la más simple y permite que las instrucciones se ejecuten una tras otra en el orden en que están listadas.

¿QUÉ ES UNA ESTRUCTURA SELECTIVA?

Las estructuras de control conocidas como "estructuras selectivas" o “condicionales” son vitales en el desarrollo de algoritmos que demandan más que una simple secuencia de instrucciones. **Son cruciales cuando te enfrentas a múltiples opciones que están condicionadas por una evaluación específica**. De este modo, te facilitan la toma de decisiones lógicas, razón por la cual también se les llama estructuras de decisión o selectivas.

En una estructura selectiva, evalúas una condición y, en función de su resultado, tomas una acción particular. Las condiciones se expresan mediante expresiones lógicas. Por ejemplo, podrías estar desarrollando un algoritmo para determinar si es necesario llevar un paraguas. La condición sería si está lloviendo, y la acción podría ser llevar un paraguas si la condición es verdadera, o no llevarlo si la condición es falsa.



¿QUÉ ES UNA CONDICIÓN?

En programación, una condición es toda sentencia que puede determinar su verdad (true) o falsedad (false). La mayoría de las veces, estas sentencias son comparaciones. Por ejemplo, "4 > 5" es una condición porque puede ser verdadera o falsa (en este caso es falsa porque 4 no es mayor que 5). Sin embargo, la sentencia "Escribir 'EggEducacion'" no es una condición, ya que no hay nada que comparar para determinar su veracidad.

Podemos utilizar como condiciones el valor de una variable lógica, ya que representa lo mismo: verdadero o falso. Por lo tanto, una condición sirve para

elegir entre una opción u otra, y normalmente se expresa con un "Si". Por ejemplo: "Si va a llover, lleva un paraguas".

Para establecer condiciones, necesitamos utilizar operadores.

¿QUÉ SON LOS OPERADORES?

Los operadores son símbolos especiales que utilizamos en programación para realizar operaciones sobre variables y valores. En el contexto de las estructuras de control, como las selectivas y otras, los operadores se utilizan para comparar valores y tomar decisiones basadas en esas comparaciones.

Los **operadores relacionale**s nos permiten comparar valores y determinar si son iguales, mayores, menores, o diferentes entre sí. Por ejemplo, los operadores ">", "<", "==", ">=", "<=", y "<>" son ejemplos de operadores relacionales que nos permiten realizar estas comparaciones.

Por otro lado, los **operadores lógicos** se utilizan para combinar o negar condiciones lógicas. Los operadores lógicos más comunes son "Y" (AND), "O" (OR) y "NO" (NOT). Estos operadores nos permiten evaluar múltiples condiciones al mismo tiempo y tomar decisiones más complejas en función de los resultados.

OPERADORES RELACIONALES

Los operadores relacionales son símbolos que se utilizan para comparar dos valores. Si el resultado de la comparación es correcto, la expresión se considera verdadera; de lo contrario, se considera falsa.

Operadores Relacionales	Significado	Ejemplo	Lectura	Resultado
>	Mayor que	3 > 2	¿ 3 es mayor que 2?	VERDADERO
<	Menor que	1 < 5	¿ 1 es menor que 5?	VERDADERO
==	Igual que	4 == 4	¿4 es igual a 4?	VERDADERO
>=	Mayor o igual que	4 >= 5	¿4 es mayor o igual que 5?	FALSO
<=	Menor o igual que	'A' <= 'B'	¿'A' es menor o igual que 'B?	VERDADERO

<>	**Distinto que**	**10 <> 8**	**¿10 distinto que 8?**	**VERDADERO**

OPERADORES LÓGICOS

Estos operadores se utilizan cuando necesitamos expresiones lógicas con múltiples variantes y proporcionan un resultado basado en si se cumple o no una determinada condición. Producen un resultado lógico y sus operadores también son valores lógicos o asimilables a ellos.

Operadores Relacionales	Significado	Ejemplo	Lectura	Resultado
Y	**Conjunción -** Devuelve un valor lógico verdadero si ambas expresiones son verdaderas. En caso contrario el resultado es falso.	**(2 < 4 Y 3 > 5)**	**2 es menor que 4 Y 2 es mayor que 5**	**FALSO**
O	**Disyunción -** Este operador devuelve verdadero si alguna de las expresiones es verdadera. En caso contrario devuelve "falso".	**(7 <= 8 O 10 >= 9)**	**7 es menor o igual que 8 O 10 es mayor o igual que 9**	**VERDADERO**
NO	**Negación - E**ste operador cambia la devolución de una expresión, al caso contrario. Si es verdadero lo hace falso y si es falso lo hace verdadero.	**no(1 == 1)**	**¿1 ES IGUAL A 1? VERDADERO**	**FALSO (ya que cambia el resultado del análisis)**

Al trabajar con operadores lógicos, para determinar si una expresión lógica devuelve Verdadero o Falso, debemos referirnos a la tabla de la verdad.

Conjunción

A	Operador	B	Resultado
V	Y	V	V
V	Y	F	F
F	Y	V	F
F	Y	F	F

Disyunción

A	Operador	B	Resultado
V	O	V	V
V	O	F	V
F	O	V	V
F	O	F	F

Negación

A	Resultado	B	Resultado
no(V)	F	no(F)	V

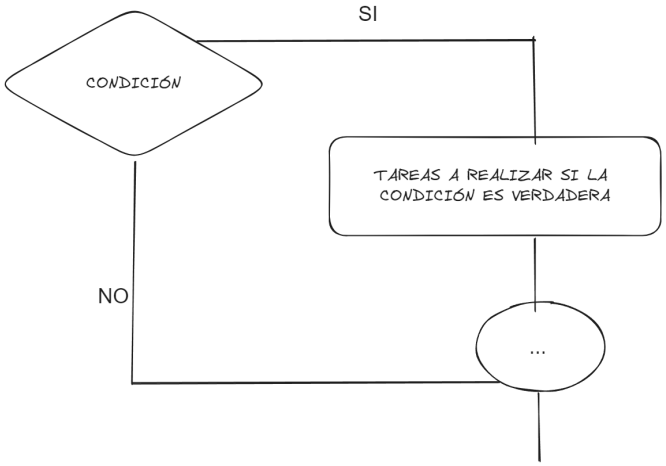
TIPOS DE ESTRUCTURAS CONDICIONALES

Es importante recordar que estas estructuras se emplean para la toma de decisiones lógicas, permitiendo elegir entre diversas alternativas de acción. Existen tres tipos principales de estructuras selectivas o condicionales:

- **Simples**: utiliza la instrucción "Si".
- **Dobles**: emplea las instrucciones "Si" y "Sino".
- **Múltiples**: se desarrolla mediante instrucciones "Según" o "Si" anidado.

CONDICIONAL SIMPLE

Las estructuras condicionales simples realizan una acción (o conjunto de ellas) únicamente cuando **la expresión a evaluar resulta en un resultado positivo, es decir, verdadero**.




```
Si expresion_logica Entonces
    acciones_por_verdadero
Fin Si
```

Ejemplo:

```
Algoritmo CondicionSimple
    Definir edad Como Entero

    Escribir "Ingrese su edad: "
    Leer edad

    Si edad ≥ 18 Entonces
        Escribir "Eres mayor de edad."
    FinSi

    Escribir "Fin del programa."
FinAlgoritmo
```

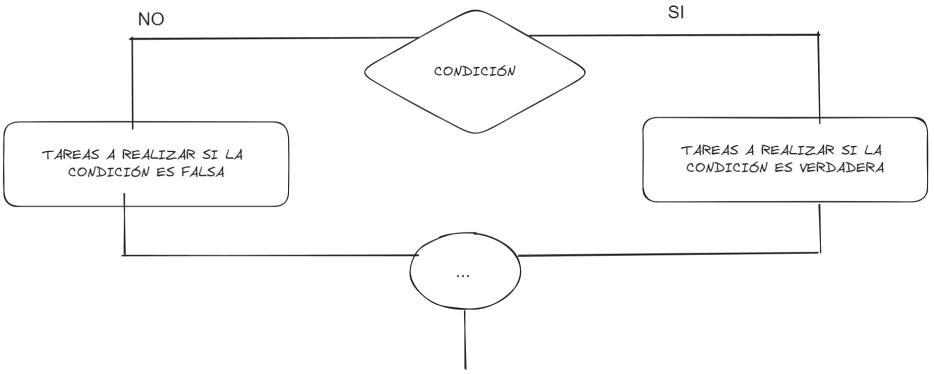
En este ejemplo, el programa solicita al usuario ingresar su edad. Luego, utiliza una estructura condicional simple para verificar si la edad ingresada es mayor o igual a 18.

Si la condición es verdadera, imprime "Eres mayor de edad.". Finalmente, el programa imprime "Fin del programa." independientemente del resultado de la condición.

CONDICIONAL DOBLE

Las estructuras condicionales dobles permiten elegir entre dos opciones o alternativas posibles, en función del cumplimiento o no de una determinada condición.

Por tanto, las estructuras condicionales dobles **presentan dos caminos diferentes que puede tomar el flujo de ejecución del programa.** Si la expresión a evaluar sale con resultado positivo, el programa se irá por una rama y si tiene resultado negativo se va por otra rama.




```
Si expresion_logica Entonces
    acciones_por_verdadero
SiNo
    acciones_por_falso
Fin Si
```

Ejemplo:

```
Algoritmo CondicionDoble
    Definir edad Como Entero

    Escribir "Ingrese su edad: "
    Leer edad

    Si edad < 18 Entonces
        Escribir "Eres menor de edad."
    Sino
        Escribir "Eres mayor de edad."
    FinSi

    Escribir "Fin del programa."
FinAlgoritmo
```

En este ejemplo, el programa solicita al usuario ingresar su edad. Luego, utiliza una estructura condicional doble para verificar si la edad ingresada es menor a 18. Si la condición es verdadera, imprime "Eres menor de edad.". Si no, imprime "Eres mayor de edad.". Finalmente, el programa imprime "Fin del programa." independientemente del resultado de la condición.

FUNCIONES PSEINT

Las funciones son herramientas proporcionadas por PSeInt que te ayudan a resolver ciertos problemas de manera más eficiente.

Por ejemplo, si necesitas calcular la raíz cuadrada de un número, PSeInt ofrece una función que, al proporcionarle un número, devuelve su raíz cuadrada. Este resultado puede asignarse a una variable o concatenarse con la instrucción "escribir" para mostrarlo directamente sin la necesidad de una variable adicional.

Además, las funciones pueden utilizarse dentro de cualquier expresión o estructura. Cuando evalúas la expresión, la función se reemplaza por su resultado correspondiente.

Existen dos tipos de funciones en PSeInt: las funciones matemáticas y las funciones de cadenas de texto. Las funciones matemáticas reciben un único parámetro numérico y devuelven un único valor numérico. Por otro lado, las funciones de cadenas de texto reciben un único parámetro de tipo cadena, pudiendo devolver un valor tanto de tipo cadena como numérico, dependiendo de la función utilizada.

FUNCIONES MATEMÁTICAS

Las funciones matemáticas son útiles para realizar operaciones numéricas. Algunas de las funciones disponibles son:

Funciones	Significado
RC(número)	Devuelve la raíz cuadrada del número.
ABS(número)	Devuelve el valor absoluto del número
LN(número)	Devuelve el logaritmo natural del número
EXP(número)	Devuelve la función exponencial del número.
SEN(número)	Devuelve el seno de número.
COS(número)	Devuelve el coseno de número.
TAN(número)	Devuelve la tangente de número.
ASEN(número)	Devuelve el arcoseno de número.
ACOS(número)	Arcocoseno de x
ATAN(número)	Arcotangante de x

MOD	Devuelve el módulo (resto de la división entera).
TRUNC(número)	Trunca el valor x (parte entera de x)
REDOND(número)	Redondea al valor más cercano a x
AZAR(número)	Entero aleatorio entre 0 y x -1
ALEATORIO(min,max)	Entero aleatorio entre valor mínimo y máximo

FUNCIONES DE CADENAS DE TEXTO

Estas funciones trabajan con cadenas de caracteres y pueden manipularlas de diversas formas. Algunas de las funciones más utilizadas son:

Funciones	Significado
Longitud(cadena)	Devuelve la cantidad de letras que compone la cadena.
Mayusculas(cadena)	Devuelve una copia de la cadena con todas sus letras en mayúsculas.
Minusculas(cadena)	Devuelve una copia de la cadena con todas sus letras en minúsculas.
Subcadena(cadena, posición_inicial, posición_final)	Devuelve una nueva cadena que consiste en la parte de la cadena que va desde la posición pos_inicial hasta la posición pos_final.
Concatenar(cadena, cadena2)	Devuelve una nueva cadena que resulta de unir las cadenas cadena1 y cadena2.
ConvertirANumero(cadena)	Recibe una cadena compuesta de números y devuelve la cadena como una variable numérica.
ConvertirACadena(cadena)	Recibe un número y devuelve una variable cadena de caracteres de dicho número.