



Fundamentos de Programación

+ Pseudocódigo avanzado



- Función
- Constantes y Variables
- Asignación de valor
- Acción u operación
- Condición booleana
- Álgebra de Boole
- Condicional
- Repetición

+ Pseudocódigo avanzado



Función:

- Contiene un algoritmo.
- Nombre
- Parámetro (entrada)
- Salida

Constantes:

- Datos cuyo valor no cambia durante la ejecución
- Se declaran y tienen un valor antes de utilizarse

Variables:

- Dato cuyo valor puede cambiar durante la ejecución
- Es una posición de memoria con nombre y que contiene un valor.

+ Pseudocódigo avanzado

Asignación de valor :

- Valores numéricos
- Valores de texto
- Valores booleanos (TRUE y FALSE)

Acción u operación:

- Operaciones sobre números (suma, resta, división, etc.)
- Operaciones de comparación (==, <>, !=, >, <, >=, <=)
- Operaciones booleanas (AND, OR, NOT)

Condición booleana

Condicional:

If (condición) {acción verdadera} else {acción falsa}

Repetición:

While(condición) do { acción{

Repeat{acción} until {condición}

+ Algebra de Boole

Estructura algebraica que esquematiza las operaciones lógicas Y, O, NO y SI

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

A	NOT A
0	1
1	0

+ Ejercicios

Dados a y b retornar un aumento de b % del valor de a

ej: `funcion(100,20) = 120`

Calcular el resto de dividir un número a , entre b

ej: `funcion(15,2) = 1`

Usando los operadores (/ , -, %) calcular la división entera de dos números

ej: `funcion(15,2) = 7`

Implementar la función XOR en booleanos usando AND, OR y NOT.

$\text{XOR}(a,b) = \text{True} \Leftrightarrow a \neq b$

ej: `funcion(true, false) = true`

ej: `funcion(true, true) = false`

+ *Ejercicios (2)*



Hacer en pseudocódigo el caso del restaurante de comida rápida.

+ *Lenguajes Interpretados*

JavaScript - Ruby – Python – PHP– Perl - MATLAB

El código fuente es traducido por el interprete a un lenguaje entendible para la máquina, instrucción por instrucción. El proceso se repite cada vez que se ejecuta el programa.

No es necesario inicializar una variable con determinado tipo de dato, puede cambiar su tipo dependiendo del valor.

Muchos son multiplataforma.

En comparación con algunos lenguajes compilados son más livianos.

Se utilizan en el desarrollo de aplicaciones o sitios web.

El tiempo que necesitan para ser interpretados, el tener que ser traducido con cada ejecución los hace más lento que en los lenguajes compilados.

+ *Lenguajes Compilados*



C - C++ - C# - Fortran – Pascal - Haskell - Visual Basic

Es traducido por un compilador a un archivo ejecutable entendible para la máquina en determinada plataforma.

El archivo es ejecutable, no hay necesidad de tener que repetir el proceso para una posterior ejecución.

Es necesario especificar tipo de dato al declarar la variable.

Común utilizarlos para software de escritorio.

Requieren de mayores recursos y acceso a archivos determinados.

+ *IDE*



Es un entorno gráfico de desarrollo que permite escribir el código fuente facilitando las tareas de escritura, compilado y debuggeo.

+ IDE (2)



Integrado: integra diferentes herramientas en el mismo entorno, como ser:

- *la compilación*
- *sugerencias mientras se escribe*
- *ver la estructura de un proyecto entero*
- *obtener ayuda sobre el lenguaje de programación, etc.*

Desarrollo: permite al programador escribir su código fuente.

Entorno: es una herramienta de software dentro de la cual se desarrolla el ciclo de desarrollo de un programa.

+ *IDE – Visual Code*



Visual Studio Code