

PROGRAMACIÓN I

GLOSARIO

algoritmo=procedimiento efectivo para resolver un problema en un numero finito de pasos. Ejemplo: receta de cocina. Todo algoritmo esta formado por 3 elementos: Secuencia, Decisión (selección) y Repetición (iteración).

código de bloque=area, parte o zona de un programa que esta delimitada por llaves: { <codigo de bloque> } . Por Ejemplo:

```
int main() {
    int variable1,i;
    for(i=0;i<10;i++) {
        int variable2;
        variable2=i+1;
        printf("variable2 vale: %d\n",variable2);
    }
    return 0;
}
```

este programa C muestra un código de bloque (el de la función main()), el cual tiene en su interior otro código de bloque (el de la instrucción for()). La variables variable1,i pertenecen al código de bloque mas amplio, mientras que la variable variable2 pertenece al código de bloque for mas interno. Es decir, intentar mostrar el valor de la variable variable2 en el código de bloque más amplio sería un error, variable2 no existe fuera del código de bloque en que fue declarada.

compilador=programa que toma como entrada un programa fuente (programa escrito por el programador) y genera un programa compilado o ejecutable (binario).

computador=circuito entrada/salida, tiene CPU, tiene memoria, ejecuta secuencia de instrucciones.

constante=similar a variable, pero en este caso el dato almacenado no cambia.

CPU=procesador, dispositivo electrónico capaz de interpretar instrucciones y realizar operaciones aritméticas y lógicas.

dato=valor que por si mismo no tiene mayor significado, ejemplo un numero cualquiera.

decisión/selección: una decisión toma la forma de “si ...proposición entonces acción”, Ejemplo 1: “Si hoy es miércoles entonces voy al cine”

Ejemplo 2: “Si hoy es miércoles entonces
 voy al cine
sino
 voy al teatro
FinSi”

hardware=se refiere a todos los elementos físicos, electrónicos, circuitos, plaquetas, etc.

información=dato con significado, en un contexto determinado. Ejemplo: un numero que representa una edad de una persona.

interprete=programa que permite interpretar y ejecutar un programa fuente

iteración/repetición:

las estructuras mas conocidas son: para, hacer – hasta, hacer – mientras, mientras

Ejemplo **para** (mostrar los numeros del 1 al 10):

```
para x valiendo 1 hasta 10 de a 1
    mostrar x
fin para
```

Ejemplo **hacer – hasta o repetir – hasta** (mostrar los numeros del 1 al 10):

```
x=0
hacer
    x = x + 1
    mostrar x
hasta x = 10
```

Ejemplo **hacer – mientras o repetir – mientras** (mostrar los numeros del 1 al 10):

```
x=0
hacer
    x = x + 1
    mostrar x
mientras x < 10
```

Ejemplo **mientras** (mostrar los numeros del 1 al 10):

```
x = 0
mientras x < 10
    x = x + 1
    mostrar x
fin mientras
```

lenguaje de programación=lenguaje artificial que permite escribir programas de computacion.

memoria=dispositivo electronico que nos permite almacenar instrucciones y datos, puede ser volátil (los datos se pierden cuando el circuito se apaga, memoria principal) o persistente (los datos no se pierden aunque se apague el circuito, memoria secundaria)

programa=conjunto de instrucciones que sirven para solucionar un problema. Algoritmo + dato.

programa fuente=programa escrito por un programador, expresado en un lenguaje de programación determinado. Por ejemplo, los programas fuentes en C terminan en .c; como prueba.c, test.c, etc. Los programas fuentes escritos en C++ terminan en .cpp (c plus plus); como prueba.cpp, test.cpp.

programa compilado/programa objeto=archivo resultante del proceso de compilación de un programa fuente. El compilador es un programa que lee un programa fuente y graba un programa compilado. No confundir el programa compilado con el programa ejecutable. El programa compilado se genera si no hay errores de compilación.

programa ejecutable/programa binario=archivo resultante del proceso de linkediación de un programa compilado/programa objeto. El linkeditor es un programa que lee un programa compilado/programa objeto, toma nota de cada llamada a cada función, incorpora el código binario de cada función llamada en el programa compilado y graba todo el código binario en el programa ejecutable/programa binario. El programa ejecutable se genera si no hay errores de linkediación.

proposición=expresión de la cual podemos decir que es verdad o mentira (verdadero o falso). Ejemplo: $a > b$, $4 > 2$, $3 \neq 1$, $f \neq a$, $!f$, $f(a > b) \&\& (a > c)$, $(a < b) \parallel (a < c)$, etc.

En el lenguaje C no existen los tipos de datos booleano o las constantes booleanas (podemos crearlos nosotros mismos), toda proposición se considera verdadera si es distinto de cero; se considera falsa si es igual a cero. El término "booleano" viene del Álgebra de Boole o Álgebra Booleana, inventada por el matemático George Boole en 1837.

secuencia=conjunto de pasos en un orden determinado acorde con el problema a resolver.

sistema operativo=conjunto de programas que permite el uso del computador de una forma sencilla por parte del usuario. Conjunto de programas que administran los recursos de un computador. Conjunto de programas que intermedian entre el usuario, las aplicaciones y el hardware del computador.

software=programas, conjuntos de instrucciones y datos que se ejecutan en un computador

variable=contenedor de un dato, lugar en donde almacenamos un dato, el cual puede variar a lo largo de la ejecución de un programa. Toda variable es de un *tipo de dato* determinado:

numero (entero, real), carácter (1 carácter o cadena de caracteres), booleano (verdadero, falso); etc. El tipo de dato también determina el espacio que ocupa en memoria, la cantidad de memoria que ocupa, ejemplo: una variable entera en C ocupa 4 bytes, un carácter ocupa 1 byte, etc.

Una variable representa un dato del programa y para facilitar su uso, toda variable tiene un *nombre* único (no confundible con otra variable) dentro de un programa. Una variable se puede inicializar con un valor y luego cambiarlo cuando se desee. Los valores que admite una variable están en relación con su tipo de dato y los valores máximos y mínimos que éste permite.

Una variable tiene un alcance dentro del programa que indica hasta dónde es válido referirse a ella, por ejemplo, en el lenguaje C, el alcance de una variable está limitado al código de bloque que la contiene.