**Programación I**

**Introducción**

Los sistemas modernos de computación consisten en una gran conjunción de elementos de circuitos (hardware) y de programación (software) que han sido diseñados para proporcionar a la computación un ambiente productivo y hasta cierta medida agradable.

         El término ***Sistema de Cómputo*** se utiliza para señalar lo que el usuario emplea, en lugar del término ***Computadora***. En los primeros años de la computación, los usuarios del sistema debían interactuar más estrechamente con el hardware real que lo que es hoy necesario, muchas funciones que debían realizar los usuarios mismos se manejan ahora por ***software*** mediante ***Sistemas Operativos***. El sistema operativo crea un ambiente en el cual los usuarios pueden preparar programas y ejecutarlos sin tener que entrar en los detalles del hardware del sistema.

         Para satisfacer el crecimiento de la demanda de medios de computación, es que se ha desarrollado la ***multiprogramación***, en la cual varios usuarios emplean el sistema de forma simultánea, como ***Windows***por ejemplo.

         Un término fundamental de esta materia es el ***Programa***, el cual es simplemente una secuencia de instrucciones que orienta a la Unidad de Control de Procesamiento (CPU) en el desarrollo de los cálculos, el cual debe expresarse de forma que sea entendido por el CPU. Un CPU sólo puede entender instrucciones que estén expresadas en términos de su lenguaje máquina, pero esto se explicará más adelante.

**Lenguajes de Programación**

Un ***Lenguaje de Programación***es aquél que es  utilizado para escribir programas de computadoras que puedan ser entendidos por ellas. Estos lenguajes se clasifican en tres grandes categorías:

 Lenguaje Máquina

 Lenguaje de Bajo nivel (ensamblador)

 Lenguaje de Alto nivel.

**Definición de Lenguajes Algorítmicos**

        Los algoritmos pueden describirse utilizando diversos lenguajes. Cada uno de estos lenguajes permiten describir los pasos con mayor o menor detalle.

 La clasificación de los lenguajes para algoritmos puede enunciarse de la siguiente manera :

 Lenguaje Natural.

 Lenguaje de Diagrama de Flujo.

 Lenguaje Natural de Programación.

 Lenguaje de Programación de Algoritmos.

 ***   Lenguaje Natural***

        Es aquél que describe en español, para nuestro caso, los pasos a seguir utilizando un vocabulario cotidiano. Se le conoce  como *lenguaje  jerga* cuando se utilizan términos especializados de una determinada ciencia, profesión o grupo.

 ***Lenguaje de Diagrama de Flujo***

        Es aquél que se vale de diversos símbolos para representar las ideas o acciones a desarrollar. Es útil para organizar las acciones o pasos de un algoritmo pero requiere de etapas posteriores para implementarse en un sistema de cómputo.

 ***Lenguaje Natural de Programación***

        Son aquéllos que están orientados a la solución de problemas que se definen de una manera precisa. Generalmente son aplicados para la elaboración de fórmulas o métodos científicos.

         Tiene las siguientes características:

 Evita la ambigüedad (algo confuso que se puede interpretar de varias maneras).

 Son precisos y bien definidos.

 Utilizan términos familiares al sentido común.

 Elimina instrucciones innecesarias.

 ***Lenguaje de Programación de Algoritmos***

            Es aquél que se utiliza para introducir en la computadora un algoritmo específico. Se les conoce también como ***Lenguaje de Programación.***

 ***   Lenguaje de Programación:***

        Es un conjunto de palabras, símbolos y reglas sintácticas mediante los cuales puede indicarse a la computadora los pasos a seguir para resolver un problema.

        Los lenguajes de programación pueden clasificarse por diversos criterios, siendo el más común su nivel de semejanza con el lenguaje natural, y su capacidad de manejo de niveles internos de la máquina.

 **Los principales tipos de lenguajes utilizados son tres :**

 Lenguaje Máquina.

 Lenguaje de bajo Nivel (ensamblador).

 Lenguajes de Alto Nivel.

 **** Lenguaje Máquina**

        Son aquéllos que están escritos en lenguajes directamente inteligibles por la máquina (computadora), ya que sus instrucciones son ***cadenas binarias*** *(*cadenas o series de caracteres de dígitos 0 y 1) que especifican una operación y las posiciones (dirección) de memoria implicadas en la operación se denominan ***instrucciones de máquina o código máquina****.* El código máquina es el conocido***código binario****.*

        Las instrucciones en lenguaje máquina dependen del hardware de la computadora y, por tanto, diferirán de una computadora a otra.

***Ventajas del Lenguaje Máquina***

   Posibilidad de cargar (transferir un programa a la memoria) sin        necesidad de traducción posterior, lo que supone una velocidad        de ejecución superior a cualquier otro lenguaje de programación.

 ***Desventajas del Lenguaje Máquina***

 Dificultad y lentitud en la codificación.

 Poca fiabilidad.

 Gran dificultad para verificar y poner a punto los programas.

 Los programas solo son ejecutables en el mismo procesador (CPU).

         En la actualidad, las desventajas superan a las ventajas, lo que hace prácticamente no recomendables a los lenguajes máquinas.

****Lenguajes de Bajo Nivel**

        Son más fáciles de utilizar que los lenguajes máquina, pero al igual que ellos, dependen de la máquina en particular. El lenguaje de bajo nivel por excelencia es el ***ensamblador***. Las instrucciones en lenguaje ensamblador son instrucciones conocidas como ***nemotécnicos***. Por ejemplo, nemotécnicos típicos de operaciones aritméticas son : en inglés : ADD, SUB, DIV, etc. ; en español : SUM, RES, DIV, etc.

 Una instrucción típica de suma sería :

ADD M, N, P

         Esta instrucción significa "sumar el contenido en la posición de memoria M al número almacenado en la posición de memoria N y situar el resultado en la posición de memoria P" . Evidentemente es más sencillo recordar la instrucción anterior con un nemotécnico que su equivalente en código máquina.

0110 1001 1010 1011

        Un programa escrito en lenguaje ensamblador, requiere de una fase de ***traducción***al lenguaje máquina para poder ser ejecutado directamente por la computadora.

        El programa original escrito en lenguaje ensamblador se denomina ***programa fuente*** y el programa traducido en lenguaje máquina se conoce como ***programa objeto***, el cual ya es directamente entendible por la computadora.

**Ventajas del lenguaje ensamblador frente al lenguaje máquina**

   Mayor facilidad de codificación y, en general, su velocidad de cálculo.

 **Desventajas del lenguaje ensamblador**

 Dependencia total de la máquina lo que impide la transportabilidad de los programas (posibilidad de ejecutar un programa en diferentes máquinas. El lenguaje ensamblador del PC es distinto del lenguaje ensamblador del Apple Machintosh.

  La formación de los programadores es más compleja que la correspondiente a los programadores de alto nivel, ya que exige no solo las técnicas de programación, sino también el conocimiento del interior de la máquina.

         Los lenguajes ensamblador tienen sus aplicaciones muy reducidas, se centran básicamente en aplicaciones de tiempo real, control de procesos y de dispositivos electrónicos.

 **** Lenguajes de Alto Nivel**

        Estos lenguajes son los más utilizados por los programadores. Están diseñados para que las personas escriban y entiendan los programas de un modo mucho más fácil que los lenguajes máquina y ensambladores. Un programa escrito en lenguaje de alto nivel es independiente de la máquina (las instrucciones no dependen del diseño del hardware o de una computadora en particular), por lo que estos programas son ***portables o transportables****.* Los programas escritos en lenguaje de alto nivel pueden ser ejecutados con poca o ninguna modificación en diferentes tipos de computadoras.

 **Ventajas de los lenguajes de alto nivel**

 El tiempo de formación de los programadores es relativamente corto comparado con otros lenguajes.

 La escritura de programas se basa en reglas sintácticas similares a los lenguajes humanos. Nombres de las instrucciones tales como READ, WRITE, PRINT, OPEN, etc.

 Las modificaciones y puestas a punto de los programas son más fáciles.

 Reducción del coste de los programas.

 Transportabilidad.

 **Desventajas de los lenguajes de alto nivel**

 Incremento del tiempo de puesta a punto al necesitarse diferentes traducciones del programa fuente para conseguir el programa definitivo.

 No se aprovechan los recursos internos de la máquina que se explotan mucho mejor en lenguajes máquina y ensambladores.

 Aumento de la ocupación de memoria.

 El tiempo de ejecución de los programas es mucho mayor.

 Para una mejor compresión de este tema se definirá el concepto de ***programa****,* por ser este un término muy utilizado en el ***diseño estructurado de algoritmos*.**

**Programa :**     Es un conjunto de instrucciones escritas en un lenguaje de programación que indican a la computadora la secuencia de pasos para resolver un problema.

***Definición de algoritmo***

       "Un ***algoritmo*** se define como un método que se realiza paso a paso para solucionar un problema que termina en un número finito de pasos".

 Las características fundamentales que debe cumplir todo algoritmo son :

 Debe ser ***preciso***. e indicar el orden de realización de cada paso.

 Debe ser ***definido***. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.

 Debe ser ***finito***. Si se sigue un algoritmo, se debe terminar en algún momento; o sea debe tener un número finito de pasos.

 La definición de un algoritmo debe describir tres partes: ***Entrada, Proceso y Salida*.**

***Algoritmos Cotidianos***

        Se refiere a todos aquéllos algoritmos que nos ayudan a resolver problemas diarios, y que los hacemos casi sin darnos cuenta de que estamos siguiendo una metodología para resolverlos.

 **Algunos ejemplos son:**

   Diseñar un algoritmo para cambiar una llanta a un coche.

1. Inicio.
2. Traer gato.
3. Levantar el coche con el gato.
4. Aflojar tornillos de las llantas.
5. Sacar los tornillos de las llantas.
6. Quitar la llanta.
7. Poner la llanta de repuesto.
8. Poner los tornillos.
9. Apretar los tornillos.
10. Bajar el gato.
11. Fin

   Un cliente ejecuta un pedido a una fábrica. La fábrica examina en su banco de datos la ficha del         cliente, si el cliente es solvente entonces la empresa acepta el pedido, en caso contrario rechazar          el pedido.

 Pasos del algoritmo:

Inicio

    Leer el pedido

    Examinar ficha del cliente

    Si el cliente es solvente aceptar pedido, en caso contrario rechazar pedido

Fin

    Determinar el mayor de tres números enteros.

Pasos del algoritmo :

1.-  Comparar el primero y el segundo entero, deduciendo cuál es el mayor.

2.-  Comparar el mayor anterior con el tercero y deducir cuál es el mayor. Este será el resultado.

         Los pasos anteriores se pueden descomponer en otros pasos más simples en los que se denomina ***refinamiento del algoritmo****.*

1.-  Obtener el primer número (entrada), denominado NUM1

2.-  Obtener el segundo número (entrada), denominado NUM2

3.-  Compara NUM1 con NUM2 y seleccionar el mayor ; si los dos enteros son iguales, seleccionar NUM1. Llamar a este número        MAYOR.

4.-  Obtener el tercer número (entrada), y se denomina NUM3.

5.-  Compara MAYOR con NUM3 y seleccionar el mayor; si los dos enteros son iguales, seleccionar el MAYOR. Denominar a este  número MAYOR.

6.-  Presentar el valor MAYOR (salida).

7.-  Fin