



INTRODUCCIÓN A LA PROGRAMACIÓN



FCyT

Facultad de Ciencia
y Tecnología



La informática

A lo largo del presente módulo se trabajará con el esquema general que se puede observar en la Figura 1, en la cual podemos observar que hay dos grandes componentes que forman un Sistema de Cómputo, el Hardware y el Software. A partir del mismo se profundizará con más detalle los componentes de este esquema de interacción y sus relaciones, a excepción del bloque usuario del cual no se analizará ningún aspecto, ya que estos exceden al área de la informática o por lo menos las analizadas en este módulo introductorio.

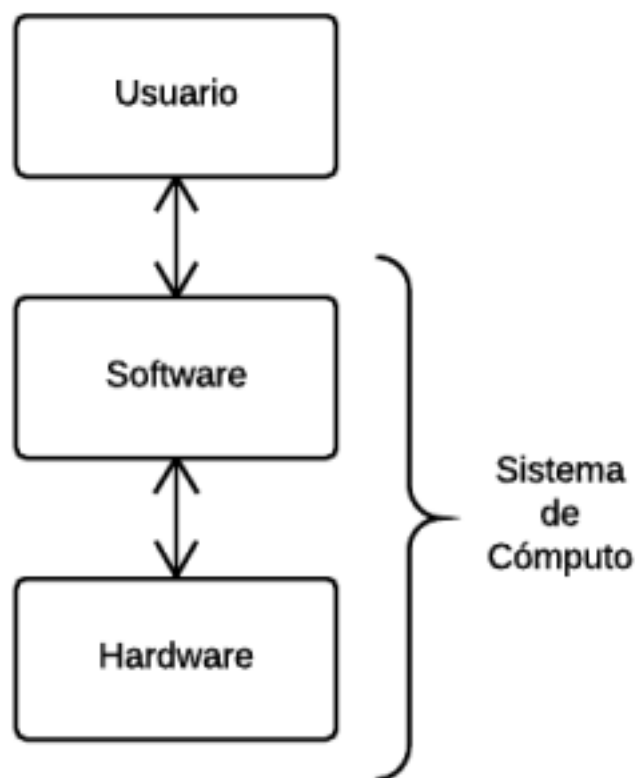


Figura 1. Esquema general de interacción

I. Hardware (HW)

El origen etimológico del término hardware lo encontramos claramente en el inglés. Y es que aquel está conformado por la unión de dos vocablos de la lengua anglosajona: hard que puede traducirse como “duro” y ware que es sinónimo de “cosas”.

Se puede definir al hardware como el conjunto de los componentes que conforman la parte material (física) de una computadora, no solo los componentes internos sino que también los externos (denominados periféricos).

El hardware se lo puede clasificar dependiendo su tipo, dando lugar a la siguiente clasificación:

- a. **Procesamiento:** se refiere a todos aquellos elementos cuyo propósito constituye la interpretación y ejecución de instrucciones, y el procesamiento de datos. Dentro de este ítem podemos encontrar a los microprocesadores, y a la Unidad Central de Procesamiento (CPU).
- b. **Almacenamiento:** como su nombre lo indica estos dispositivos tiene la capacidad de almacenar datos e información, ya sea de forma temporal o permanente. Principalmente cuando se habla de hardware de almacenamiento nos referimos a la memoria RAM y la memoria ROM. También pueden ser memoria USB, DVD, CD, pero en general estos son considerados como dispositivos de Entrada/Salida.
- c. **Entrada:** este tipo de hardware es utilizado con el objetivo de introducir datos e información a la CPU, también denominados periféricos de entrada. Algunos ejemplos de hardware de entrada son teclado, escáner, mouse, micrófono, webcam, etc.
- d. **Salida:** incluye a todos los dispositivos capaces de dirigir los datos generados por la computadora hacia el exterior, también denominados periféricos de salida. Algunos ejemplos de hardware de salida son impresora, monitor, auriculares, proyectores, parlantes, etc.
- e. **Entrada/Salida:** reciben esta denominación debido a que combinan las funcionalidades de los dispositivos de entrada y salida. Es decir, tienen la capacidad tanto de introducir datos e información, como de dirigirla hacia el exterior. Algunos ejemplos de hardware de entrada/salida son placa de red, router, placa de audio, discos rígidos, impresoras multifunción, etc.

La historia del desarrollo del hardware, por otra parte, marca diversos hitos tecnológicos o etapas denominadas generaciones. Estas se clasifican de la siguiente manera (los datos de fecha son aproximados):

a. **Primera Generación (1938-1958)**: el hito tecnológico que marcó esta generación fue el uso de tubos de vacío y tarjetas perforadas para su funcionamiento.

b. **Segunda Generación (1958-1963)**: el hito tecnológico que marcó esta generación fue el uso de transistores, logrando reducción de tamaño y aumento de velocidad respecto de la anterior.

c. **Tercera Generación (1964-1971)**: el hito tecnológico que marcó esta generación fue el uso de circuitos integrados (un chip de silicio), nuevamente se logró reducir de tamaño y aumentar de velocidad respecto de la anterior.

d. **Cuarta Generación (1983-hoy)**: el hito tecnológico que marcó esta generación fue el uso de microprocesadores, esto incrementó drásticamente los mismos aspectos antes mencionados tamaño (minimización) y velocidad (minimización).

e. **Quinta Generación (???)**: en muchos lugares se habla de una quinta generación, la cual es fundamentada como inteligencia artificial IA, lo cual es un gravísimo error hablar de esto dado que la IA es una técnica utilizada por distintos softwares para determinadas actividades y no un hito tecnológico que modifique el hardware. Por otro lado, el avance en los estudios sobre nanotecnología prevén la aparición de un hardware más avanzado a nano escala (explotando todas sus propiedades) en los próximos años, pero lo cierto es que hasta el momento no existe nada al respecto.

Además de toda la información expuesta hasta el momento tampoco podemos pasar por alto lo que se conoce como hardware libre. Este es un concepto tomado gran relevancia e importancia en estos últimos años. Esto hace referencia a que se puede acceder a los diseños y la programación del hardware. Uno de los líderes de este movimiento y tendencia es Arduino.

II. Software (SW)

El software es un conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora. Se considera que el software es el equipamiento lógico e intangible de un ordenador. En otras palabras, el concepto de software abarca a todas las aplicaciones informáticas. Este componente es esencial para el sistema de cómputo, ya que sin este no podríamos utilizar el hardware.

El software se lo puede clasificar dependiendo su tipo, dando lugar a la siguiente clasificación como se observa en la Figura 2:

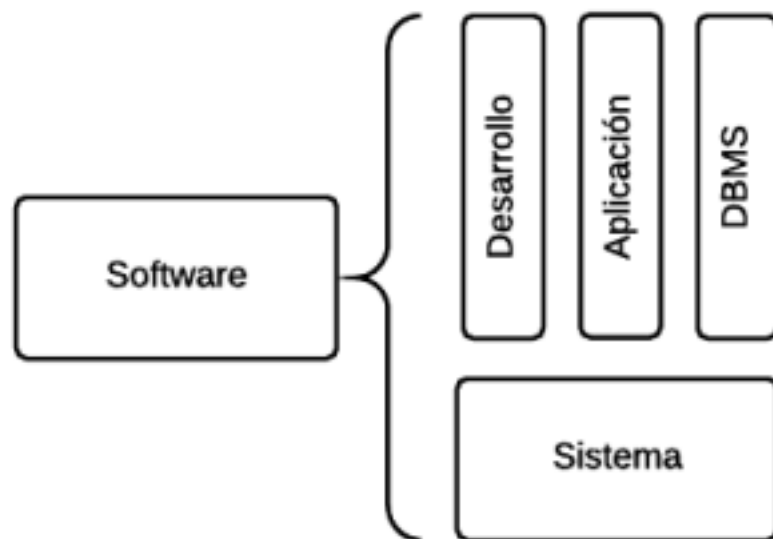


Figura 2. Clasificación del Software

a. **Sistema o controlador:** es aquel que permite a los usuarios interactuar con el hardware así como también controlarlo. Este sistema está compuesto por una serie de programas que tienen como objetivo administrar los recursos del hardware y, al mismo tiempo, le otorgan al usuario una interfaz. Algunos ejemplos de este software son los sistemas operativos Linux, Mac, Android y Windows. Además de lo mencionado anteriormente los controladores de dispositivos también son catalogados como este tipo de software.

b. **Desarrollo:** son aquellas herramientas que un programador utiliza para poder desarrollar programas informáticos (software). Básicamente podemos decir que es un software para desarrollar software. Para esto, el programador se vale de distintos lenguajes de programación. Algunos ejemplos son compiladores, programas de diseño asistido por

computador, paquetes integrados, editores de texto, enlazadores, depuradores, intérpretes, IDE, etc.

c. **Aplicación:** aquí se incluyen todos aquellos programas que permiten al usuario realizar una o varias tareas específicas. Algunos ejemplos son procesadores de texto, hojas de cálculo, editores de imágenes, videojuegos, etc.

d. **DBMS:** son las siglas en inglés para los Sistemas de Gestión de Bases de Datos. Es un software que controla la organización, almacenamiento, recuperación, seguridad e integridad de los datos en una base de datos con el objetivo primordial de lograr la independencia de los datos. Acepta solicitudes de la aplicación y ordena al sistema operativo transferir los datos apropiados. Los DBMS pueden trabajar con distintos lenguajes de programación tradicionales.

El software también se lo puede clasificar de otra manera a la antes mencionada, esta depende su tipo de licencia, dando lugar a la siguiente clasificación:

a. **Libre:** software libre es aquel que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es decir, es una cuestión de libertad, no de precio. Para entender el concepto, piense en “libre” como en “libre expresión”, no como en “libre de costo”.

El software libre suele estar disponible gratuitamente, o al precio de coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar el término de “software libre” al de “software gratuito”.

b. **Propietario:** software propietario o privativo, es aquel software del cual no existe una forma libre de acceso a su código fuente. El cual solo se encuentra a disposición de su desarrollador y no se permite su libre modificación, adaptación o incluso lectura por parte de terceros.

III. Arquitectura de Von Neumann

La arquitectura Von Neumann, también conocida como modelo de Von

Neumann o arquitectura Princeton, es una arquitectura de computadoras basada en la descrita en 1945 por el matemático y físico John Von Neumann y otros, en el primer borrador de un informe sobre el EDVAC.

Un computador digital de programa almacenado es uno que mantiene sus instrucciones de programa, así como sus datos, en memoria de acceso aleatorio (RAM) de lectura-escritura. Las computadoras de programa almacenado representaron un avance sobre los ordenadores controlados por programas de la década de 1940, como la Colossus y la ENIAC, que fueron programadas ajustando interruptores e insertando parches, conduciendo datos de la ruta y para controlar las señales entre las distintas unidades funcionales. En la gran mayoría de las computadoras modernas, se utiliza la misma memoria tanto para datos como para instrucciones de programa. La Figura 3 muestra la arquitectura de modelo de Von Neumann.

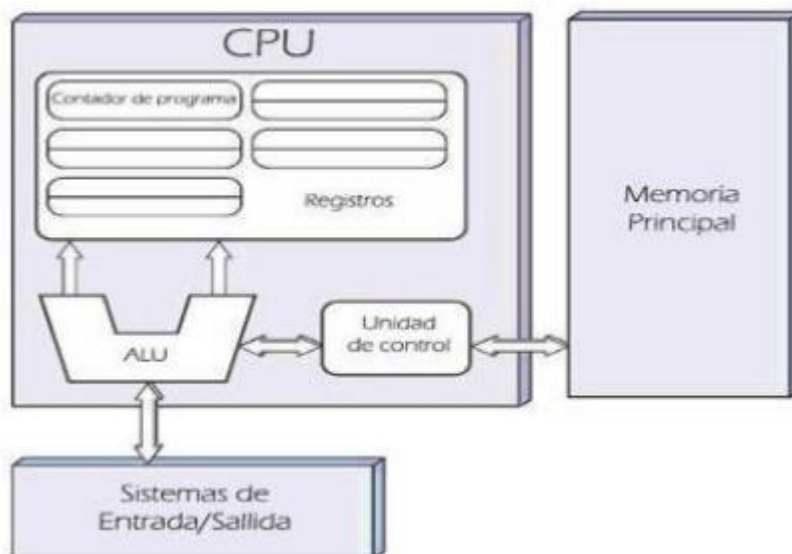


Figura 3. Arquitectura Modelo de Von Neumann

Veamos ahora la descripción en detalle de cada uno de los componentes de esta arquitectura:

a. **Unidad Central de Proceso (CPU)**, es el lugar en el que se procesa la información de acuerdo a las instrucciones del programa. La CPU de un sistema informático que repite una serie de pasos en los que continuamente accede a memoria para leer la próxima instrucción a ejecutar, realiza lo que ordena la instrucción y vuelta a empezar.

b. **Unidad aritmético lógica (ALU)**, se encarga de realizar las operaciones aritméticas (sumas, restas, multiplicaciones, etc.) y lógicas (AND, OR, rotaciones, desplazamientos, etc.) con los datos. Normalmente los datos con los que opera, así como los resultados de la operación se encuentran en registros de la CPU.

c. **Registros**, proporciona un espacio de almacenamiento para los datos con los que trabaja la CPU. Los registros se deben cargar con información que proviene de la memoria principal antes de comenzar a operar, cuando se necesita dejar espacio libre en el banco de registros para operar con nuevos datos su valor debe escribirse en la memoria principal. Todos los registros no tienen las mismas características. Normalmente se distingue entre:

- i. Registros de datos: Guardan la información con la que se trabaja.
- ii. Registros de direcciones: Guardan direcciones de memoria.
- iii. Registros de control: Controlan el estado de la CPU.
- iv. Registro de instrucción: Guarda la instrucción que se está ejecutando.
- v. Contador de programa: Registro que apunta a la dirección de memoria de la próxima instrucción a ejecutar. Se incrementa automáticamente después de ejecutar cada instrucción.

d. **Unidad de control**, se encarga de leer las instrucciones máquina almacenadas en la memoria principal y de generar las señales de control necesarias para controlar y coordinar el resto de las unidades funcionales de un ordenador, con el fin de ejecutar las instrucciones leídas.

e. **Memoria principal**, tiene por objeto guardar información que es accesible a la CPU. La CPU puede leer y/o escribir datos en las diferentes posiciones de memoria que componen la memoria principal. La memoria principal tiene menor capacidad que la memoria secundaria (que virtualmente es ilimitada), pero es mucho más rápida.

f. **Controlador de Entrada/Salida (I/O)**, en la medida en la que el

sistema informático precisa comunicarse con el mundo exterior (utilizando diferentes periféricos), es necesario un elemento que controle el flujo de información que entra y/o sale del sistema informático. Los periféricos del sistema informático se pueden clasificar en:

- i. Periféricos de entrada.
- ii. Periféricos de salida
- iii. Periféricos de entrada/salida

g. **Buses del sistema**, los buses son las vías de comunicación que permiten mover la información entre los distintos elementos de la arquitectura Von Neumann. Los tres tipos de buses son:

- i. **Bus de datos**: Como su nombre indica transporta datos. Estos datos pueden ser la información que se está procesando o las instrucciones del programa que se ejecuta.
- ii. **Bus de direcciones**: El bus de direcciones se utiliza para indicar el origen y/o el destino de los datos. En el bus de direcciones se indica la posición de memoria a la que se está accediendo en cada momento. Puede tratarse de una dirección de la memoria principal o puede tratarse de una dirección de memoria en la que está mapeado un periférico.
- iii. **Bus de control**: El bus de control proporciona señales para coordinar las diferentes tareas que se realizan en el sistema informático.

IV. Sistemas Operativo (SO)

Un sistema operativo es el software principal o conjunto de programas de un sistema informático que gestiona los recursos de hardware y provee servicios a los programas de aplicación de software. Es una de las piezas de software más complejo desarrollado. Como se Observa en la Figura 4 es el nexo que uno el Hardware con el Software.



Figura 4. Comunicación entre Hardware y Software

El sistema operativo cumple con tres funciones básicas:

- a. Suministrar una interfaz amigable al usuario y/o aplicación.
- b. Administración de recursos.

En cuanto a la interfaz al usuario, el sistema se encarga de que el sujeto pueda cargar programas, acceder a archivos y realizar otras tareas con la computadora. Puede ser mediante una consola de comando o mediante ventanas gráficas.

La administración de recursos permite la dirección del hardware, entiéndase por recursos los siguientes aspectos:

- a. Procesador y procesos
- b. Memoria
- c. Archivos
- d. Periféricos de Entrada/Salida
- e. Acceso a recursos
- f. Seguridad

V. Sistemas Numéricos

Un sistema numérico es un conjunto provisto de dos operaciones que verifican ciertas condiciones relacionadas con las propiedades conmutativa, asociativa y distributiva. Además debe tenerse en cuenta que dado un sistema numérico existen diversas formas de representarlo, por ejemplo para el conjunto de los

enteros podemos usar la representación decimal, la binaria, la hexadecimal, etc.

a. El **Sistema Binario** en ciencias de la computación, es un sistema de numeración en el que los números se representan utilizando solamente dos cifras, cero y uno (0 y 1). Es uno de los sistemas que se utilizan en las computadoras, debido a que estas trabajan internamente con dos niveles de voltaje, por lo cual su sistema de numeración natural es el sistema binario.

b. El **Sistema Hexadecimal** es el sistema de numeración posicional que tiene como base el 16. Su uso actual está muy vinculado a la informática y ciencias de la computación donde las operaciones de la CPU suelen usar el byte u octeto como unidad básica de memoria. En principio, dado que el sistema usual de numeración es de base decimal y, por ello, sólo se dispone de diez dígitos, se adoptó la convención de usar las seis primeras letras del alfabeto latino para suplir los dígitos que nos faltan. El conjunto de símbolos sería, por tanto, el siguiente:

$$S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$$

Se debe notar que $A = 10$, $B = 11$, $C = 12$, $D = 13$, $E = 14$ y $F = 15$. En ocasiones se emplean letras minúsculas en lugar de mayúsculas. Como en cualquier sistema de numeración posicional, el valor numérico de cada dígito es alterado dependiendo de su posición en la cadena de dígitos. El sistema hexadecimal actual fue introducido en el ámbito de la computación por primera vez por IBM en 1963.

VI. Introducción a la Programación

Al hablar de programación se hace referencia al efecto de programar, es decir, de organizar una secuencia de pasos ordenados a seguir para hacer cierta cosa.

Sin embargo, uno de los usos más amplios y complejos de la palabra programación se da en el ámbito de la informática, para hablar de la acción de crear programas o aplicaciones, a través del desarrollo de un código fuente, el cual se basa en el conjunto de instrucciones que sigue el ordenador para ejecutar un programa. Estas instrucciones se encuentran escritas en lenguaje

de programación que luego son traducidas a un lenguaje de máquina, que puede ser interpretado y ejecutado por el hardware del equipo (parte física del equipo). Dicho código fuente es creado, diseñado, codificado, mantenido y depurado a través de la programación, donde el principal objetivo a lograr es el desarrollo de sistemas que sean eficaces, accesibles y agradables o amigables para el usuario.

Los programas informáticos suelen denominarse algoritmos, que son el conjunto de instrucciones que se encuentran organizadas y relacionadas entre sí de cierta manera, y que permiten llegar a la solución de un problema, y que a su vez contienen módulos más pequeños que resuelven detalles más específicos.

Una variable es un espacio reservado en memoria principal, al cual se referencia o accede mediante un identificador o nombre definido por el usuario asociado a dicho espacio. Además dependiendo del lenguaje de programación utilizado, estas variables pueden ser débilmente o fuertemente tipadas, para este módulo utilizaremos variables fuertemente tipadas. Por lo que a la definición antes mencionada le agregamos que la variable tiene asignado un tipo de datos, es decir el tipo de información que en ella se almacenará (por ejemplo entero, real, carácter, etc.).

Durante el desarrollo de este módulo de introducción trabajaremos con pseudocódigo. En ciencias de la computación, y análisis numérico, el pseudocódigo es una descripción de alto nivel compacta e informal del principio operativo de un programa informático u otro algoritmo.

Utiliza las convenciones estructurales de un lenguaje de programación real, pero está diseñado para la lectura humana en lugar de la lectura mediante máquina, y con independencia de cualquier otro lenguaje de programación. Normalmente, el pseudocódigo omite detalles que no son esenciales para la comprensión humana del algoritmo, tales como declaraciones de variables, código específico del sistema y algunas subrutinas.

Instrucciones Básicas

| Instrucción | Descripción |
|-------------|-------------|
|-------------|-------------|

| | |
|---------------------------|--|
| Leer(variable) | representa la lectura de un valor ingresado por el usuario mediante el teclado, dicho valor se almacena en la variable. |
| Mostrar(variable frase) | representa que se mostrará en pantalla el valor almacenado en la variable o una frase o leyenda. |
| identificador : tipo_dato | define una variable con el nombre que pongamos como identificador del tipo de dato que determinemos después de los dos puntos. |
| $a \leftarrow b \mid 10$ | asignación guardar un valor en una variable, asigna a la variable a el valor de otra variable o un determinado valor. |

Operaciones Aritméticas

| Operador | Descripción | Ejemplo | Resultado |
|----------|---|----------------------|-----------|
| + | suma dos valores o variables | suma = a +45 | 75 |
| - | resta dos valores o variables | resta = 13 - b | 3 |
| * | multiplica dos valores o variables | area = base * altura | 10 |
| / | divide dos valores o variables (decimal) | mitad = valor / 2 | 3.5 |
| div | divide dos valores o variables (entera) | mitad = valor div 2 | 3 |
| mod | calcula el resto de la división entera de dos valores o variables | resto = 5 mod 2 | 2 |

* donde a=30, b=10, altura=2, base=5, valor=7

Operadores de Control

| Operador | Descripción | Ejemplo | Resultado |
|----------|---------------|-----------|-----------|
| > | mayor | 10 > 5 | verdadero |
| < | menor | - 10 < 0 | verdadero |
| >= | mayor o igual | 5 >= 14 | falso |
| <= | menor o igual | -5 <= -15 | false |
| = | igual | 10 = 4 | false |
| <> | distinto | 3 <> -3 | verdadero |

* los ejemplos se leen de izquierda a derecha, pueden utilizarse con variable también

Estructuras Condicionales

Las estructuras condicionales nos permiten ejecutar cierta parte de nuestro algoritmo cuando se cumple o no determinada condición. Entiéndase por condición una expresión lógica una comparación entre dos variables o valores utilizando los operadores de control antes visto (por ejemplo $a > 10$, $b = 5$, etc.).

Condicional Simple, esto implica que si se cumple la condición, es decir es verdadero, se ejecutará la acción (una instrucción o un conjunto de instrucciones definidas).

Si condición Entonces acción

Condicional Compuesto, esto implica que si se cumple la condición, es decir es verdadero, se ejecutarán las acciones 1 y sino, es decir es falso, se ejecutarán las acciones 2 (Siendo acciones 1 y acciones 3, una instrucción o un conjunto de instrucciones definidas)

Si condición Entonces acciones 1

Sino acciones 2

Condicional Anidado, esto implica que dentro de cada uno de los bloques de instrucciones sea la condición verdadero o falso, podemos definir nuevos condicionales de cualquiera de los tipos antes mencionados, permitiéndonos resolver problemas más complejos.

Operadores Lógicos

Tabla de verdad operador "AND", este operador se interpreta de la siguiente manera, si condición 1 y condición 2 --> resultado.

| condición 1 | condición 2 | resultado |
|-------------|-------------|-----------|
| verdadero | verdadero | verdadero |
| verdadero | falso | falso |
| falso | verdadero | falso |
| falso | falso | falso |

* único caso que el resultado es verdadero es que ambas condiciones sean verdadero

Tabla de verdad operador "OR", este operador se interpreta de la siguiente manera, si condición 1 o condición 2 --> resultado.

| condición 1 | condición 2 | resultado |
|-------------|-------------|-----------|
| verdadero | verdadero | verdadero |
| verdadero | falso | verdadero |
| falso | verdadero | verdadero |
| falso | falso | falso |

* único caso que el resultado es falso es que ambas condiciones sean falso

VII. Guía de Ejercicios

Resolver los siguientes ejercicios en sistema binario:

- $101 + 11$
- $110 + 100$
- $1001 + 111 + 10$
- $10111011 + 11111101 + 111101$
- $111111111111 + 111001 + 11001 + 100$
- $10001 + 111 + 1001$
- $100 * 10$
- $1001 * 111$
- $10 * 11$
- $1000111101 * 1101101$
- $10001 + 1000 * 11110$
- $1110 + 1000 * 1111 + 100$

Resolver los siguientes ejercicios en sistema hexadecimal:

- $9 + A$
- $19 + 5$
- $AF3 + 1A0 + B$

- d. $10 + A + FF5ED + C4$
- e. $CB145A + 001F + AB + FFF$
- f. $F130 + 11C + A99$
- g. $F45D + 999 + 561$
- h. $A48 + EE31 + 91$
- i. $211D + 480 + CC1$
- j. $123 + ABCD + 456$

Resolver los siguientes ejercicios en pseudocódigo:

- a. Calcular y mostrar superficie y perímetro de un cuadrado
- b. Ingresar la edad de una persona y calcular su año de nacimiento
- c. Cargar el precio de un artículo y mostrar su equivalente en dólares
- d. Cargar el precio de costo y la ganancia de un artículo y mostrar su precio de venta
- e. Calcular y mostrar la nota promedio de los tres parciales de un alumno
- f. Determinar si un número es positivo
- g. Determinar si un número es par o impar
- h. Dados dos número obtener el mayor y calcular su cuadrado
- i. Determinar si un número es par y múltiplo de 5
- j. Determinar si un número está entre 3 y 81 inclusive ambos valores
- k. Dada la edad y el nombre de dos personas mostrar el nombre del mayor
- l. Dado el número de un mes (1-12) determinar cuántos días tiene dicho mes
- m. Determinar el mayor de tres número
- n. Dada una temperatura y su escala (C o F) convertir dicho valor a la otra escala
- o. Dado dos número y un carácter (+, -, *, /), resolver la operación correspondiente al carácter ingresado con los números

Referencias

- [1] M. C. Guinzburg - Introducción General a la Informática – La PC por dentro
- [2] Patricia Quiroga - Arquitectura de Computadoras
- [3] William Stallings - Organización y Arquitectura de Computadoras - Ed. Prentice Hall
- [4] Joyanes Aguilar - Programación en Turbo Pascal - Addison Wesley.
- [5] Dale Lilly - Pascal y Estructuras de datos - Mc Graw Hill