2012

UNAN – LEÓN
Departamento de Computación

Autor:
Ing: Karina Esquivel Alvarado.

Asignatura:
LÓGICA DE PROGRAMACIÓN



UNIDAD I:

CONOCIMIENTO DE LA COMPUTADORA

UNIDAD I: CONOCIMIENTO DE LA COMPUTADORA

1.1 INTRODUCCIÓN:

La computadora no ha cumplido ni los cien años de existencia desde su primera generación. Sin embargo es un invento que ha venido a revolucionar la forma en la que trabajamos, nos entretenemos y se ha convertido en un aparato esencial en nuestra vida diaria.

La computadora no solamente es una máquina que puede realizar procesos para darnos resultados, sin que tengamos la noción exacta de las operaciones que realiza para llegar a esos resultados. Con la computadora además de lo anterior también podemos diseñar soluciones a la medida de problemas específicos que se nos presenten. Más aún, si estos involucran operaciones matemáticas complejas y/o repetitivas, o requieren del manejo de un volumen muy grande de datos.

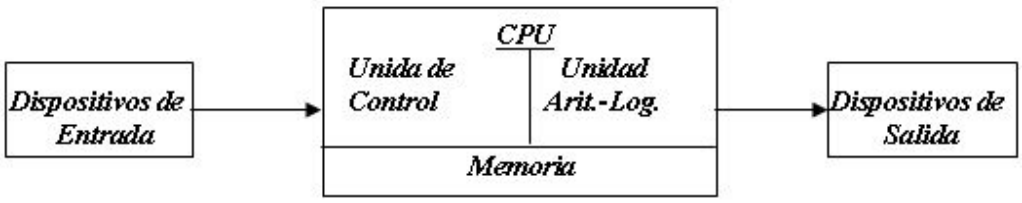
Computadora: Es un dispositivo electrónico utilizado para procesar información y obtener resultados. Los datos y la información se pueden introducir en la computadora como entrada (input) y a continuación se procesan para producir una salida (output). Se ha convertido en un auxiliar del hombre que le presta ayuda en una enorme gama de actividades, tales como:

• Proveer a los médicos información actualizada sobre la salud del paciente.
• Preparar gráficas de patrones climatológicos y producir planes de vuelo de acuerdo con las condiciones climatológicas.
• Prestar ayuda en la oceanografía y la navegación.
• Registrar marcas y evaluar estadísticas de competencias deportivas.
• Prestar ayuda a los ingenieros en los cálculos.
• Controlar simuladores de vuelo para dar al piloto un entrenamiento inicial en tierra.
• Coordinar el funcionamiento de los semáforos para que el tránsito no sufra interrupciones.
• Verificar la cantidad de dinero depositado en una cuenta. Un empleado del banco habla con una computadora a través del teléfono.
• Proveer información sobre los productos en el mercado.

Proceso de información en la computadora



1.2 ORGANIZACIÓN FÍSICA DE UNA COMPUTADORA:



♦ **Dispositivos de Entrada:** Como su nombre lo indica, sirven para introducir datos (información) en la computadora para su proceso. Los datos se leen de los dispositivos de entrada y se almacenan en la memoria central o interna. **Ejemplos:** teclado, scanners (digitalizadores, mouse (ratón), joystick (palancas de juego), lápiz óptico.

♦ **Dispositivos de Salida:** Regresan los datos procesados que sirven de información al usuario. **Ejemplo:** monitor, impresora.

♦ **Unidad Central de Procesamiento (C.P.U) se divide en dos:**

- **Unidad de Control:** Coordina las actividades de la computadora y determina que operaciones se deben realizar y en que orden; así mismo controla todo el proceso de la computadora.
- **Unidad Aritmético - Lógica:** Realiza operaciones aritméticas y lógicas, tales como suma, resta, multiplicación, división y comparaciones.

♦ **La Memoria de la computadora se divide en dos:**
♦ **Memoria Central o Interna**
♦ **Memoria Auxiliar o Externa**

- **Memoria Central (interna):** La CPU utiliza la memoria de la computadora para guardar información mientras trabaja con ella; mientras esta información permanezca en memoria, la computadora puede tener acceso a ella en forma directa. La memoria interna consta de dos áreas de memoria:
  - **La memoria RAM (Randon Access Memory):** Recibe el nombre de memoria principal o memoria del usuario, en ella se almacena información solo mientras la computadora esta encendida. Cuando se apaga o arranca nuevamente la computadora, la información se pierde, por lo que se dice que la memoria RAM es una memoria volátil.

- **La memoria ROM (Read Only Memory):** Es una memoria estática que no puede cambiar, la computadora puede leer los datos almacenados en la memoria ROM, pero no se pueden introducir datos en ella, o cambiar los datos que ahí se encuentran; por lo que se dice que esta memoria es de solo lectura. Los datos de la memoria ROM están grabados en forma permanente y son introducidos por el fabricante de la computadora.

- **Memoria Auxiliar (Externa):** Es donde se almacenan todos los programas o datos que el usuario desee. Los dispositivos de almacenamiento o memorias auxiliares (externas o secundarias) más comúnmente utilizados son: cintas magnéticas y discos magnéticos.

1.3 ELABORACIÓN DE PROGRAMAS PARA COMPUTADORA:

Las computadoras pueden ejecutar un gran número de operaciones a muy altas velocidades y con intervención mínima de seres humanos. Sin embargo, a una computadora se le debe indicar exactamente qué operaciones debe ejecutar y en qué orden. A estos conjuntos de instrucciones se les denomina **programas** para computadora y son elaborados por programadores. Un **programa** es una serie de instrucciones escritas en forma codificada que la computadora puede traducir a su propio lenguaje.

La solución de problemas por medio de computadora debe emplear la siguiente metodología:

♦ **Definición del Problema:** Esta fase está dada por el enunciado del problema, el cual requiere una definición clara y precisa. Es importante que se conozca lo que se desea que realice la computadora; mientras esto no se conozca del todo no tiene mucho caso continuar con la siguiente etapa.

♦ **Análisis del Problema:** Una vez que se ha comprendido lo que se desea de la computadora, es necesario definir:

- Los datos de entrada.
- Cual es la información que se desea producir (salida)
- Los métodos y fórmulas que se necesitan para procesar los datos.

Cuando se construye una aplicación o se modifica una que ya estaba en uso, se realiza un análisis de sistemas. Este análisis determina los tipos de procesamiento de datos que se requieren, cómo y dónde deben registrarse los datos, los tipos de informe a producirse y los tipos de equipo que mejor se adaptan a esa aplicación particular. Una vez que se ha decidido lo que se hará con los datos, se le encomienda a un programador la elaboración de un programa o conjunto de instrucciones para el sistema de computación.

Al escribir el programa que se va a usar, el programador sigue una serie de etapas, que deben cumplirse. Cuando a un programador se le pide que elabore un programa, se le suele indicar cómo debe ser la salida y qué datos de entrada están disponibles. En la etapa de análisis de sistemas se deciden estas cuestiones. Cuando el programador analiza la entrada debe saber qué dispositivos se van a emplear, en qué orden están colocados los campos de datos en los registros y cuál es el tamaño de esos campos, ya que a menudo los registros contienen más datos de los necesarios para cualquier trabajo.

En el cuerpo del programa se describe qué operaciones se van a ejecutar con los datos; se decide en qué orden se van a ejecutar las operaciones y en qué punto se van a tomar decisiones (ramificaciones).

Por último, el programador analiza los requisitos de la salida: si se debe grabar en cinta o imprimir un listado; si la salida va a ser impresa, cuáles deben ser los títulos y dónde se deben colocar los campos de salida.

Antes de pasar a la siguiente etapa, es necesario hacer todas esas preguntas y encontrar las respuestas. Para ello, el programador tiene que hablar con las personas que diseñaron el sistema y con las personas que utilizan los informes. Con estas últimas hablará principalmente acerca de la forma como desean tener los datos de salida, para que sean de mayor utilidad.

♦ **Diseño del Algoritmo:** Cuando el programador tiene toda la información necesaria acerca del programa, elabora un **algoritmo**. Los algoritmos son muy útiles porque muestran, en orden, las operaciones que se van a ejecutar con los datos, las comparaciones, y las ramificaciones condicionales e incondicionales que formarán parte del programa.

Un algoritmo es una serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico.

Los tipos de algoritmos son:

♦ **Cualitativos:** Son aquellos en los que se describen los pasos utilizando palabras.

♦ **Cuantitativos:** Son aquellos en los que se utilizan cálculos numéricos para definir los pasos del proceso.

Lenguajes Algorítmicos: Serie de símbolos y reglas que se utilizan para describir de manera explícita un proceso. Los tipos de lenguajes algorítmicos son:

♦ **Gráficos:** Es la representación gráfica de las operaciones que realiza un algoritmo (diagrama de flujo).

♦ **No Gráficos:** Representan en forma descriptiva las operaciones que debe realizar un algoritmo (pseudocódigo).

Al terminar la elaboración del algoritmo se puede verificar la exactitud general del programa propuesto. El programador puede tomar algunos datos como muestra y suponer que él es el sistema de computación. Él hará con los datos, exactamente, lo que el algoritmo indique.

La salida final que obtenga debe ser exactamente lo que espera. Si no es así, suponiendo que cometió errores por descuido, su algoritmo tiene algunas instrucciones equivocadas y debe ser corregido. Obtener la salida correcta no es garantía de que el programa final será perfecto, debido a que aún quedan muchas cosas por hacer. No obstante, esta simple verificación mostrará las imperfecciones lógicas que pueda haber en sus planteamientos. A la verificación del diagrama detallado se le conoce como prueba de escritorio.

Ejemplo: Algoritmo representado mediante un pseudocódigo que suma dos números.

```
Proceso Suma
    Leer a,b;
    c <- a+b;
    Escribir 'La suma es:', c;
FinProceso
```

- **Codificación:** Con base en el algoritmo terminado, se escribe una serie de instrucciones detalladas. Las que se denominan **programa fuente** y suelen estar escritas en algún lenguaje de programación.

Lenguajes de Programación: Es un conjunto de símbolos, caracteres y reglas (programas) que le permiten a las personas comunicarse con la computadora. Los lenguajes de programación tienen un conjunto de instrucciones que nos permiten realizar operaciones de entrada/salida, calculo, manipulación de textos, lógica/comparación y almacenamiento/recuperación.

Los lenguajes de programación se clasifican en:

- **Lenguaje Máquina:** Son aquellos cuyas instrucciones son directamente entendibles por la computadora y no necesitan traducción posterior para que la CPU pueda comprender y ejecutar el programa. Las instrucciones en lenguaje maquina se expresan en términos de la unidad de memoria mas pequeña el bit (dígito binario 0 o 1).
- **Lenguaje de Bajo Nivel (Ensamblador):** En este lenguaje las instrucciones se escriben en códigos alfabéticos conocidos como mnemotécnicos para las operaciones y direcciones simbólicas.
- **Lenguaje de Alto Nivel:** Son aquellos en los que las instrucciones o sentencias a la computadora son escritas con palabras similares a los lenguajes humanos, lo que facilita la escritura y comprensión del programa, entre los cuales se encuentran: **BASIC, PASCAL, FORTRAN, C, C++, JAVA.**

La selección del lenguaje para el programa fuente dependerá de muchos factores. Dos de las consideraciones principales son el tipo de tarea que se programó y la computadora que se utiliza. Luego de escrito, el programa fuente se almacena en: disco duro, memorias USB, etc.

El sistema de computación no puede entender directamente el lenguaje que usó el programador. Antes debe ser traducido a lenguaje de máquina. El compilador efectúa esta traducción. El **compilador** es un programa especial que única y exclusivamente traduce el programa escrito en un lenguaje de programación al lenguaje de máquina. Al programa traducido se le denomina **programa objeto** y puede colocarse directamente en la memoria o almacenarse en discos o cintas.

♦ **Prueba y Depuración:** Cuando el programa fuente se desea traducir a lenguaje de máquina, o sea, cuando se compila, sucede otra cosa importante. Dado que el programa fuente puede contener muchos errores, el compilador produce una lista impresa en donde aparece el programa fuente y otra lista de todos los errores que contiene el programa. El programador utiliza esta lista de mensajes de error cuando está buscando y eliminando los errores de su programa. Los errores que detecta el compilador se denominan diagnósticos del tiempo de compilación (errores de sintaxis). Indican en qué instrucción hay un error y de qué tipo es éste. Todos los errores señalados por los diagnósticos del tiempo de compilación deben corregirse antes de que el sistema pueda procesar los datos.

Una vez que se han corregido los errores del tiempo de compilación, deben usarse algunos datos para probar el programa objeto. Cuando el sistema de computación ejecuta el programa objeto puede encontrar algunos errores en los datos de entrada. Por ejemplo, es posible que al dispositivo de entrada se le ordene que lea cierto campo que se supone contiene datos numéricos, y en vez de esto se encuentra con datos alfabéticos. Esto hará que la computadora detenga la ejecución del programa objeto e imprima alguna advertencia de error. A estos mensajes se les denomina **diagnósticos del tiempo de ejecución**. Las instrucciones para transferencia de control defectuosas también causarán un **error del tiempo de ejecución**. Es posible llevar el sistema a un ciclo cerrado, es decir, el sistema ejecutará una secuencia de instrucciones y lo regresa a la primera instrucción de la secuencia. En teoría significa que la computadora ejecutará estas instrucciones para siempre; sin embargo, en la práctica, se fija un límite de tiempo para la ejecución del programa, y si éste no termina antes de que transcurra este tiempo, el sistema se detiene automáticamente e imprime un mensaje indicando porqué terminó la ejecución.

El programa fuente se almacena en memoria auxiliar, después de que se haya depurado completamente y se haya verificado que procesa los datos tal y como lo desea el programador; de esta manera se encuentra disponible para usarse en el momento que se requiera. Cuando se va a correr el programa con datos se extrae del almacenamiento, se compila y se coloca en la memoria principal en la unidad central de procesamiento como un programa objeto en lenguaje de máquina.

♦ **Documentación:** La documentación es la etapa final en la elaboración de un programa y tiene lugar después de que se ha almacenado y corregido. Documentar el programa se refiere a la recopilación, en un solo lugar, de toda aquella información de importancia que se usó para elaborar el programa.

En un solo lugar se recopilan las descripciones de todas las tareas, los algoritmos, la codificación, y así por el estilo. Si en una fecha posterior el programa revela algún error o tiene que ser modificado, esta documentación le facilitará a cualquier otro programador saber qué se ha hecho y porqué.

También se elabora un manual de corrida (manual del operador) que contiene todas las instrucciones que necesita el operador de la computadora para correr el programa, qué archivos de entrada se van a cargar, qué tipos de salida se deben preparar, etcétera.

1.4 SISTEMAS NUMÉRICOS:

Un sistema numérico en base **r** es un sistema que utiliza distintos símbolos para representar **r** dígitos. Los números se representan por hileras de símbolos de dígitos. Para determinar la cantidad que el número representa es necesario multiplicar cada dígito por una potencia entera de **r** y luego ejecutar la suma de todos los dígitos pesados. (Dígito pesado es aquél que tiene dos valores: uno intrínseco y otro posicional).

Por ejemplo, el **sistema numérico decimal**, que es el que utilizamos a diario, está en base 10 (r=10), lo cual significa que utiliza diez dígitos para representar todas las cantidades posibles. Los diez símbolos utilizados por dicho sistema son: **1,2, 3, 4, 5, 6, 7, 8, 9 y 0**. La hilera de dígitos **724.5** se utiliza para representar la cantidad:

$$\mathbf{7x10^2 + 2x10^1 + 4x10^0 + 5x10^{-1}}$$

El **sistema numérico binario** utiliza la base dos. Los dos dígitos que utiliza son el **0 y el 1**. **Por ejemplo**: La hilera de dígitos **101101** se utiliza para representar la cantidad:

$$\mathbf{1x2^5 + 0x2^4 + 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 45}$$

Para distinguir los números de diferentes bases, los dígitos serán encerrados entre paréntesis y la base del número será escrita como un subíndice. Por ejemplo, para mostrar la igualdad entre el 45 en base 10 y en base 2, escribiremos: $\mathbf{(101101)_2 = (45)_{10}}$

Además de los sistemas numéricos decimal y binario, **los sistemas octal (base 8) y hexadecimal (base 16) son importantes en el trabajo de computadoras digitales.** Los ocho símbolos o dígitos del sistema octal son: **0, 1, 2, 3, 4, 5, 6, 7,** Los dieciséis símbolos o dígitos del **sistema hexadecimal** son: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F**. Los últimos seis símbolos son desafortunadamente idénticos a las letras del alfabeto y pueden causar alguna confusión, a veces. Sin embargo, esto se utiliza por convención, y los símbolos A, B, C, D, E, F cuando se utilizan para representar dígitos hexadecimales corresponden a los números decimales 10, 11, 12, 13, 14, 15, respectivamente.

Un número en base **r** se puede convertir al familiar sistema decimal ejecutando la suma de sus dígitos pesados. Por ejemplo, el número $(736.4)_8$ para convertirlo a base 10 se hace como sigue:

$$(736.4)_8 = 7x8^2 + 3x8^1 + 6x8^0 + 4x8^{-1}$$

Y se ejecutan las operaciones en sistema decimal, esto es:

$$7x64 + 3x8 + 6x1 + 4/8 = (478,5)10$$

El equivalente número decimal del hexadecimal **F3** se obtiene efectuando los siguientes cálculos: $(F3)_{16} = Fx16^1 + 3x16^0 = 15x16 + 3x1 = (243)_{10}$

La conversión desde **decimal** a su equivalente representación en base **r** se ejecuta separando el número en su parte entera y su parte decimal y convirtiendo cada parte por separado. La conversión de un entero decimal a su equivalente en base **r** se ejecuta mediante divisiones sucesivas por la base **r** y se acumulan los residuos en orden inverso a como fueron apareciendo.

La conversión de la **parte decimal** a su equivalente en base **r** se obtiene mediante multiplicaciones sucesivas por la base **r** y se acumulan los dígitos enteros obtenidos en el orden de aparición.

El siguiente ejemplo ilustra estos procedimientos: La conversión del número decimal 41,6875 a binario se ejecuta separando primero la parte entera 41 y la parte decimal 0,6875 y dividiéndolos sucesivamente por la base (2).

ENTERO 41		FRACCION 0.6875
41 |		
20 | 1		0.6875 * 2 = 1.3750
10 | 0		0.3750 * 2 = 0.7500
5 | 0		0.7500 * 2 = 1.5000
2 | 1		0.5000 * 2 = 1.0000
1 | 0		(fracción cero)
0 | 1	(cociente cero)	$(0.6875)_{10} = (0.1011)_2$

$(41)_{10} = (101001)_2$

Por consiguiente $(41.6875)_{10} = (101001.1011)_2$

La parte entera se convierte dividiendo 41 por la base a la cual se va a trasladar (r = 2) y se obtiene un cociente de 20 y un residuo de 1. El cociente se divide de nuevo por la nueva base (r = 2) y se obtiene un nuevo cociente y un nuevo residuo. Este proceso se **repite hasta que el cociente sea cero**. El número equivalente en binario será entonces el residuo obtenido en el orden inverso de aparición.

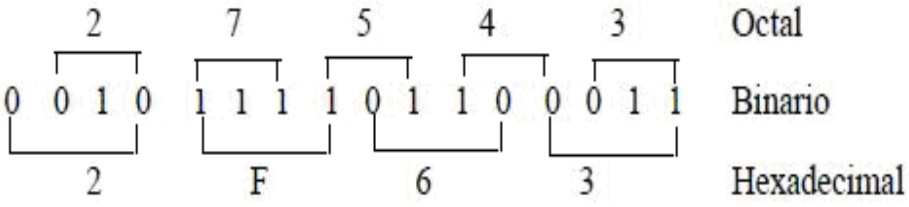
La parte fraccionaria se convierte multiplicándola por la nueva base (r=2) y se obtiene una parte entera y una fracción. La nueva fracción (sin la parte entera) se multiplica de nuevo por la nueva base (r=2) para obtener una nueva parte entera y una nueva fracción. Este proceso se **repite hasta que la parte fraccionaria llegue a ser cero** o hasta que el número de dígitos obtenidos dé la precisión deseada. La fracción binaria se obtiene de los dígitos enteros obtenidos al realizar cada multiplicación, colocándolos en el orden de aparición. Finalmente, se unen la parte entera y fraccionaria y se obtiene el resultado requerido.

1.4.1 REPRESENTACIÓN BINARIA, OCTAL Y HEXADECIMAL:

La conversión desde y hacia, de representación numérica **binaria**, **octal y hexadecimal**, juega un papel muy importante en computadores digitales. Debido a que dos al cubo es igual a ocho y dos a la cuarta es igual a dieciséis, cada dígito octal corresponde a tres dígitos binarios y cada dígito hexadecimal corresponde a cuatro dígitos binarios.

- La **conversión de binaria a octal** se ejecuta fácilmente dividiendo el número binario en grupos de tres dígitos de derecha a izquierda. A cada grupo de tres dígitos se le asigna el correspondiente dígito octal, y la hilera de dígitos así obtenida da la representación octal del número binario. Consideremos un registro de 16 bits. Físicamente uno puede pensar que el registro está compuesto por 16 celdas de almacenamiento binario, cada una de las cuales es capaz de almacenar un valor, cero o uno.

Supongamos que la configuración de bits almacenada en un registro es como la mostrada en la figura:



Debido a que un número binario consiste en una hilera de unos y ceros, el registro de 16 bits se puede utilizar para almacenar un número binario **desde uno hasta 2 a la 16 menos 1($1…2^{n-1}$)**. Para el particular ejemplo mostrado en la figura, el número almacenado en binario es equivalente al decimal **12131**.

Comenzando **desde la derecha**, partimos el registro en grupos de tres bits. A cada grupo de tres bits se le asigna su equivalente octal y se coloca en la parte superior del registro. La hilera de dígitos octales, así obtenida, representa el equivalente octal del número binario.

♦ La **conversión de binario a hexadecimal** es similar, excepto que los dígitos se agrupan en conjuntos de **cuatro dígitos**. El correspondiente dígito hexadecimal para cada grupo de cuatro dígitos se anota en la parte inferior del registro. La hilera de dígitos, así obtenida, representa el número hexadecimal equivalente al número binario.

El correspondiente dígito octal para cada grupo de tres bits se encuentra en la siguiente tabla:

Número octal	Octal codificado en binario	Decimal equivalente
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

La correspondencia entre dígitos hexadecimales y su equivalente en grupos de cuatro bits se encuentran en la siguiente tabla:

Número hexadecimal	Hexadecimal codificado en binario	Decimal equivalente
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

1.4.2 REPRESENTACIÓN DE DATOS:

La información binaria en computadoras digitales está almacenada en memoria, en registros procesadores. Los registros pueden contener datos o información de control. La información es un **bit o grupo de bits**, utilizado para especificar la secuencia de comandos de señales necesarias para manipular los datos de otros registros. Los datos son números u otra información codificada en binario operados para lograr los resultados computacionales requeridos.

Los tipos de datos encontrados en los registros de computadoras digitales se pueden clasificar en alguna de las siguientes categorías:
a) números, utilizados en cálculos aritméticos
b) letras del alfabeto, utilizadas en procesamiento de datos
c) otros símbolos, utilizados para propósitos específicos.

Todos los tipos de datos, excepto números binarios, se representan en registros de computadores en forma codificada en binario. Esto se debe a que los registros son construidos con elementos biestables, los cuales son dispositivos que sólo pueden almacenar unos o ceros. El sistema numérico binario es el sistema natural en computadoras digitales. Pero, algunas veces es conveniente emplear diferentes sistemas numéricos, especialmente el sistema numérico decimal, el cual es el ejecutado por los humanos para efectuar cálculos aritméticos.

El caso es que una hilera de unos y ceros almacenados en un registro puede representar un número binario, un número octal codificado en binario o un número hexadecimal codificado en binario. Los registros en computadoras digitales contienen muchos bits. Si especificamos su contenido por su representación en binario, requeriremos una hilera muy larga de dígitos binarios. Es más conveniente especificar el contenido de los registros por su equivalente octal o hexadecimal. El número de dígitos se reduce en una tercera parte en representación octal, y en una cuarta parte en representación hexadecimal. Por ejemplo, el número binario 111111111111 tiene doce dígitos. Si lo expresamos en forma octal utiliza cuatro dígitos (7777), en hexadecimal utiliza tres dígitos (FFF). Los computadores utilizan la representación octal o hexadecimal para especificar el contenido de los registros.

1.4.3 REPRESENTACIÓN ALFANUMÉRICA:

Muchas aplicaciones de computadores digitales requieren el manejo de datos que no son únicamente números, sino también letras del alfabeto y ciertos caracteres especiales. Un conjunto de caracteres alfanuméricos es un conjunto de elementos que incluye los diez dígitos decimales, las 26 letras del alfabeto y caracteres especiales tales como $, +, =, *, etc. Tal conjunto contiene entre 32 y 64 (si se incluyen únicamente letras mayúsculas), entre 64 y 128 (si se incluyen letras mayúsculas y minúsculas), o entre 128 y 256 (si se incluyen caracteres especiales).

En el primer caso se requerirán códigos binarios de seis bits, en el segundo caso, códigos binarios de siete bits y en el tercer caso códigos binarios de 8 bits. Es bueno aclarar que el computador siempre utilizará **un byte** (un conjunto de 8 bits) para almacenar datos alfanuméricos. En algunos sistemas de codificación utiliza los 8 bits para representar los caracteres; en otros, como el ASCII, utiliza 7 bits para representar los caracteres y el octavo como bit de paridad para controlar la transmisión de datos alfanuméricos.

El código binario estándar alfanumérico es el ASCII (American National Standard Code For Information Interchange), el cual utiliza ocho bits para codificar 256 caracteres. Los códigos binarios para letras mayúsculas, dígitos decimales y unos cuantos caracteres especiales se muestran en la siguiente tabla:

Carácter binario	Código	Carácter binario	Código
A	1000001	0	0110000
B	1000010	1	0110001
C	1000011	2	0110010
D	1000100	3	0110011
E	1000101	4	0110100
F	1000110	5	0110101
G	1000111	6	0110110
H	1001000	7	0110111
I	1001001	8	0111000
J	1001010	9	0111001
K	1001011		
L	1001100	Blanco	0100000
M	1001101		0101110
N	1001110	(	0101000
O	1001111	+	0101011
P	1010000	$	0100100
Q	1010001	*	0101010
R	1010010	)	0101001
S	1010011	-	0101101
T	1010100	/	0101111
U	1010101	,	0101100
V	1010110	=	0111101
W	1010111		
X	1011000		
Y	1011001		
Z	1011010		

Los códigos binarios juegan un papel muy importante en las operaciones en computadoras digitales. Los códigos deben estar en binario debido a que los registros sólo pueden guardar información binaria. Las operaciones especificadas para computadoras digitales deben tener en cuenta el significado de los bits almacenados en los registros, de tal manera que las operaciones se ejecuten sobre operandos del mismo tipo. Los códigos binarios se pueden asignar para cualquier conjunto de elementos: colores, notas musicales, piezas de ajedrez, etc.

Los códigos binarios se utilizan también para formular instrucciones que especifiquen información de control a la computadora. Aquí nos ocuparemos únicamente de la representación de datos.

EJERCICIOS PROPUESTOS

1. **Dado el número 1280_{10} ¿Cuál de las siguientes opciones correspondería a su representación en binario?**

A. 10100000000
B. 10011110110
C. 10110000000

2. **Dado el número 1070_8 ¿Cuál de las siguientes opciones correspondería a su representación en decimal?**

A. 670_{10}
B. 107_{10}
C. 345_{10}
D. 568_{10}

3. **Dado el número 3554_{10} ¿Cuál de las siguientes opciones correspondería a su representación en Hexadecimal?**

A. 12132_{16}
B. $DE2_{16}$
C. $D132_{16}$
D. $AB2_{16}$

4. **Dado el número 111100010 ¿Cuál de las siguientes opciones correspondería a su representación en Octal?**

A. 742_8
B. 722_8
C. 720_8

5. **Dado el número $3BC_{16}$ ¿Cuál de las siguientes opciones correspondería a su representación en Binario?**

A. **001110111100**
B. 0011101111000
C. 1110111100

RESUELVA:

1. **Convertir los siguientes números del sistema binario al decimal:**

A. 101100000
B. 1111101

2. Convierta los siguientes números del sistema octal al sistema decimal

A. 2630_8

B. 1731_8

3. Convierta los siguientes números del sistema decimal al sistema hexadecimal

A. 4586_{10}

B. 963_{10}

4. Convierta los siguientes números del sistema binario al sistema octal

A. 1111100001

B. 10111101010

5. Convierta los siguientes números del sistema hexadecimal al sistema binario

A. $8D3_{16}$

B. $E29_{16}$