

# Lógica de programación orientada a objetos

```
package Cap7_Ejer06;
public class Cuenta {
    private long numero;
    private int clave;
    private String nombre;
    private float saldo;

    public Cuenta(long num, int cla, String nom, float sal) {
        numero = num;
        clave = cla;
        nombre = nom;
        saldo = sal;
    }

    public void consultarSaldo() {
        System.out.println("SU SALDO ES: " + saldo);
    }

    public void consignarDinero(float valorConsignacion) {
        saldo += valorConsignacion;
        System.out.println("CONSIGNACION REALIZADA");
        consultarSaldo();
    }

    public void retirarDinero(float valorRetiro) {
        if (saldo >= valorRetiro)

```

ECOE  
EDICIONES



UNIVERSIDAD  
DE ANTIOQUIA  
1803

Efraín Oviedo Regino







# **Lógica de programación orientada a objetos**

---

Efraín M. Oviedo Regino

Oviedo Regino, Efraín Manuel

Lógica de programación orientada a objetos / Efraín M. Oviedo Regino. -- 1a. ed.

– Bogotá : Ecoe Ediciones : Universidad de Antioquia, 2015

444 p. – (Ingeniería y salud en el trabajo. Informática)

Incluye complemento virtual SIL (Sistema de Información en Línea)

www.ecoedediciones.com -- “Ana Isabel Oviedo Carrascal Coautora del capítulo 7

(orientación a objetos)” -- (p. leg.)

ISBN 978-958-771-136-3

1. Programación orientada a objetos (Computación)

2. Programación lógica 3. Algoritmos I. Título II. Serie

CDD: 005.131 ed. 20

CO-BoBN– a952157

---

**Colección:** *Ingeniería y salud en el trabajo*

**Área:** *Informática*

**ECOE**  
**EDICIONES**



**UNIVERSIDAD**  
**DE ANTIOQUIA**  
1803

© Efraín M. Oviedo Regino

© Ana Isabel Oviedo Carrascal  
Coautora del capítulo 7 (Orientación a  
objetos)

© Ecoe Ediciones Ltda.  
e-mail: info@ecoedediciones.com  
www.ecoedediciones.com  
Carrera 19 No. 63 C 32, Tel.: 2481449  
Bogotá, Colombia

**Primera edición:** Bogotá, enero de 2015

**Reimpresión:** Bogotá, junio de 2015

**ISBN:** 978-958-771-136-3

**e-ISBN:** 978-958-771-137-0

**Coordinación editorial:** Andrea del Pilar Sierra

**Corrección de estilo:** Juan Mikan

**Diseño y diagramación:** Diana Vanessa Gaviria

**Carátula:** Wilson Marulanda

**Impresión:** La Imprenta

Calle 77 No. 27 A - 39

Prohibida la reproducción total o parcial por cualquier medio sin la autorización escrita del titular de los derechos patrimoniales.

## ***Dedicado a:***

*Estas personitas que alegran la vida de  
Carmen Elena y la mía*

*Samuel David Ríos  
María Isabel Oviedo  
Emilia Ríos*

*y los demás nietecitos que han de venir*



# Contenido

<b>Presentación</b> .....	XI
<b>Capítulo 1: Generalidades del computador</b> .....	1
1.1 Evolución del computador .....	1
1.2 Descripción de un computador .....	10
1.3 Clasificación de los computadores.....	11
1.4 Estructura lógica de un computador .....	13
1.4.1 La unidad de entrada .....	13
1.4.2 La unidad de salida .....	15
1.4.3 Unidad de control .....	16
1.4.4 Unidad aritmética y lógica .....	16
1.4.5 Unidad de memoria.....	16
1.6 Definiciones básicas .....	19
1.7 Sistemas de numeración .....	23
1.7.1 Paso de una base cualquiera $r$ a decimal .....	24
1.7.2 Paso de decimal a una base cualquiera $r$ .....	25
1.8 Representación de datos .....	34
1.8.1 Representación alfanumérica .....	35
1.8.2 Representación numérica.....	37
1.9 Clases de información .....	48
1.10 Aspectos importantes para tener en cuenta .....	50
1.11 Ejercicios propuestos .....	51
<b>Capítulo 2: Conceptos básicos sobre algoritmos</b> .....	53
2.1 La lógica computacional.....	53
2.2 El algoritmo .....	54
2.2.1 Algoritmos cualitativos .....	55
2.2.2 Algoritmos cuantitativos .....	57
2.3 Pasos para la solución de un problema a través del computador .....	58
2.3.1 Definición del problema .....	59
2.3.2 Análisis del problema .....	60
2.3.3 Crear el algoritmo.....	60
2.3.4 Prueba de escritorio .....	61
2.3.5 Codificación.....	61
2.3.6 Transcripción .....	62
2.3.7 Compilación .....	62



2.3.8 Ejecución .....	63
2.3.9 Documentación externa.....	63
2.4 Representación de los algoritmos mediante pseudocódigo.....	68
2.5 Estructuras de la programación.....	69
2.6 Aspectos para tener en cuenta .....	69
2.7 Ejercicios propuestos .....	69
<b>Capítulo 3: Estructura de control secuencial.....</b>	<b>71</b>
3.1 Representación.....	71
3.2 Instrucción de asignación .....	72
3.2.1 La expresión.....	72
3.3 Instrucción de entrada de datos.....	77
3.4 Instrucción de salida de datos .....	78
3.5 Aspectos para tener en cuenta .....	84
3.6 Ejercicios propuestos .....	85
<b>Capítulo 4: Estructuras de control selectivas .....</b>	<b>90</b>
4.1 Estructura de decisión lógica.....	90
4.1.1 Decisión lógica simple .....	90
4.1.2 Decisión doble.....	92
4.1.3 Decisiones compuestas o anidadas .....	93
4.2 Selección múltiple o estructura caso .....	104
4.3 Aspectos para tener en cuenta .....	111
4.4 Ejercicios propuestos .....	111
<b>Capítulo 5: Estructura de control repetitivas .....</b>	<b>116</b>
5.1 Estructura de control MIENTRAS.....	116
5.1.1 Variables tipo contador .....	117
5.1.2 Variables tipo acumulador.....	118
5.1.3 Esquema cuantitativo .....	123
5.1.4 Esquema cualitativo .....	131
5.1.5 Variables tipo bandera .....	138
5.1.6 Ruptura de ciclos.....	143
5.2 Estructura de control PARA .....	153
5.3 Estructura de control HACER MIENTRAS.....	160
5.4 Aspectos para tener en cuenta .....	169
5.5 Ejercicios propuestos .....	170
<b>Capítulo 6: Métodos .....</b>	<b>179</b>
6.1 Clasificación según la forma en que se invoca el método .....	179
6.2 Invocación de métodos .....	180

6.3	Parámetros: comunicación entre métodos .....	181
6.4	Clases de métodos según el retorno .....	182
6.4.1	Métodos que no retornan o funciones tipo <i>void</i> .....	182
6.4.2	Métodos que retornan .....	192
6.5	Aspectos para tener en cuenta .....	203
6.6	Ejercicios propuestos .....	203
<b>Capítulo 7: Orientación a objetos .....</b>		<b>205</b>
7.1	Definición de clase y objeto .....	205
7.2	Propiedades de la programación orientada a objetos .....	208
7.3	Abstracción .....	208
7.4	Encapsulamiento y ocultamiento de la información .....	211
7.5	Especificadores o modificadores de acceso .....	212
7.6	Formato de una clase .....	214
7.7	Constructores y destructores .....	215
7.8	Creación de objetos .....	218
7.9	Definición del método principal .....	219
7.10	Pasos para solucionar un problema mediante POO .....	220
7.11	Métodos accesores .....	232
7.12	Sobrecarga de métodos .....	239
7.13	Herencia y reutilización .....	243
7.13.1	Tipos de herencia .....	251
7.14	Polimorfismo .....	257
7.15	Otros modificadores de acceso .....	264
7.16	Clases abstractas .....	265
7.17	Aspectos para tener en cuenta .....	272
7.18	Ejercicios propuestos .....	273
<b>Capítulo 8: Arreglos .....</b>		<b>280</b>
8.1	Arreglos de una dimensión o vectores .....	280
8.1.1	Lectura de un vector .....	283
8.1.2	Imprimir los elementos de un vector que contenga n elementos .....	284
8.1.3	Búsqueda secuencial .....	288
8.1.4	Ordenamiento .....	288
8.1.5	Búsqueda binaria .....	293
8.1.6	Inserción .....	293
8.1.7	Borrado .....	294
8.2	Arreglo de objetos .....	320
8.3	Arreglos de dos dimensiones o matrices .....	327
8.3.1	Clases de matrices .....	329

8.3.2 Principales operaciones con matrices.....	332
8.4 Arreglos multidimensionales.....	370
8.5 Aspectos para tener en cuenta .....	378
8.6 Ejercicios propuestos .....	379
<b>Capítulo 8: Archivos .....</b>	<b>386</b>
9.1 Clasificación de los archivos según su uso .....	386
9.2 Clasificación de los archivos según los datos que almacenan .....	386
9.3 Operaciones básicas con archivos.....	387
9.4 Archivos de texto.....	388
9.5 Archivos binarios.....	392
9.5.1 Almacenamiento de archivos binarios y su acceso a la información .....	394
9.5.2 Operaciones adicionales con archivos binarios.....	396
9.5.3 Forma de operación de un archivo binario.....	397
9.6 Aspectos para tener en cuenta .....	414
9.7 Ejercicios propuestos .....	415
<b>Índice temático.....</b>	<b>419</b>
<b>Bibliografía y cibergrafía.....</b>	<b>429</b>
<b>Acerca del autor .....</b>	<b>431</b>



Al final del libro está ubicado el código para que pueda acceder al Sistema de información en Línea – SIL, donde encontrará ejercicios de los problemas propuestos en el libro y los siguientes videos:

Video 1. Generalidades del computador

Video 1.1 Sistemas numéricos

Video 2. Generalidades Algoritmos

Video 3. Estructura Secuencial

Video 4. Estructuras Selectivas

Video 5. Estructuras Repetitivas

Video 6. Métodos

Video 7. Programación Objetual

Video 8. Arreglos

Video 9. Archivos

# Presentación

Algunos de los vertiginosos avances que han sufrido las ciencias de la computación —reflejados no sólo en las ramas de la Ingeniería de Sistemas, Informática y carreras afines y del aprendizaje autodidacta— se han basado en lo fundamental en el paradigma de la programación orientada a objetos y desde luego en la programación a través de internet; esto me ha llevado a presentar esta nueva obra titulada: *Lógica de programación con orientación a objetos*, esperando sea de gran ayuda a todo aquel que incurse por primera vez en la solución de problemas a través de un computador, donde el razonamiento lógico debe predominar para que pueda alcanzar soluciones correctas.

Razonamiento es pensar, de tal manera que el razonamiento lógico depende del individuo, es algo que no se puede enseñar en forma directa. Esto solo lo adquiere usted mismo solucionando muchos ejercicios. Recuerde que nadie puede pensar por usted; por ello el libro está diseñado de modo que usted inicie solucionando ejercicios sencillos y en la medida que avance encontrará nuevas estructuras que aumentarán gradualmente la complejidad de los algoritmos. Realmente, hacer algoritmos es muy sencillo si sigue los pasos que están descritos en el libro; es la razón por la cual le aconsejo trabajar muy bien los capítulos iniciales, estudiando los ejercicios resueltos y elaborando ojalá todos los ejercicios propuestos que le permitan crear ese pensamiento lógico. Para abordar la solución de problemas que se van a resolver mediante un computador, usted debe tener en cuenta que todo problema se soluciona de una manera distinta y que, a diferencia de otras ramas del saber, no tenemos fórmulas que se puedan generalizar: todo problema tiene diversas soluciones algorítmicas.

Los contenidos del libro no presuponen conocimiento previo de informática ni de programación, puesto que el desarrollo de los temas se hace en forma incremental, comenzando con ejercicios muy simples. Es usted quien debe poner disciplina de su parte, acatando las sugerencias que el libro le va dando en la medida en que avanza en él. Por ello el texto tiene 87 ejercicios resueltos, en cuya solución se usan las etapas que se aconsejan para problemas de este tipo. También tiene 204 ejercicios propuestos distribuidos en cada uno de sus nueve capítulos. Los 87 ejercicios resueltos los encontrará usted representados en los lenguajes C++, Java (versión 1.7) y C# (versión 2012). También se le han agregado al texto 10 videos que explican las partes más importantes de cada capítulo.

Por último, doy grandes agradecimientos a la PhD Ana Isabel Oviedo Carrascal, docente de la Universidad Pontificia Bolivariana, sede Medellín, quien no solo actuó como correctora, sino que también me ayudó a escribir el capítulo de programación orientada a objetos, capítulo 7 de este libro.

# Capítulo 1

---

# Generalidades del computador<sup>1</sup>

---

Antes de introducirnos en el oficio de la programación presentamos conceptos generales sobre el computador que nos ayudarán a entender su funcionamiento e historia.

## 1.1 Evolución del computador

La necesidad de hacer cálculos por el hombre no es de los últimos tiempos; en realidad, no se tiene una fecha exacta determinada de cuándo se planteó el hombre la necesidad de agilizar a través de una máquina las operaciones que se le presentaban. Uno de los primeros dispositivos mecánicos para llevar cuentas fue el *ábaco*, cuya aparición data de miles de años, en el cercano Oriente (3000 años a.C. aproximadamente). Cuando esta técnica se conoció aparecieron varias versiones, tales como el *ábaco chino* y el *sorobán japonés*.

En 1642, el francés Blaise Pascal desarrolló una calculadora que era operada por un conjunto de discos, unidos en engranajes, que tenían los números del cero al nueve en forma de círculo, de tal forma que cuando un disco daba

---

<sup>1</sup> Ver video 1: Generalidades del computador en el complemento SIL de [www.ecoediciones.com](http://www.ecoediciones.com)

una vuelta, automáticamente el disco de la izquierda avanzaba un dígito. Los indicadores sobre los discos mostraban las respuestas correctas, pero esta calculadora solo servía para sumar.

Uno de los grandes pensadores de los siglos XVII y XVIII, a quien se le conoce como “el último genio universal”, Gottfried Wilhelm Leibniz, creó en 1694 una máquina que multiplicaba y dividía directamente. El dispositivo que utilizó Leibniz, en forma de rueda escalonada, aún se usa en algunas calculadoras.

Los pensadores Newton, Pascal y Leibniz iniciaron una búsqueda del cálculo computarizado mucho antes que se desarrollaran las tecnologías de sus tiempos, trabajando de una manera aislada, pero ninguno de sus esfuerzos individuales para conseguir una máquina de procesos de datos era confiable.

El desarrollo e interés por la navegación, la astronomía y el hilado textil da pie a que se creen máquinas para la ayuda en la solución de cálculos complejos y para ingresar patrones a través de tarjetas a los telares. Joseph Marie Jacquard diseñó en 1801 las tarjetas perforadas para controlar una máquina de hilado textil, en donde cada línea de tejido se presentaba en una tarjeta perforada como patrón y este se tejía automáticamente. La entrada de datos a través de tarjetas perforadas no solamente fue usada en la parte textil, sino también en la entrada de información en aplicaciones de tipo comercial y científico en la segunda mitad del siglo XIX y en la primera mitad del siglo XX. Charles Babbage desarrolló en 1822 un proyecto denominado máquina diferencial, la cual era capaz de producir tablas logarítmicas de seis cifras decimales de precisión. Animado por los resultados obtenidos, Babbage continuó sus experimentos, pero el poco desarrollo de la tecnología de la época le impidió crear una máquina analítica.

El norteamericano Hernán Hollerith, cuando nació la necesidad de crear máquinas programables, retomó el trabajo realizado por Babbage desarrollando un mecanismo basado en tarjetas perforadas, cuya primera utilización exitosa se llevó a cabo en 1886 durante la tabulación del censo de población de 1880 en los Estados Unidos. Este equipo significó un gran adelanto en el campo de los computadores.

Las guerras han sido el motor para el desarrollo de los computadores, en especial a partir de la Segunda Guerra Mundial, que da el mayor impulso y fuerza para el desarrollo definitivo de los computadores. Durante esta época se desarrollaron los computadores COLOSSUS, ENIAC y MARK1.

El COLOSSUS fue construido por Alan Turing, T. Flowers y M. Newman (1943) en un laboratorio británico. Estaba hecho de válvulas de vacío y no tenía dispositivos electromecánicos.

Simultáneamente técnicos de la IBM, empresa fundada por Hollerith, la Universidad de Harvard y la Universidad de Pennsylvania, construyeron el MARK 1 y el ENIAC, cuya utilización estaba relacionada con los misiles guiados de uso militar.

El ENIAC (Electronic Numerical Integrator And Computer) es considerado como el primer computador electrónico de propósito general y es derivado del computador alemán Z3, creado por John Presper Eckert y Jhon William Mauchly. Era totalmente digital y fue presentado al público el 15 de febrero de 1946, aunque su diseño fue anterior a esta fecha (su construcción duró 30 meses). Para que pudiera operar necesitaba un área de 167 metros cuadrados con 17468 tubos al vacío y que se reconectaran constantemente los cables que lo componían. Los datos eran recibidos a través de una cinta perforada y leída por la unidad de control, y su única unidad eléctrica era el motor, con una frecuencia de 1 Hz. Esta máquina, que pesaba alrededor de 27 toneladas y tenía dimensiones de 2,4 x 0.9 x 30 metros, utilizaba 500 conmutadores electromagnéticos. Cuando su sistema de operación requería de modificaciones, tardaba semanas para volverse a instalar manualmente. Contaba con una velocidad aproximada de miles de multiplicaciones por minuto, pero su procesador debía ser modificado manualmente. A diferencia de nuestros computadores actuales, que operan bajo el sistema numérico binario, el ENIAC operaba con el sistema numérico decimal. Su construcción marcaría el surgimiento de la primera generación de computadores. Funcionó hasta el 2 de octubre de 1955.

En 1944 la Universidad de Manchester desarrolló el computador MARK1, que tenía 15.5 metros de largo, 2.40 metros de alto y unos 60 centímetros de ancho. Pesaba cinco toneladas y constaba de un complejo de 78 máquinas sumadoras y calculadoras, conectadas con 800 kilómetros de cables. Este computador representaba un avance con respecto al ENIAC ya que, a pesar de que las instrucciones se daban en cintas de papel, una vez que la máquina ejecutaba la primera instrucción no requería de la intervención humana. Sin embargo, se presentaban restricciones a causa de su lentitud y dificultades en su operación.

Para la misma época, John Von Neumann propuso que los programas se almacenaran en forma digital en la memoria del computador, al igual que los



datos, y reemplazó la aritmética decimal por la binaria. Este diseño es conocido como arquitectura de Von Neumann y ha sido la base para el computador digital.

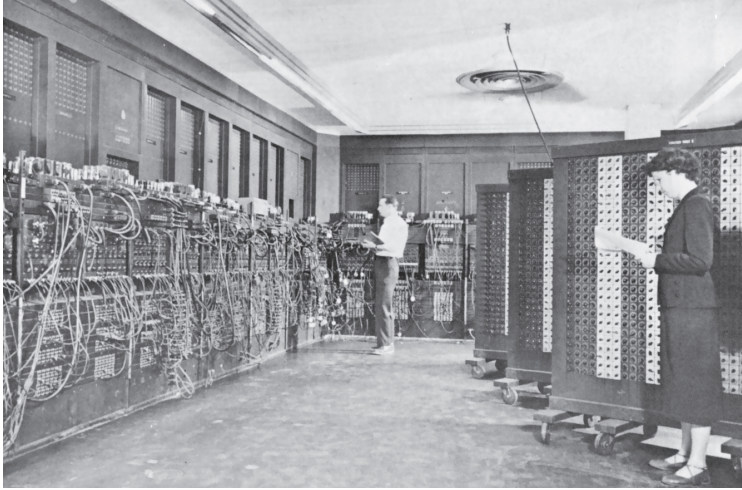
Fue así como John W. Mauchly y J.P. Eckert, casi en la misma época, desarrollaron una máquina que denominaron EDVAC, que operaba con números binarios y podía almacenar instrucciones escritas en ceros y unos. El control del computador se realizaba mediante el alambrado de cables removibles o cinta de papel y, una vez que el procesamiento de los datos había sido iniciado, ninguna modificación podía efectuarse a pesar de que surgiera la necesidad de hacerlo.

No tardó mucho para que las primeras máquinas de computación, que fueron desarrolladas para propósitos militares, incursionaran en los ámbitos comerciales e industriales gracias al nacimiento del transistor, que fue desarrollado por los laboratorios de Bell Telephone y vino a reemplazar los tubos al vacío.

*El primer computador* que se produjo para efectos comerciales fue el UNIVAC 1 (Universal Automatic Computer), construido por Eckert y Mauchly como producto del avance tecnológico que ya tenía unos alcances vertiginosos, reduciendo costos y tamaño. Este computador fue instalado en el departamento de censos de los Estados Unidos en 1951 y contenía varias de las características de los computadores actuales. El UNIVAC fue entonces el inicio de la carrera en la creación de nuevos equipos de cómputo en forma acelerada.

Las máquinas construidas entre los años 1945 y 1958 constituyen la primera generación de computadores. Entre sus particularidades se puede señalar que generaban un alto consumo de energía y calor intenso; además, necesitaban muchos analistas programadores, codificadores y personal de datos y de mantenimiento. La programación se hacía directamente en el lenguaje de la máquina, o sea, representando la información con unos y ceros.

Estos computadores se caracterizaban por utilizar tubos al vacío en los componentes básicos de sus circuitos, los cuales desprendían bastante calor y tenían poco tiempo de vida útil. Las máquinas eran grandes y pesadas, almacenaban la información —suministrada a través de tarjetas perforadas— en tambores magnéticos, presentaban fallas continuas en su funcionamiento y tenían alto costo por el calor que desprendían, que hacía necesaria la instalación de aire acondicionado. A esta generación, desde luego, pertenecen los computadores ENIAC, UNIVAC y otros.



**Imagen 1.1. Foto de ENIAC.**

U. S. Army (Autor). (1947-1955). Eniac.jpg [Fotografía]. Recuperado el 12 de marzo de 2014, de: <http://commons.wikimedia.org/wiki/File:Eniac.jpg>

*La segunda generación de computadores (1959 a 1964)* se dio por el gran progreso de la industria e inició cuando los bulbos fueron sustituidos por transistores. Los avances en equipos periféricos también fueron notables: impresoras cada vez más rápidas, mejores lectoras de tarjetas y de cinta perforada y —sobre todo— bobinas de cinta magnética capaces de memorizar y de volver a leer datos en número ilimitado.

Los transistores permitieron aumentar la confiabilidad y velocidad operativa de los nuevos equipos, cuyo tamaño y costo ya eran más reducidos. Asimismo, tenían redes de núcleos magnéticos, los cuales reemplazaron a los tambores giratorios para el almacenamiento de la información. También hubo avance en los lenguajes de programación: se pasó del binario a lenguajes más fáciles como el de alto nivel. Los datos e instrucciones ya no se introducían a la máquina solo a través de tarjetas perforadas, sino también por medio de cintas perforadas y de tableros. La velocidad en los procesos se medía en menos de segundos, las memorias internas se construyeron en núcleos de ferrita, y el almacenamiento de información se hacía en cintas y discos magnéticos. Desde luego, el avance tecnológico hizo que se mejoraran los dispositivos de entrada y salida, ya que le dio mucho impulso a las aplicaciones de tipo comercial.

En esta generación proliferaron diferentes modelos de computadores, lo que permitió incorporar su uso en el área de los negocios. Con estos fines se desarrolló el lenguaje de programación COBOL y se incrementó la utilización del

lenguaje FORTRAN. Se crearon entonces los ensambladores, que utilizaban un código nemotécnico para representar las instrucciones.

La creación del lenguaje COBOL permitió que en esta época las empresas comenzaran a utilizar los computadores en el almacenamiento de registros tales como nómina, presupuestos, inventarios, contabilidad, etcétera para manipularlos en las aplicaciones comerciales. Dentro de los computadores de esta generación están: Philco 212, UNIVAC M460, IBM 7090 y NCR 315.



**Imagen 1.2. Foto de IBM 7090.** Nasa (Autor). (1961).

IBM 7090 computer.jpg [Fotografía]. Recuperado el 25 de marzo de 2014, de: [http://commons.wikimedia.org/wiki/File:IBM\\_7090\\_computer.jpg?uselang=es](http://commons.wikimedia.org/wiki/File:IBM_7090_computer.jpg?uselang=es)

*La tercera generación* comprende los computadores creados desde 1965 hasta 1970, cuando los avances tecnológicos fueron (y continúan siendo) sorprendentes. Los transistores se sustituyeron por los circuitos integrados (IC). La base de creación para esta nueva etapa fueron los circuitos integrados monolíticos (pastillas de silicio) y la miniaturización de la electrónica, lo que permitió que los computadores comenzaran a ser cada vez más pequeños, más eficientes y más rápidos y que el calor que emanaban fuera menor.

Los circuitos integrados permitieron la flexibilidad de los programas y se buscó llegar a modelos estándares. Las máquinas de esta generación (IBM 360, IBM370, POP6) fueron fabricadas con la característica de que las pertenecientes a una misma serie eran compatibles entre sí, facilitando su uso; además, eran utilizables tanto para aplicaciones de tipo científico como comercial. A medida que estas máquinas se fueron perfeccionando surgieron la

multiprogramación, el multiprocesamiento, las comunicaciones de datos, otros lenguajes (BASIC y PL1) y los paquetes especializados de programación. Cabe destacar, por ejemplo, que en esta época nacieron las calculadoras de bolsillo y los microcomputadores.

Otras características de los computadores de esta generación son: menor consumo de energía, ampliación de las aplicaciones, orientándose a todas las esferas del saber, compatibilidad con otros sistemas operativos, inclusión de millones de circuitos en placas de silicio muy pequeñas (chip), nacimiento del teleproceso y aumento de la confiabilidad.

Al principio de los años 90 surgieron los computadores personales. Asimismo, las estaciones de trabajo se aumentaron y fueron reemplazadas por computadores de mayor velocidad y menor tamaño, que se usaron como terminales o estaciones de trabajo en red. De este modo nació un concepto nuevo, las redes distribuidas, lo que hoy se conoce como cliente-servidor.



**Imagen 1.3. Foto de IBM 360.** Don DeBold (Autor). (2011). IBM 360 Computer [Fotografía]. Recuperado el 25 de marzo de 2014, de: <http://www.flickr.com/photos/ddebold/5900607306/>

La cuarta generación se caracteriza por la aparición de los microprocesadores —circuitos integrados de alta velocidad y densidad— como un gran avance de la microelectrónica. Esta generación abarca los computadores construidos entre 1971 y 1981, cuando se logró una reducción de los circuitos integrados incorporando miles de componentes con un espacio menor a una micra. En esta época apareció el lenguaje PASCAL como uno de los más poderosos por ser de aplicación general e incluir los conceptos introductorios de lo que hoy

se conoce como programación estructurada, que se constituye como pieza principal para el desarrollo de los métodos en la programación orientada a objetos.

PASCAL fue uno de los lenguajes de programación creados para el desarrollo de la parte académica. Era muy sencillo de manejar y tenía muy pocas complicaciones para el programador; infortunadamente, cuando se dio inicio a la programación orientada a objetos salieron adelante otros lenguajes que hoy marcan la pauta en este paradigma de programación.

Se incorporaron dos grandes mejoras en el inicio de esta nueva generación: por un lado, se reemplazaron las memorias con núcleos magnéticos por los chips de silicio y, por otro, surgió el concepto de miniaturización en los circuitos. Desde luego, la velocidad de los computadores estuvo ligada con todos estos nuevos avances, reduciendo el tiempo de respuesta. El uso de los computadores se hizo común en todas las esferas, se crearon memorias más rápidas, nacieron las bases de datos y se le dio un gran impulso al microproceso.



**Imagen 1.4.** Foto de IBM PC. Skies (Autor). (2007).

Televideo925Terminal adjusted.jpg [Fotografía]. Recuperado el 25 de marzo de 2014, de: [http://commons.wikimedia.org/wiki/File:Televideo925Terminal\\_adjusted.jpg](http://commons.wikimedia.org/wiki/File:Televideo925Terminal_adjusted.jpg)

En *la quinta generación de computadores (1982-1989)* los computadores se caracterizan por tener muchos microprocesadores interconectados trabajando en paralelo, el reconocimiento de la voz e imágenes y la inteligencia artificial: todo esto como producto de un proyecto de investigación del Ministerio de Industria y Comercio de Japón en el año de 1982, enmarcado desde luego en el continuo avance de la microelectrónica y computación, los altos alcances en la inteligencia artificial, los sistemas expertos, las redes neuronales, la

teoría de robótica, la teoría de casos, los algoritmos genéticos, las fibras ópticas y las telecomunicaciones. También es necesario mencionar la creación en la década del 70 del primer supercomputador diseñado por Seymour Cray.

Una de las pocas distinciones que se puede hacer entre la cuarta y quinta generación es el proceso en paralelo, es decir, el hecho de que estos computadores puedan trabajar sincronizadamente con otros microcomputadores, dado que la memoria es adecuada para que pueda atender en forma rápida procesos simultáneos, particionada en módulos compartidos que asignan tareas a cada uno de los procesadores.

Otros aspectos relevantes del proyecto de quinta generación es que estas máquinas pueden reconocer voz e imágenes, la comunicación se hace a través de lenguaje natural y hay un proceso de toma de decisiones en la medida en que hacen “aprendizaje”. Además, el almacenamiento de información se hace en dispositivos ópticos con decenas de gigabytes.

La inteligencia artificial o los programas que pueden aprender a partir de otros programas dieron lugar a la creación de sistemas de reconocimiento de formas tridimensionales, la teoría de robótica, la teoría de juegos y los traductores de lenguajes. Este paradigma simula algunas de las reglas mentales mediante las cuales se pueden obtener nuevos programas, que cambian la forma de interactuar con la información por un lenguaje de programación lógica.



**Imagen 1.5. Foto de robot con inteligencia artificial.** Rico Shen (autor). (2008). [Fotografía]. Recuperado el 25 de marzo de 2014, de: [http://commons.wikimedia.org/wiki/File:2008\\_Taipei\\_IT\\_Month\\_Day2\\_Taipei\\_City\\_Government\\_Intelligent\\_Housekeeping\\_Robot.jpg](http://commons.wikimedia.org/wiki/File:2008_Taipei_IT_Month_Day2_Taipei_City_Government_Intelligent_Housekeeping_Robot.jpg)

La *sexta generación* de computadores va desde los *años 90 en adelante*. Estas máquinas, creadas por el mismo proceso anterior, cuentan con arquitecturas combinadas paralelo- vectoriales, cientos de microprocesadores en funcionamiento y unidades aritméticas y lógicas capaces de realizar un billón de operaciones de punto flotante por segundo, utilizando el desarrollo de las redes de comunicación, la fibra óptica y satélites con anchos de banda exorbitantes. Las tecnologías de esta generación (algunas ya desarrolladas y otras en proceso) tienen en común el avance en la inteligencia artificial distribuida, sistemas difusos, teoría del caos, transistores ópticos y heliografía, elementos que les permiten comunicarse con los computadores en un lenguaje más cotidiano y no a través de lenguajes de control.



**Imagen 1.6. Foto de computador de sexta generación.** Tom (Autor). (2010). Computer sheet [Fotografía]. Recuperado el 14 de marzo de 2014, de: <http://www.flickr.com/photos/turkletom/4279242364/>

## 1.2 Descripción de un computador

Un computador es una máquina capaz de efectuar una secuencia de operaciones mediante un programa apropiado, de tal manera que se haga una transformación (proceso) sobre un conjunto de datos que le son suministrados y se obtenga un nuevo conjunto de datos, llamados resultados. Todo esto lo hace con alta velocidad y gran capacidad de almacenamiento. También se puede describir como una calculadora electrónica veloz que acepta información digitalizada de entrada, la procesa de acuerdo con un programa que reside en su memoria y produce resultados de salida.

Estamos en un momento de transformación grandísima en la sociedad, enmarcado inicialmente por la industrialización y actualmente por la informática y

por las telecomunicaciones. El uso de la información cada vez más útil afecta de manera directa a la sociedad, haciendo cada vez más grande la brecha entre ricos y pobres.

El computador es una máquina que nos permite hacer toda una gama de tareas diferentes, por lo que ha influido en muchos aspectos de nuestras vidas. Dentro de esas tareas podemos mencionar:

- 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 en 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 un computador a través del teléfono.
- Proveer información sobre los productos en el mercado.
- Manipular electrodomésticos, teléfonos inteligentes y tableta.
- Comunicación a través de las redes, etc.

Mucho se habla de que el computador está influyendo en nuestra privacidad y sustituyendo mano de obra, creando desempleo. Estos aspectos no pueden discutirse sin conocer a fondo lo que es un computador. Solo a medida que se avanza en su conocimiento es posible emitir un concepto inteligente. Hay que tener en cuenta además que un computador no puede hacer algo a menos que el ser humano le diga qué hacer. Debemos controlar el computador; no él a nosotros.

### 1.3 Clasificación de los computadores

Según la clase de información que maneja un computador, estos se pueden clasificar en:

- **Computadores digitales:** Son aquellos que aceptan y procesan datos discretos que han sido convertidos al lenguaje de la máquina. Trabajan directamente contando números (dígitos) que representan cifras, letras u otros símbolos especiales. Así como los relojes digitales cuentan los segundos y minutos en una hora, los procesadores digitales cuentan



valores discretos para alcanzar los resultados deseados. Los computadores digitales son usados en el proceso de datos, como puede ser el proceso de una contabilidad. Los computadores digitales pueden ser:

**De propósitos generales:** Son computadores que se programan para múltiples tareas o aplicaciones como cálculos matemáticos, estadísticos y cualquier proceso del área comercial (nómina, presupuestos, inventarios, etc.). Pueden ser computadores grandes *mainframes* o minicomputadores.

**De propósitos especiales:** Están dedicadas a un solo propósito o tarea: por ejemplo, dar informes de tiempo, monitorear desastres naturales, hacer lecturas de velocidad o de gasolina, controlar carros a control remoto, medir la hora en relojes digitales, medir tiempo en hornos eléctricos; en general, todas las tareas personalizadas.

- **Computadores analógicos:** Este tipo de computadores aceptan y procesan señales continuas (por ejemplo, las fluctuaciones de voltaje o de frecuencias) y las registran con un alto grado de precisión. Son usados para controlar procesos con alta precisión, con un valor correcto que puede ser hasta de 0,1%. Se dedican al control físico de actividades como las de un ensamble automatizado o las de un sistema de control de temperatura. El termostato es el computador análogo más sencillo.
- **Computadores híbridos (analógico-digitales):** Son computadores digitales cuyo fin es procesar informaciones análogas que previamente han sido convertidas a ceros y unos (información binaria). Son una combinación de los dos anteriores: por ejemplo, en la unidad de control interactivo de un hospital los dispositivos analógicos pueden medir el funcionamiento del corazón de un paciente, la temperatura y los signos vitales; estas medidas pueden ser convertidas a números y enviadas a un componente digital del sistema, que es usado para controlar los signos vitales del paciente y enviar una señal a la estación de las enfermeras cuando sean detectadas lecturas anormales. Podemos decir que su utilización se extiende al control de procesos y en el área robótica.

Tanto los computadores híbridos como los analógicos son importantísimos, pero los más usuales son los computadores digitales.

Como ya vimos anteriormente, de acuerdo con el avance tecnológico se pueden establecer categorías de computadores así:

- **Primera generación:** Usaban tubos al vacío.
- **Segunda generación:** Pasan de tubos al vacío a transistores.
- **Tercera generación:** Los transistores se sustituyeron por los circuitos integrados (IC).

- **Cuarta generación:** Los circuitos integrados son de más alta velocidad y las memorias se construyen a través de núcleos magnéticos. El salto a esta generación lo da el nacimiento de los microprocesadores.
- **Quinta generación:** Computadores con muchos microprocesadores interconectados que trabajan en paralelo, reconocen voz e imágenes y cuentan con inteligencia artificial.
- **Sexta generación:** Computadores con arquitectura combinada paralelo-vectorial, con muchos microprocesadores trabajando al mismo tiempo y cada vez con más cantidad de almacenamiento y altísima velocidad.

## 1.4 Estructura lógica de un computador

El computador tiene una variedad de componentes que actúan junto a programas para realizar cálculos, organizar datos y comunicarse con otros computadores. Todas las partes de un sistema de computación operan bajo el control de una de ellas: la unidad de control.

Aunque un sistema de computación está compuesto por muchos dispositivos, cada sistema tiene cinco componentes básicos. En la Imagen 1.7 se muestran estos componentes y sus relaciones entre sí.

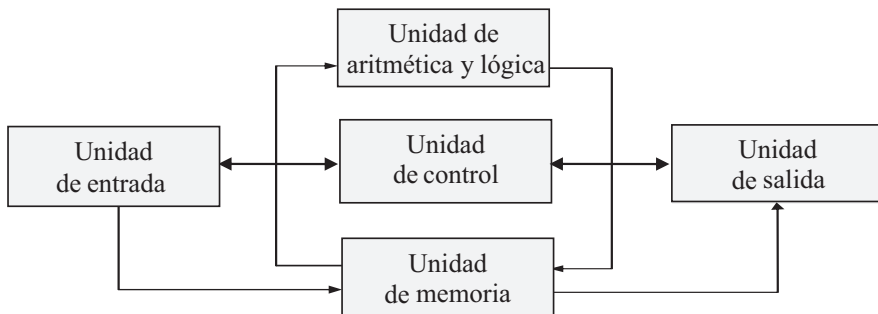


Imagen 1.7. Unidades básicas de un computador.

### 1.4.1 La unidad de entrada

Alimenta la memoria con datos e instrucciones, es decir, pasa información desde un medio externo de entrada de datos a la memoria principal del computador, donde es almacenada en forma temporal. Esto significa que la memoria principal es volátil y que no podemos guardar en ella información en forma permanente. La unidad de memoria almacena instrucciones y datos en grupos de bits que tienen una única dirección de memoria y su tamaño puede variar.

Independientemente de cómo se introduzcan los datos, estos se transforman en impulsos eléctricos que se transfieren a direcciones predeterminadas dentro de la memoria. La entrada se divide básicamente en dos tipos: *instrucciones para el proceso de datos* y *los datos por procesar*.

Como ejemplos de medios de entrada tenemos:

- *El teclado*: La acción mecánica de pulsar una tecla genera internamente pulsos eléctricos que permiten identificar la información. Cada tecla representa un código que es reconocido por el computador.
- *Mouse y joysticks*: Convierten el movimiento físico en señales eléctricas binarias, que son convertidas a su vez en caracteres y se muestran en la pantalla.
- *Escáner o digitalizador de imágenes*: Interpreta uno o más caracteres, gráficos o textos que son convertidos a unos y ceros (lenguaje de máquina).
- *Lápices ópticos*: Toman lo que hay en una tableta electrónica y lo transfieren al computador.
- *Micrófonos*: Convierten la voz en señales digitales.
- *Disco duro*: Está formado por varios discos magnéticos que giran sobre un eje central. La información se lee y se imprime mediante cabezas lectoras que codifican y decodifican la información. Está dividido en cilindros, y estos, en sectores cuya cantidad está determinada por el tipo de disco y por el formato. Son de tamaño variable.
- *CD-ROM*: Posee gran flexibilidad y almacenan por mucho tiempo grandes cantidades de información. Los dispositivos de este tipo son semejantes a los ya desaparecidos disquetes.
- *Memorias USB (Universal Serial Bus)*: Es un dispositivo de almacenamiento que utiliza una *memoria flash* para guardar información. Se le conoce también con el nombre de unidad flash USB, lápiz de memoria, lápiz USB, minidisco duro, unidad de memoria, llave de memoria, entre otros. Es resistente a los golpes y a los tratamientos externos y se ha convertido en el medio de almacenamiento más común por su fácil manejo.
- *Cintas magnéticas*: Son utilizadas para almacenar grandes volúmenes de información: sobre todo archivos históricos en las grandes empresas.
- *Dispositivos OCR (Optical Character Recognition)*: Reconocen caracteres ópticos. La mayoría de datos que van a introducirse al computador se registran primero en un documento fuente, que puede ser manuscrito o mecanografiado. Los dispositivos OCR pueden leer estos datos y convertirlos directamente en impulsos eléctricos. A su vez, estos impulsos se pueden registrar directamente en cinta magnética, disco, etcétera.

No se requiere perforar tarjetas, así que no solo se economiza tiempo y dinero, sino que se disminuyen las posibilidades de error.

- *Cámara de fotos digital:* Este medio le comunica al computador las imágenes que capta. Existen varios tipos: cámara de fotos digital, cámara de videos y webcam (usada para videoconferencias).
- *Lector de código de barras:* Por medio de un láser lee dibujos formados por barras o espacios paralelos. Los lectores de códigos de barras son medios eficientes para la captación automática de datos.

### 1.4.2 La unidad de salida

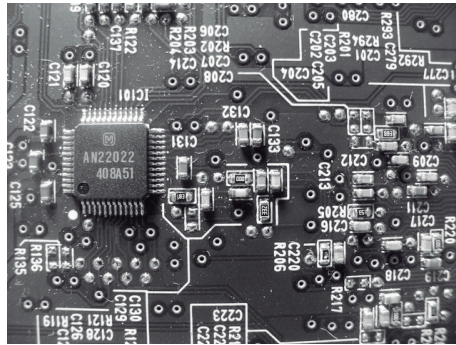
Los medios de salida de información le permiten al usuario registrar en ellos información que está guardada en la memoria tal como datos, archivos de texto, programas, gráficas, etcétera. Algunos ejemplos de medios de salida son:

- *La pantalla o monitor:* Hoy en día los dispositivos de este tipo más comunes son los de pantallas de cristal líquido (LCD) y los de diodos emisores de luz (LED).
- *La impresora:* Permite obtener los resultados en papel, que es de fácil transporte.
- *El módem:* Usado para unir computadores transformando información digital en analógica y viceversa. Es útil para que la información pueda ser transmitida a través de redes de comunicación, aunque no es de uso masivo en la actualidad.
- *Altavoces:* Por este dispositivo se emiten señales procedentes de la tarjeta de sonido. Es muy usado en la actualidad y bastante común.
- *Auriculares:* Pequeños botones que se colocan en el oído para escuchar lo que mueve la tarjeta de sonido.
- *Multimedia:* Combinación de hardware y software para la salida de texto, gráficos, animación, video, música, voz y efectos de sonido.
- *Plotters:* Medios que permiten obtener información en forma de dibujos.
- *Fax:* Este medio imprime información tomada de un texto impreso y la transmite por medio telefónico.

Los medios de almacenamiento usados como entrada y salida al mismo tiempo a menudo se conocen como *memoria auxiliar o secundaria*, y en ellos se puede almacenar información en forma permanente. A diferencia de la memoria principal, la información puede ser guardada hasta que no la necesitamos y su capacidad de almacenamiento es mucho mayor que la memoria del computador.

### 1.4.3 Unidad de control

Se compone de circuitos y/o programas que interpretan las instrucciones y comandos o mandatos que le damos al computador, y determina y da órdenes a las unidades que deben realizar un proceso. Extrae de la memoria principal los datos o instrucciones que se deben ejecutar, mandándolos (si son instrucciones de cálculo) a la unidad aritmética y lógica, que realiza las operaciones y devuelve los resultados para que sean almacenados en la memoria.



**Imagen 1.8. Unidad de control.** Balu1998 (Autor). (2013). Computadora computación tecnología de la información [Fotografía]. Recuperado el 19 de agosto de 2014, de: <http://tastyrates.com/site/wp-content/uploads/2014/01/Computer-Chip.jpg>

### 1.4.4 Unidad aritmética y lógica

Es la encargada de hacer las operaciones aritméticas (sumas, restas, multiplicaciones, divisiones y demás) y de comparación, supervisada por la unidad de control. Puede determinar si un valor es menor, menor o igual, mayor, mayor o igual, igual o diferente de otro.

Las unidades de control, memoria aritmética y lógica forman la *unidad central de proceso (CPU)*.

### 1.4.5 Unidad de memoria

La memoria principal de un computador está formada por millones de circuitos biestables (tienen dos estados), por lo que es fácil almacenar información digital. El estado prendido del circuito representa un 1, y si está apagado representa un 0. A cada uno de los estados de un circuito se le denomina bit (dígito binario). Desde luego, en un bit solo podemos guardar información que solo tenga dos estados, como el signo de un número o si es verdadero o falso.

La memoria guarda información a través de grupos de bits a los que les asigna una única dirección de memoria y que se denominan unidades de almacenamiento de información. Estos grupos para almacenar información son:

- **Bit:** Es la unidad elemental de información y puede tener dos valores (0 o 1). Constituye la unidad básica de almacenamiento utilizada en un computador.
- **Byte:** Es un conjunto de ocho bits consecutivos que se tratan como una sola entidad. También se define como el número de bits necesarios para representar un carácter, que puede ser una letra, un dígito o un carácter especial. En total son 256 caracteres: a cada uno, para representarlo, se le asigna un código numérico preestablecido.
- **Palabra de computador:** Son unidades de almacenamiento a las que se les asigna una dirección de memoria. Son tratadas por los circuitos del computador como una entidad, por la unidad de control como una instrucción y por la unidad aritmética como una cantidad. Este grupo de bits es el más grande tratado a través del procesador central como una sola unidad. El tamaño de la palabra depende de la arquitectura del computador; se pueden tener palabras de dos, cuatro, ocho o más bytes (16, 32, 64, 128 bits).

La máxima y mínima cantidad entera que se puede representar en una palabra está dada por la siguiente relación:

$$-2^{n-1} \leq \text{cantidad entera} \leq 2^{n-1}-1$$

Donde  $n$  es la cantidad de bits de la palabra. Por ejemplo, si la palabra es de 16 bits, esa cantidad entera estará dentro de los siguientes rangos:

$$-2^{15} \leq \text{cantidad entera} \leq 2^{15}-1 \quad (-32767 \leq \text{cantidad entera} \leq 32768).$$

Si deseamos almacenar cantidades más grandes, lo que hacemos es unir dos o más palabras mediante el lenguaje de programación que estemos utilizando.

La memoria principal está dividida en dos partes:

**Memoria RAM (Random Access Memory o memoria de acceso aleatorio):** Es donde el computador guarda los datos que está utilizando en el momento presente. Se llama de acceso aleatorio porque el procesador accede a la información que está en la memoria en cualquier punto sin tener que acceder a la información anterior y posterior. Es la parte de la memoria disponible para

guardar información temporal que está utilizando un programa. La aleatoriedad hace que la información sea guardada en cualquier parte de la memoria en grupos de bits que tienen una única dirección de memoria. Las direcciones de memoria se suelen representar en hexadecimal. Por ejemplo: 0x8f95fff4, 0x8f95fff2, 0x8f95fee.

**Memoria ROM (Read Only Memory o memoria de solo lectura):** Es un pequeño espacio de la memoria utilizado en los computadores y dispositivos electrónicos que permite solo la lectura de la información y no su escritura. La información contenida en una memoria ROM es fija, independientemente de la presencia o no de una fuente de energía. Trae programas que están estrechamente ligados al hardware específico, como los que ponen en marcha el ordenador y realizan los diagnósticos, y es poco probable que requiera actualizaciones frecuentes u otro contenido vital para el funcionamiento del dispositivo.

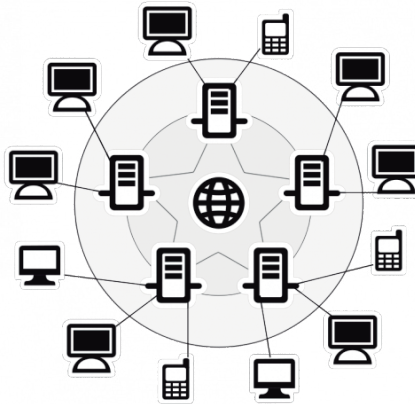
## 1.5 Tipos de procesamiento

Consideremos ahora un aspecto diferente en cuanto al procesamiento de datos en un centro de cómputo. En el caso de la elaboración de una nómina que se paga quincenalmente, las novedades respecto a tiempo extra u otros temas se reúnen cada semana y se guardan o almacenan hasta que llegue el momento de procesarlas, es decir, las novedades no se procesan en el instante en que se generan, sino que se acumulan para hacerlo en el momento oportuno. Este tipo de proceso, que no es muy utilizado en la actualidad, se denomina *proceso por lotes*.

Ahora supongamos que alguien va a una oficina de una aerolínea para hacer reservaciones en algún vuelo. Como es obvio, el empleado no puede guardar la solicitud para procesarla luego con muchas otras, sino que los datos deben procesarse inmediatamente. A este tipo de proceso se le denomina *procesamiento directo o en línea*.

Con el tiempo, los requisitos de procesamiento avanzaron a *tiempo compartido*, donde el computador está programado de tal manera que reparte el tiempo disponible de procesamiento entre varios usuarios y/o tareas. De este modo el sistema, al mismo tiempo que está produciendo una cierta salida para algún usuario, ejecuta determinados cálculos para otro. A las tareas se les asigna una cuota de tiempo de procesamiento, lo que da la sensación de procesamiento paralelo.

Las telecomunicaciones, junto con los adelantos en los sistemas de comunicación, han hecho posible la práctica de lo que se denomina *procesamiento en red*, donde el computador está programado de tal manera que admite peticiones de procesamiento hechas desde otras máquinas en el mismo lugar o desde lugares lejanos. Las telecomunicaciones se refieren a aquellas técnicas que permiten la transmisión y recepción de señales entre dispositivos como tabletas, equipos móviles, entre otros, que requieren comunicarse por medio de una red.



**Imagen 1.9. Comunicación de datos en una red.** OpenClips (Autor). (2013). Internet Red Esquema Globalización Redes [Imagen]. Recuperado el 19 de agosto de 2014, de: <http://publicdomainvectors.org/es/vectoriales-gratuitas/Esquema-de-Vector-de-Internet/7769.html>

Cuando se requiere comunicar dos o más computadores o dispositivos equivalentes, se genera desde el transmisor una serie de códigos. Estos pasan, por lo general, a un sistema de codificación de datos que se encarga de prepararlos para que puedan transmitirse por el canal empleado en la comunicación. Este canal puede ser cableado o inalámbrico, lo que implica el uso de diversas tecnologías de comunicación. En el dispositivo receptor se encuentra otro sistema que se encarga de hacer la decodificación de los datos, dejándolos preparados para ingresar a la unidad central de procesamiento.

## 1.6 Definiciones básicas

A continuación se presentan una serie de definiciones necesarias para desarrollar el contenido del libro:

- **Soporte físico (hardware):** Son todas las partes tangibles de un computador, también llamadas *partes duras* o conjunto de componentes de



material. Comprende la parte eléctrica, electrónica, mecánica, electromecánica, de cableado, periférica, tarjetas e incluso los mismos circuitos de los distintos componentes.

- **Soporte lógico (software):** Es la parte *blanda* del computador, que comprende todo el conjunto de componentes lógicos de un sistema de cómputo tales como los programas que hacen las tareas que se quieren resolver (entre ellos los programas que hacemos nosotros y los que vienen con el computador) y las aplicaciones informáticas como el sistema operativo, los procesadores de texto, el software del sistema, etcétera. El término fue concebido para contrastar con el hardware de un sistema de computador.
- **Dirección de memoria:** Cada computador tiene una cantidad determinada de *almacenamiento interno* denominada memoria principal. Esta es direccionable, lo que quiere decir que a cada variable o instrucción se la asigna una única dirección de memoria, y esta corresponde a un pequeño espacio en la memoria RAM. Las direcciones de memoria están en hexadecimal y son comparables a las direcciones de las casas de una ciudad, ya que no están repetidas y cada dirección corresponde al espacio de la casa.
- **Campo:** Es un grupo de caracteres que se pueden tratar como una unidad de información simple y a los que se les asigna una dirección de memoria única. Los programas, por ejemplo, se elaboran con campos, mientras que las instrucciones de los lenguajes están constituidas por varios campos. Dependiendo del computador, del lenguaje de programación utilizado y de la forma como el campo sea definido, a un campo se le asignará una o más palabras. Los campos pueden ser:
  - a) **Campos constantes.** Su contenido no cambia en el transcurso del programa, estos pueden ser:
    - 1) **Campos constantes numéricos.** Formados por dígitos con el punto y decimales sin el punto
      - 1.1) **Campos constantes numéricos enteros.** Formados por dígitos sin el punto decimal. Ejemplos:  
251, 1951, 32675, 675432198654
      - 1.2) **Campos constantes numéricos reales.** Formado por dígitos incluido el punto decimal. Ejemplos:  
3.1416, 9.8, 65433.8765
    - 2) **Campos constantes de caracteres.** Formados por caracteres y se denotan por comillas. Cuando cualquier carácter se encierra entre comillas, se trata es de la figura que se denota y es almacenada en Bytes. Ejemplos: ‘a’, ‘,’’, ‘B’, “Samuel David”, “ab”, “María Isabel”, “Emilia”, “Placa”, ‘1’, “123”.

**b) Campos variables o simplemente variables:** Las variables son usadas para llevar a memoria todo dato que entre el usuario o que vaya a ser calculado como resultado intermedio o como resultado final. Nosotros, como programadores, construimos este tipo de campos, que deben iniciar con una letra, seguida de letras o dígitos. Dependiendo del lenguaje de programación utilizado, se puede incluir el carácter subrayado (se recomienda que las letras estén en minúscula para diferenciarlas de las constantes que trae el lenguaje de programación que vienen en mayúscula). Ejemplos: a, ab12, s899, salario, nombres, k, i, nota, salario\_bruto, salarioNeto, etcétera.

Se aconseja también que los campos variables que uno seleccione se parezcan en algo a lo que uno quiere representar para darle claridad al programa. Todos los valores de un programa se guardan en variables. Por ejemplo, si queremos guardar una cédula, debemos escoger un nombre de variable para que sea guardada en la memoria del computador. Cada quien es libre de escoger sus propios nombres de variables, pero es mejor seleccionar caracteres que de por sí denoten lo que están representando. Para el caso de la cédula, podemos seleccionar n36cd, pero es más entendible si seleccionamos cc o cédula.

- **Instrucción:** Son campos (constantes y/o variables) unidos a través de operadores. Por ejemplo:  $a=b+c-20$ .
- **Programa fuente:** Es un conjunto de instrucciones relacionadas y escritas en un lenguaje de programación que realizan una tarea específica. El computador no entiende el programa fuente; por lo tanto, hay que traducirlo a lenguaje de máquina (unos y ceros). Para esto se necesita de un traductor, es decir, de un ensamblador, un compilador o un intérprete. Un ejemplo de un programa fuente codificado (representado) en visual C++ es:

```
#include<iostream>
using namespace std;
class suma{};
int main(){
    int x, y, z;
    cout<<"DIGITE UN VALOR ENTERO: ";
    cin>>x;
    cout<<"DIGITE OTRO VALOR ENTERO: ";
    cin>>y;
    z= x +y;
    cout<<"LA SUMA DE: "<<x<< "+"<<y<<" ES: "<<z<<endl;
    system("pause");
}
```

- **Ensamblador:** Traduce las instrucciones simbólicas de un lenguaje de programación en las instrucciones necesarias de máquina para que sean ejecutadas. En los lenguajes ensambladores se define un código especial (llamado nemónico) para cada una de las operaciones de la máquina y se introduce una notación especial para especificar el dato con el cual debe realizarse la operación. El ensamblador origina una instrucción en lenguaje de máquina por cada instrucción del lenguaje que se va a traducir. En general, es utilizado con los lenguajes de bajo nivel.
- **Compilador:** Son traductores más complejos, usados con lenguajes de alto nivel. Traducen los programas que realiza el usuario al lenguaje de máquina necesario para que sean ejecutados. Además de hacer dicha traducción, el compilador chequea que el programa fuente esté correcto en cuanto a sintaxis e instrucciones propias de cada lenguaje.

La traducción, en general, se realiza en dos pasos: primero las instrucciones en lenguaje de alto nivel se traducen a una secuencia de instrucciones en el lenguaje de bajo nivel, y a continuación se efectúa la traducción al lenguaje de máquina. El compilador asigna las posiciones de memoria por cada campo que encuentra conforme lee y traduce las instrucciones.

- **Interpretador.** Es un programa de computador que traduce cada instrucción o sentencia del lenguaje original a una secuencia de instrucciones del lenguaje de la máquina.
- **Programa objeto.** Es un programa fuente que ha sido traducido a lenguaje de máquina. Es el programa que ejecuta el computador. Se convierte, entonces, en un conjunto de instrucciones de lenguaje de máquina para la solución de un problema, obtenido como el resultado final de un proceso de traducción.



Imagen 1.10. Proceso de traducción.

- **Lenguaje de computador:** Conjunto definido de caracteres que sirven para formar símbolos, palabras, instrucciones, etcétera, así como de las reglas para combinar estos caracteres, lo que da como resultado comunicaciones significativas, por ejemplo FORTRAN, PASCAL, PROLOG, C, C++, Java, entre otros.
- **Lenguaje de máquina:** Es el conjunto final que todos los computadores deben utilizar (lenguaje binario). Todos los demás lenguajes de programación deben compilarse o traducirse, a la larga, a un código binario antes de introducirse en el procesador.

- **Lenguaje de bajo nivel:** Es el primer paso de la sofisticación de los lenguajes del computador. Permite una comunicación más fácil con la máquina. Las instrucciones escritas en este lenguaje deben ser traducidas por medio de un traductor a lenguaje de máquina. La traducción que se hace es uno a uno, es decir, una instrucción en lenguaje de bajo nivel se transforma en una instrucción de lenguaje de máquina.
- **Lenguaje de alto nivel:** Es el lenguaje simbólico más fácil de entender y de programar. Se traduce a lenguaje objeto mediante un compilador y permite preparar un programa para ser procesado por el computador. A diferencia de los anteriores, este es más flexible, fácil de aprender y más apropiado para corregir los errores de programación. Una instrucción de lenguaje de alto nivel es traducida a lenguaje de máquina en una o varias instrucciones. En la medida en que se simplifica el trabajo del programador, crece el trabajo del traductor.
- **Archivo:** Es utilizado para almacenar información, relacionada con una determinada entidad, persona o cosas, en forma masiva. Podemos decir entonces que un archivo es un conjunto de registros relacionados. Por ejemplo, suponga que la compañía A tiene seis empleados y que se mantiene la siguiente información acerca de cada uno de ellos: cédula, nombre y salario. La información individual de cada cliente se denomina *registro*, y un conjunto de registros formará un archivo, es decir, un archivo de datos de los registros de los clientes. Los archivos pueden tenerse en varias formas y en diferentes medios de almacenamiento.
- **Base de datos:** Es un archivo amplio, sin información repetida, relacionado de una manera especial y que continuamente se está actualizando.
- **Representación de datos:** Define la forma en que el computador almacena y procesa diferentes tipos de datos tales como alfanuméricos, numéricos enteros y numéricos reales.
- **Sistema numérico:** Define los símbolos necesarios para representar  $r$  dígitos. La cantidad  $r$  se conoce como la base numérica y las más comunes son decimal ( $r=10$ ), binaria ( $r=2$ ), octal ( $r=8$ ) y hexadecimal ( $r=16$ ).

## 1.7 Sistemas de numeración<sup>2</sup>

Teniendo en cuenta que los computadores utilizan sistemas numéricos diferentes al que utilizamos los humanos (decimal), en esta sección se presentan los sistemas de numeración existentes y la forma en que se puede operar con ellos para contribuir al entendimiento operacional del computador.

---

2 Nota: Ver video 1.1 Sistemas numéricos en el complemento SIL de [www.ecoediciones.com](http://www.ecoediciones.com)

Un *sistema de numeración* o *sistema numérico* es un conjunto de símbolos y reglas de generación que permiten construir todos los números válidos del sistema de numeración con que se trabaje. Todo sistema de numeración o numérico tiene: un nombre (decimal, binario, octal, etc.), unos símbolos o dígitos que lo conforman y la base ( $r$ ), que depende del sistema de numeración. El sistema que nosotros más usamos es el decimal, y el computador usa el sistema numérico binario.

### 1.7.1 Paso de una base cualquiera $r$ a decimal

Se hace multiplicando cada dígito por potencias enteras de la base  $r$ . Luego se suman todos los dígitos pesados, es decir, todos aquellos que tengan 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 824.25 se utiliza para representar la cantidad:

$$8 \cdot 10^2 + 2 \cdot 10^1 + 4 \cdot 10^0 + 2 \cdot 10^{-1} + 5 \cdot 10^{-2}$$

Esto es: 8 centenas + 2 decenas + 4 unidades + 2 décimos + 5 centésimos.

Observe que el dígito 4 se multiplica por  $10^0$  (o sea 1) porque representa unidades, lo cual daría  $4 \cdot 1 = 4$ ; el 2 se multiplica por  $10^1$  (o sea 10) por que representa decenas; el 8 por  $10^2$  para indicar centenas. Después del punto, el 2 se multiplica por  $10^{-1}$  para indicar décimas; el 5 se multiplica por  $10^{-2}$ , para indicar centésimas y así sucesivamente. También lo podemos mostrar en forma vertical así:

$$\begin{array}{r}
 824.25 \\
 \begin{array}{l}
 \left. \begin{array}{l} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \right\} \begin{array}{l}
 5 \times 10^{-2} \rightarrow 5/100 = 0.05 \\
 2 \times 10^{-1} \rightarrow 2/100 = 0.2 \\
 4 \times 10^0 \rightarrow 4 \times 1 = 4 \quad + \\
 2 \times 10^1 \rightarrow 2 \times 10 = 20 \\
 8 \times 10^2 \rightarrow 8 \times 100 = 800 \\
 \hline
 824.25
 \end{array}
 \end{array}
 \end{array}$$

- En el sistema numérico binario,  $r=2$  (base=2); por lo tanto, se utilizan dos dígitos, que son 0 y 1.

- En el sistema numérico octal,  $r=8$  (base=8); los dígitos que se utilizan son: 0, 1, 2, 3, 4, 5, 6 y 7.
- En el sistema de numeración hexadecimal,  $r=16$  (base =16); los dígitos que se utilizan son: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F. Como son solo 10 dígitos decimales, se utilizan las primeras letras del alfabeto para representar los dígitos restantes. Así, A=10, B =11, C=12, D=13, E=14 y F=15. Observe que el último dígito de un sistema de numeración es la base menos uno.

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 45 en base dos escribiremos:

$$(\text{Número})_{\text{base origen}} \rightarrow (\text{Número})_{\text{base destino}}$$

$$(45)_{10} \rightarrow (101101)_2$$

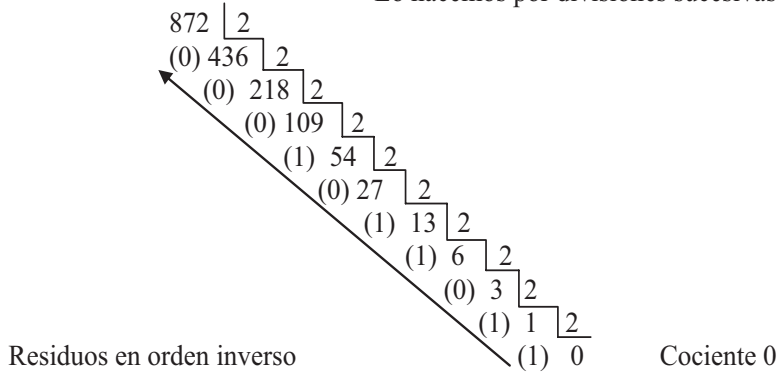
### 1.7.2 Paso de decimal a una base cualquiera r

Se hace separando el número en su parte entera y su parte decimal (si la tiene) y convirtiendo cada parte por separado. La conversión de la parte entera se ejecuta mediante divisiones sucesivas por la base  $r$  hasta obtener un cociente cero, y se toman 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 dicha base hasta llegar a una fracción cero o a la precisión deseada (podemos multiplicar hasta obtener, por ejemplo, ocho cifras decimales), y se toman los dígitos enteros obtenidos en el orden de aparición.

• **Paso de decimal a binario**

Se hace a través de divisiones sucesivas por la base 2.

$(872)_{10} \rightarrow ( \quad )_2$  Pasar de base 10 a base 2.  
Lo hacemos por divisiones sucesivas.



$(872)_{10} = (1101101000)_2$

En representación numérica en base 2, binaria, es común utilizar el bit de mayor peso (el bit más a la izquierda de la palabra) para indicar si una cantidad es positiva o negativa. Cuando este bit sea 0 implica que la cantidad representada es positiva y si es 1 la cantidad será negativa. Dicho bit se conoce como bit de signo.

Otra forma de pasar de base 10 a base dos es a partir de la distribución del número en la palabra, donde cada bit de esta representa potencias de 2, así:

$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	
Bit de signo	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

Si queremos representar 13 en binario, lo descomponemos en potencias de 2,  $(8+4+1)=2^3+2^2+2^0$ , lo que quiere decir que el bit que representa a 1 ( $2^0$ ) estará prendido, o sea, es 1; el que representa a 2 ( $2^1$ ) estará apagado; el que representa a 4 ( $2^2$ ) será 1 y el que representa a 8 ( $2^3$ ) será 1. Los demás bits a la izquierda serán ceros.

0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
												8	4	1	

Bit de signo

Para el paso de 872 a binario, buscamos la potencia de dos más cercana a 872 por debajo, o sea, 512, y se la restamos ( $872-512=360$ ). Ahora buscamos la potencia de dos más cercana a 360, que es 256, y la restamos ( $360-256=104$ ); buscamos la potencia de dos más cercana a 104, que es 64, y la restamos ( $104-64=40$ ); buscamos la potencia de dos más cercana a 40, que es 32, y la restamos ( $40-32=8$ ); buscamos la potencia de dos más cercana a 8, que es el mismo 8, y la restamos ( $8-8=0$ ). Hasta aquí llegamos buscando potencias de dos: esto quiere decir que llegamos hasta un punto en el que la resta da cero; por lo tanto,  $872=512+256+64+32+8$ . Esto implica que los bits dentro de la palabra que deben estar prendidos (serán unos) son los que representan a 512, 256, 64, 32 y 8, y los demás serán ceros. Gráficamente lo podemos representar de esta manera:

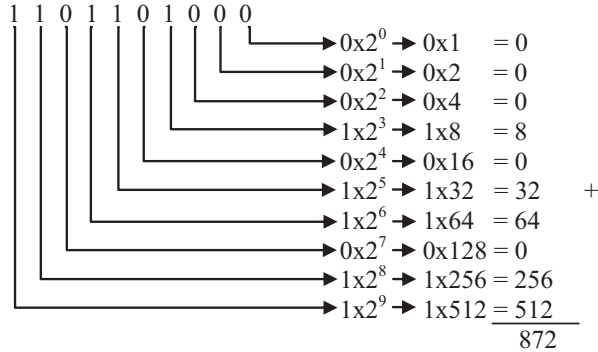
$$\begin{array}{r|l}
 872 & 512 \\
 -512 & \\
 \hline
 360 & 256 \\
 -256 & \\
 \hline
 104 & 64 \\
 -64 & \\
 \hline
 40 & 32 \\
 -32 & \\
 \hline
 8 & 8 \\
 -8 & \\
 \hline
 0 & 
 \end{array}$$

Lo anterior quiere decir que los bits que representan a las potencias de dos que aparecieron están prendidos y las potencias de dos que no aparecieron están apagadas. Por lo tanto:

$$(872)_{10} \rightarrow (1101101000)_2$$



- **Paso de binario a decimal:** Ahora hagamos lo contrario, es decir, pasemos de  $(1101101000)_2$  a  $(872)_{10}$  así:



- **Otros ejemplos de cambios de base:**  
 1.  $(321.625)_{10} ( \quad )_2$

Como el número decimal tiene parte fraccionaria, se pasa a base dos por partes: primero la parte entera mediante divisiones sucesivas y luego la parte fraccionaria multiplicando la fracción en forma sucesiva por la base destino hasta obtener una fracción cero o la precisión deseada.

*Parte entera:*

$$(321)_{10} \rightarrow ( \quad )_2$$

321	256
-256	64
65	
-64	
1	1
-1	
0	

$$(321)_{10} \rightarrow (101000001)_2$$

*Parte fraccionaria:*

$$(0.625)_{10} \rightarrow (0,101)_2$$



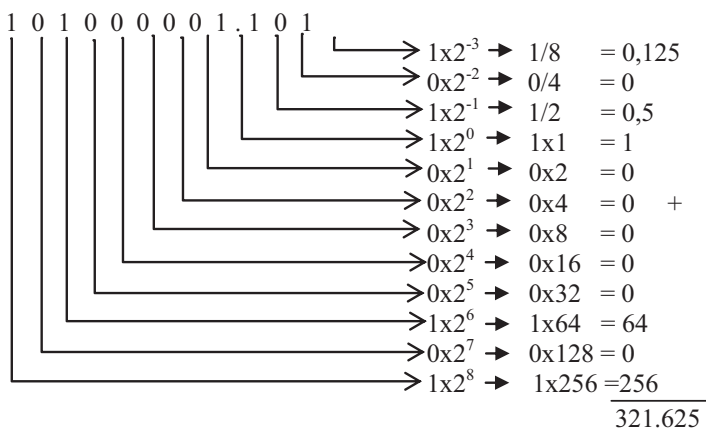
$$0.625 * 2 = 1.25 \text{ (fracción 0.25)}$$

$$0.25 * 2 = 0.5 \text{ (fracción 0.5)}$$

$$0.5 * 2 = 1.0 \text{ (fracción 0)}$$

Tomamos la parte entera en el orden de aparición, es decir, de arriba hacia abajo=0.101. Luego, uniendo las dos representaciones nos queda que  $(321.625)_{10}$  es  $(101000001.101)_2$ .

2.  $(101000001.101)_2 \rightarrow ( \quad )_{10}$



$(101000001.101)_2 \rightarrow (321.625)_{10}$

3.  $(130.23)_{10} \rightarrow ( \quad )_2$

Parte entera:

$(130)_{10} \rightarrow (10000010)_2$

Parte fraccionaria:

$(0.23)_{10} \rightarrow ( \quad )_2$

- ↓ Tomamos la parte entera en el orden de aparición:
- $0.23 * 2 = 0.46$
  - $0.46 * 2 = 0.92$
  - $0.92 * 2 = 1.84$
  - $0.84 * 2 = 1.68$
  - $0.68 * 2 = 1.36$
  - $0.36 * 2 = 0.72$
  - $0.72 * 2 = 1.44$
  - $0.44 * 2 = 0.88$
- No se llega a obtener una fracción cero,  
por lo que tomamos ocho dígitos decimales.

$(0.23)_{10} \rightarrow (0.00111010)_2$

Unimos las dos representaciones:

$(130.23)_{10} \rightarrow (10000010.00111010)_2$

- **Paso de decimal a octal:** En este caso la parte entera se divide sucesivamente por 8 y la parte fraccionaria se multiplica sucesivamente por 8. Ejemplo:

$$(954.25)_{10} \rightarrow ( \quad )_8$$

*Parte entera:*

$$(954)_{10} \rightarrow ( \quad )_8$$

954	8		
(2) 119	8		
(7) 14	8		
(6) 1	8		
(1) 0	8		Cociente cero

Residuos en orden  
inverso.

$$(954)_{10} \rightarrow (1672)_8$$

*Parte fraccionaria:*

$$(0.25)_{10} \rightarrow ( \quad )_8$$

$$0.25 \times 8 = 2.0: \text{cociente cero, parte entera 2.}$$

*Unimos las dos representaciones:*

$$(954.25)_{10} \rightarrow (1672.2)_8$$

- **Paso de octal a decimal:**

$$(1672.2)_8 \rightarrow (954.25)_{10}$$

$$1 \cdot 8^3 + 6 \cdot 8^2 + 7 \cdot 8^1 + 2 \cdot 8^0 + 2 \cdot 8^{-1} = 512 + 384 + 56 + 2 + 0.25 = 954.25$$

- **Paso de decimal a hexadecimal:**

$$(954.25)_{10} \rightarrow (\quad)_{16}$$

*Parte entera:*

$$(954)_{10} \rightarrow (\quad)_{16}$$

$$\begin{array}{r} 954 \overline{)16} \\ \underline{(10)59} \phantom{0} \overline{)16} \\ \phantom{0}(11)3 \phantom{0} \overline{)16} \\ \phantom{00}(3)0 \phantom{0} \overline{)16} \text{Cociente cero} \end{array}$$

Residuos en orden  
inverso

$$(954)_{10} \rightarrow (3BA)_{16}$$

*Parte fraccionaria:*

$$(0.25)_{10} \rightarrow (0.4)_{16}$$

$$0.25 * 16 = 4.0: \text{cociente cero y parte entera } 4$$

*Unimos las dos representaciones:*

$$(954.25)_{10} \rightarrow (3BA.4)_{16}$$

- **Paso de hexadecimal a decimal:**

$$(FC1.D2)_{16} \rightarrow (4033.8203)_{10}$$

$$\begin{aligned} &15 * 16^2 + 12 * 16^1 + 1 * 16^0 + 13 * 16^{-1} + 2 * 16^{-2} \\ &= 3840 + 192 + 1 + 0.8125 + 0.0078125 = 4033.8203 \end{aligned}$$

La conversión entre representaciones numéricas binarias, octales y hexadecimales 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. Esto significa que si encontramos un  $n$  tal que  $2^n = r$ , cada dígito de la base  $r$  necesita de  $n$  dígitos binarios para ser representado.

- **Paso de octal a binario:** Esta conversión se hace representando cada dígito octal en tres dígitos binarios, de derecha a izquierda antes del punto y de izquierda a derecha después del punto. Por ejemplo:

$$(1\ 7\ 6\ 2\ 3)_8 \rightarrow ( \quad \quad \quad )_2$$

Una de las formas de solución a este ejercicio sería usando la base 10 como auxiliar, así:

$$(1\ 7\ 6\ 2\ 3)_8 \rightarrow ( \quad \quad \quad )_{10} \rightarrow ( \quad \quad \quad )_2$$

lo que sería muy largo y tedioso.

La otra forma es muy sencilla. Como dijimos, se pasa cada dígito de la base ocho a binario y se representa en tres dígitos binarios y si el dígito octal no tiene tres dígitos binarios, lo completamos con ceros:

$$(1\ 7\ 6\ 2\ 3)_8 \rightarrow (1111110010011)_2$$

1 111 110 010 011

Así podemos obtener una tabla de conversión de octal a binario:

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

$$(650.134)_8 \quad ( \quad \quad \quad )_2$$

$$( \overset{\leftarrow}{6} \overset{\rightarrow}{5} 0. 1 \overset{\leftarrow}{3} \overset{\rightarrow}{4} )_8 \rightarrow (110101000.0010111)_2$$

110 101 000 001 011 100

- **Paso de binario a octal:** La conversión de binaria a octal se ejecuta fácilmente dividiendo el número binario en grupos de tres dígitos de derecha a izquierda antes del punto y de derecha a izquierda después del punto. A cada grupo de tres dígitos se le asigna el correspondiente dígito octal, y la