

Capítulo 1

Marco conceptual de la Lógica de Programación

1.1 Concepto de Lógica

El desarrollo de la Informática como ciencia que estudia el tratamiento de la información a través de máquinas de procesamiento electrónico de datos al servicio de una sociedad digital y global se basa tanto en la tecnología como en la capacidad racional humana, con el fin de transformar un conjunto de hechos a partir de los cuales se logra deducir a través de la razón humana un conjunto de resultados, que al ser valorados en su condición de verdad pueden ser lógicamente correctos o incorrectos. Lo anterior implica, para el avance informático, tanto el desarrollo tecnológico como el desarrollo de las formas de razonamiento humano.

El ser humano racional tiene su base biológica en el cerebro, como órgano que permite a través del lenguaje utilizar los métodos y principios para distinguir el razonamiento correcto del incorrecto. La capacidad de razonamiento humano se basa en un conjunto de reglas o formas válidas de deducción o inferencia, utilizadas por la civilización griega. El conjunto de reglas mencionado conformaba un sistema de razonamiento, y los griegos “llamaron a este sistema de razonamiento ‘Lógica’” (Dean, 2003, p. 1). El filósofo griego Aristóteles fue el primero en referirse a la noción de lógica en el entendido de que los argumentos derivados de la razón son considerados como útiles para lograr la verdad en una determinada ciencia. Luego el término “lógica”, que etimológicamente se deriva del latín *lógica*, que a su vez proviene del griego antiguo *logike*, es usado para referirse a un sistema de razonamiento formal. Es este sistema de razonamiento formal el que se utiliza en la construcción de algoritmos; y es precisamente el sistema de reglas que componen un algoritmo lo que se codifica y se ejecuta en un computador para derivar de dicha ejecución resultados que, en síntesis, se convierten en información útil al usuario.

Los sistemas de razonamiento a través del lenguaje han evolucionado de un razonamiento puramente verbal denominado “lenguaje natural” a razonamiento simbóli-

co o lógica simbólica, justificado por el hecho de que

... en la lógica simbólica la información es representada usando letras y símbolos, similares al Álgebra. La lógica simbólica posibilita un cierto grado de automatización del razonamiento; sin embargo, su motivación original fue el deseo de ser capaz de solucionar un problema lógico a través de “cálculos”, justamente como la respuesta a un problema numérico puede ser calculado usando aritmética. Por esta razón, la lógica simbólica está cerradamente asociada con computadores... (Dean, 2003, p. 2)

Luego, con base en lo anterior, la lógica es una rama de la filosofía que estudia de manera formal las deducciones válidas que se derivan de un sistema de razonamiento fundamentado en un conjunto de reglas. Si el sistema de razonamiento mencionado se expresa en un lenguaje matemático, recibe el nombre de “lógica matemática”; en el caso de que el sistemas de razonamiento utilice un lenguaje simbólico y un conjunto de reglas de inferencia recibe el nombre de “lógica simbólica”; y es precisamente esta última la que a través de los algoritmos conformados por estructuras lógicas ha permitido el desarrollo de la informática. El concepto de lógica simbólica antes mencionado, en su intento por alcanzar una automatización utiliza la algoritmia, a cuya ciencia pertenece el concepto de algoritmo.

1.2 Concepto de algoritmo

1.2.1 Síntesis histórica del desarrollo de los algoritmos

Los algoritmos relacionados con la lógica, y esta, a su vez, teniendo relación tanto con la matemática como con el lenguaje, no pueden ser aislados en su desarrollo histórico, por lo tanto, de la historia de los sistemas numéricos y de los sistemas de alfabetos de soporte a un lenguaje. Las primeras construcciones de algoritmos se remontan a la civilización babilónica (2000-600 a.C.), y están relacionadas con el campo matemático, particularmente en la resolución de ecuaciones y en la utilización del sistema numérico sexagesimal (Base 60 (b_{60})); sistema heredado de los sumerios (2500 a.C.), cuya ubicación geográfica es el valle del río Euphrates (Irak, sur de Bagdad); sistema que se fundamentó en signos especiales para los números 1, 10, 60, 600 y 3600, de los cual se puede concluir que trabajaban tanto en base 10 como en la base sexagesimal.

Es importante destacar que “... Al comienzo del segundo siglo antes de Jesucristo, los babilonios desarrollaron un sistema de numeración, el cual se mantiene en nuestros días en el uso de los minutos y segundos para el tiempo y en la medida de un ángulo” (Chabert & Barbin, 1999, p. 11).

La necesidad del lenguaje en el proceso de comunicación dentro de las sociedades se hizo inicialmente a través de la descripción de acciones cotidianas utilizando figuras (3000 a.C.), para posteriormente ser simplificada la representación de acciones

en símbolos bases para la generación de alfabetos, de lo cual se derivó el lenguaje escrito. “El primer alfabeto verdadero apareció cerca de 1700 a.C. en el Este del Mediterráneo” (Wear et al., 1991, p. 7).

La contribución de la antigua civilización egipcia en el desarrollo de los algoritmos se basa en su sistema decimal, representado a través de símbolos que siendo escritura jeroglífica sirvió para representar palabras (2000 d.C.); “los egipcios también usaron unidades fraccionarias” (Chabert & Barbin, 1999, p. 15), y de esta época data la representación de fracciones como $2/3$ y $1/3$, a la cual se le llamó ‘la pequeña mitad’.

Teniendo en cuenta que hay muchos algoritmos para la multiplicación, se destaca que la multiplicación en forma de tabla, como se conoce actualmente, tuvo su génesis en diferentes tiempos en la China, la India, el mundo árabe y la zona europea. Las evidencias de las tablas de multiplicar en China datan del año 1450, utilizadas a nivel financiero en la provincia de Zhejiang; por su parte, en Europa la más reciente evidencia del algoritmo de multiplicación aparece “... en un manuscrito en latín, alrededor del año 1300 en Inglaterra en la región de Edward II” (Chabert & Barbin, 1999, p. 25).

El desarrollo del lenguaje a través de la teoría de alfabetos después de su primera aparición se complementó con la creación de los alfabetos etrusco y griego alrededor de los años 800-700 d.C., llegando hasta el antiguo alfabeto romano en el año 600-100 d.C., y finalmente el nuevo alfabeto romano durante los años 400-1500 (a.C.).

En el desarrollo algorítmico de las operaciones aritméticas es importante destacar el ábaco chino por su relación con los cálculos de las operaciones suma, resta y la multiplicación, ubicado históricamente en la segunda mitad del siglo XVI. Las operaciones aritméticas solo fueron posibles con el desarrollo de los sistemas de numeración, cuyo desarrollo fue paralelo al alfabeto, por la necesidad de la sociedad de tener un sistema de conteo; así, el sistema más comúnmente utilizado fue el sistema decimal o en base 10 (b_{10}). En el sistema mencionado, y teniendo como base la unificación de precios y medidas de la antigua dinastía Qin en China (210 d.C.), se utilizaron las decenas, centenas y unidades de mil como cantidades posicionales del sistema decimal, lo cual hizo posible la aparición de fracciones decimales. El sistema de numeración binario o base dos (b_2) tiene como representante en la aritmética binaria a

Leibniz, quien escribió su memoria sobre La Explicación de la Aritmética Binaria en 1703, pero estuvo por más de quince años interesado en un sistema de numeración, el cual fue para él un modelo de lo que llamó una característica universal. (Chabert & Barbin, 1999, p. 11).

El sistema binario tuvo un sistema posicional generalizado construido por el mismo Leibniz en 1666; pero se debe tener en cuenta que realmente el primer sistema de posicionamiento digital se le debe a Thomas Harriot (1560-1621), cuya obra no fue publicada y Leibniz lo realizó de una forma independiente. Es importante destacar que con la creación del sistema binario fue posible reducir la lógica a operaciones mecánicas; y siendo reducida la lógica a estas operaciones con el nacimiento de la

1.2. Concepto de algoritmo

máquina para la realización de sumas llamada “máquina de Pascal”, habiendo Leibniz inventado una máquina para la ejecución de las cuatro operaciones aritméticas, este tipo de máquinas precedió a las de computación moderna, entre las cuales se encuentra la de Charles Babbage, cuya identidad es la ‘máquina de diferencias’, inventada por este en 1822. De forma paralela, es importante destacar los siguientes eventos en la evolución de la matemática relacionada con la lógica binaria: el tratamiento matemático a través de los conceptos del cálculo diferencial e integral se le acreditan a Leibniz y Newton entre 1675 y 1687. La introducción del concepto de función matemática se le debe a Euler, quien alrededor de 1748 formalizó los conceptos de la matemática analítica. La definición del Álgebra Lineal se le debe a Johan Carl Friedrich Gauss (1777-1855), a quien también se le atribuyen importantes contribuciones en los campos de Teoría de Números, Estadística, Geofísica y Electrostática, entre otros campos, dentro de los cuales se encuentran importantes algoritmos, como son los casos del cálculo del promedio y la desviación estándar. Adicionalmente, la creación de las llamadas retículas booleanas o el Álgebra de Boole se le debe a George Boole (1815-1864), de fundamental importancia para el desarrollo computacional.

La necesidad que ha tenido la sociedad de llevar un registro de sus avances en términos cuantitativos y cualitativos hizo posible el desarrollo de la teoría de alfabetos y lenguajes de comunicación entre humanos, desde los antiguos papiros egipcios, pasando por la invención de la imprenta (1398-1400) por Johannes Gutemberg y llegando hasta los actuales sistemas de información basados en el computador. Igualmente, el registro de las acciones humanas permitió con base en el avance de los sistemas numéricos, y particularmente sobre el fundamento del sistema binario, el desarrollo de la computación actual.

Teniendo en cuenta los desarrollos en el campo del lenguaje y la matemática, se le atribuye a Howard Aiken y Grace Hopper la creación del primer computador, identificado como el *Harvard Mark I Computer*, hacia 1944; y posteriormente, la invención del computador de tubos al vacío, llamado el ENIAC I, en 1946, a cargo de Jhon Presper Eckert y John W. Mauchly, a quienes también se les acredita la creación del computador UNIVAC en 1953. En el mismo año, la compañía International Business Machines (IBM) entra a participar en el desarrollo de la computación con la fabricación del computador IBM 701 EDPM; y en 1954 se crea el primer lenguaje programable en un computador de alto nivel llamado FORTRAN, a cargo de John Backus y la compañía IBM.

Los eventos o acciones más importantes en el desarrollo de la algoritmia desde la Edad Antigua hasta la Edad Moderna son los siguientes, teniendo en cuenta que dichos eventos están necesariamente ligados con el desarrollo de la Matemática, la Ciencia de la Computación y la Informática:

- (3000 a.C.): creación de símbolos para representativos de las acciones humanas.
- (2000 a.C.): los babilonios inventan el sistema sexagesimal.
- (1700 a.C.): creación de las letras para la representar sonidos individuales.

Capítulo 1. Lógica de programación

- (1600 a.C.): Demócrito de Abdera encontró la fórmula para calcular el volumen de una pirámide.
- (750-850 a.C.): Al Khowarismi inventó las reglas de las cuatro operaciones básicas de la aritmética, época en que se acuñó el término “algorism”.
- (408 a.C.): Eudoxo inventó el método para demostrar de una forma rigurosa el cálculo de áreas y volúmenes utilizando aproximaciones sucesivas.
- (300 a.C.): Euclides inventó el algoritmo para calcular el máximo común divisor de dos números, n y m . MCD (n, m).
- (222 a.C.): a Apolonio de Pergeo se le debe el tratado sobre las funciones cónicas y los nombres de la elipse, la parábola y la hipérbola.
- (1170-1250 d.C.): Leonardo de Pisa o Leonardo de Fibonacci generó el algoritmo para la sucesión de los números de Fibonacci, que son 1, 1, 2, 3, 5, 8, 13, ...
- (1550-1617 d.C.): John Napier descubrió los logaritmos.
- (1596-1650 d.C.): René Descartes creó la Geometría Analítica.
- (1623-1662 d.C.): Blaise Pascal inventó la rueda de Pascal o pascalina, considerada una de las primeras calculadoras.
- (1642-1727 d.C.): Isaac Newton desarrolló el teorema del Binomio o el Coeficiente Binomial.
- (1646-1716 d.C.): Gottfried Leibniz creó el concepto de arreglo o matriz; porque fue el primero en organizar los coeficientes de un sistema de ecuaciones lineales en forma de filas y columnas.
- (1736-1813 d.C.): Joseph Louis de Lagrange en su obra *Mecánica Analítica* creó las ecuaciones de Lagrange para sistemas dinámicos.
- (1749-1827 d.C.): Pierre Simón Laplace escribió la obra *Teoría Analítica de las Probabilidades* (1812).
- (1815-1864 d.C.): En la obra *Investigación de las Leyes del Pensamiento* (1854) George Boole su hace un aporte estructural; creó el Álgebra booleana (1848).
- (1862-1943 d.C.): David Hilbert introdujo la noción del espacio de Hilbert, que es uno de los pilares del análisis funcional.
- (1903-1957 d.C.): John von Neuman contribuyó con el desarrollo de los primeros computadores electrónicos.
- (1903-1995 d.C.): Alonzo Church, uno de los más importantes teóricos de la Ciencia de la Computación en el siglo XX, a quien se le debe el concepto del Cálculo de Lambda, definió la idea de “Calculabilidad Efectiva”.

1.2. Concepto de algoritmo

- (1912-1954 d.C.): Alan Mathison Turing introdujo el concepto de la máquina de computación abstracta denominada la Máquina de Turing.
- (1924-d.C.): Donald L. Shell creó el algoritmo de ordenamiento Shell (1959).
- (1938-d.C.): Donald Knuth creó la obra *El arte de Programar Computadores* e hizo uno de los mayores aportes en la Ciencia de la Computación, realizando trabajos importantes en las áreas de Análisis de Algoritmos y Compiladores.

Adicionalmente, se presentan algunos de los científicos que mayores aportes han realizado:

- Wirth Niklaus (1934-d.C.), científico suizo de la computación, quien se desempeñó como jefe de Diseño de los lenguajes de programación identificados como Euler, Algol W, Pascal, Módulo I, Módulo II y Oberón.
- Charles Leiserson (1953-d.C.), científico informático de los Estados Unidos cuyos trabajos en algoritmia han sido relevantes, particularmente en las áreas de la computación paralela y distribuida.
- El algoritmo de Sethi-Ullman (1970-d.C.), llamado finalmente el algoritmo de Ravi Sethi y Jeffrey D. Ullman, crearon en el área de compiladores el algoritmo para traducir los árboles del análisis sintáctico a códigos de máquina.
- Michael R. Garey y David S. Johnson autores del importante libro *Computers and Intractability. A guide to the Theory of NP-Completeness*, el cual contiene la teoría para el tratamiento de problemas NP-Complejos y clasificación de los tipos de problemas NP-Complejos en las áreas de la Ciencia de la Computación relacionadas con Teoría de Grafos, Diseño de Redes, Autómatas y Teoría de Lenguajes entre otras.

Concluida la síntesis histórica del desarrollo de los algoritmos, la siguiente sección desarrolla el concepto de algoritmo.

1.2.2 Algoritmo

Un *algoritmo* es un conjunto finito de reglas bien definidas en su lógica de control que permiten la solución de un problema en una cantidad finita de tiempo. En la resolución del problema con las reglas mencionadas, el algoritmo realiza un conjunto de pasos cuya ejecución para dar la solución del problema puede ser ejecutada manualmente, mecánicamente o utilizando una máquina de procesamiento electrónico de datos.

Al estar conformado por un conjunto finito de reglas, el algoritmo tiene: i) una(s) regla(s) de entrada(s) al algoritmo o punto inicial del algoritmo; ii) una o un conjunto de reglas intermedias (reglas de proceso o de cálculos); iii) un conjunto o una regla de

finalización del algoritmo, que aseguran su terminación, punto en el cual se obtiene una respuesta útil derivada de la ejecución del conjunto de reglas del algoritmo, y iv) exactitud, en el proceso lógico de organización de las reglas a fin de dar respuesta al usuario, o lo que es equivalente: “Un algoritmo se espera resuelva un problema” (Joyanes, 2008, p. 13) de una forma correcta. Si el algoritmo tiene un conjunto infinito de reglas, implicaría que su construcción o está por terminarse pero el número de sus reglas no es mensurable o su construcción es tan compleja que requeriría una escritura infinita de reglas, lo cual no es útil para el usuario en términos de la resolución de problemas.

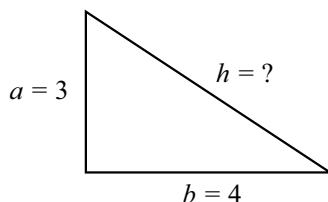
El algoritmo en la ejecución del conjunto de reglas utiliza un tiempo finito entre la regla de entrada del algoritmo y la ejecución de su regla final; lo cual implica que el tiempo de demora del algoritmo es mensurable; luego, si el tiempo de demora del algoritmo no se puede acotar, o lo que es equivalente, es infinito, el algoritmo y sus reglas no tendrían un sentido útil para el usuario, por cuanto este no puede esperar un tiempo infinito en la solución del problema factible de ser tratado a través de la teoría de algoritmos.

Luego, si R_i es el conjunto de reglas lógicas que conforman un algoritmo, acotadas las reglas lógicas del algoritmo por k ($1 \leq i \leq k$), y sea n el número de datos de entrada del algoritmo, entonces el tiempo de demora del algoritmo en función del número de datos de entrada se identifica matemáticamente por $T(n)$. Las R_i reglas que componen el algoritmo deben estar bien definidas y corresponder a una lógica de control algorítmico. Esta lógica de control tiene unos constructores lógicos o estructuras básicas para la construcción de la lógica de control algorítmico. Estas estructuras, al estar bien definidas, permiten la obtención de resultados (cálculos aritmético/lógicos) y, consecuentemente, la solución del problema. La organización de las estructuras básicas mencionadas crean dependencias funcionales entre sí; en tal sentido, no se pueden obtener unos resultados sin los datos de entrada (materia prima de la solución de un problema), como tampoco se tendrían resultados sin reglas que conformen el proceso de transformación de los datos mencionados en información de resultado para el usuario.

Ejemplo 1.1 Cálculo de la hipotenusa de un triángulo rectángulo

Sean a y b los lados de un triángulo rectángulo, cuyos valores son $a = 3$ y $b = 4$, diseñar un conjunto de reglas algorítmicas para calcular la hipotenusa del triángulo.

Análisis:



1.2. Concepto de algoritmo

Con base en el teorema de Pitágoras, la hipotenusa de un triángulo rectángulo es igual a $h^2 = a^2 + b^2$.

Solución:

Primero se necesita una regla mediante la cual obtengamos los datos de entrada. Entonces R_1 es igual a obtener a y b .

Posteriormente se hace necesario una segunda regla que calcule el valor de la hipotenusa, de acuerdo con la ecuación de cálculo. Luego, como $h^2 = a^2 + b^2$, entonces $h = \sqrt{a^2 + b^2}$, por lo tanto: $R_2 =$ Calcule el valor de la hipotenusa h que es igual a la raíz cuadrada de a elevado al cuadrado más b elevado al cuadrado ($h = \sqrt{a^2 + b^2} = \sqrt{3^2 + 4^2} = \sqrt{25} = 5$).

Adicionalmente es necesaria una tercera regla que muestre el resultado del cálculo. Luego, $R_3 =$ Muestre el valor de la hipotenusa (h) calculada.

Finalmente, organizamos de una forma lógica las reglas definidas anteriormente. Es decir, su ejecución manual (la cual es equivalente a lo realizado por un computador) es $R_1 \rightarrow R_2 \rightarrow R_3$. Si y solo si disponemos la secuencia de reglas de esta forma logramos el cálculo del valor de la hipotenusa. Por lo tanto, la secuencia de reglas R_1, R_2, R_3 es un algoritmo. La sencillez en el cálculo de la hipotenusa con los datos dados contrasta con la complejidad de su cálculo cuando los lados a y b del triángulo rectángulo tienen una longitud de uno, cuyo cálculo es

$$h = \sqrt{a^2 + b^2} = \sqrt{1^2 + 1^2} = \sqrt{2} = 1.41421356237309 \dots$$

Lo anterior induce que de acuerdo con los primeros datos de entrada ($a = 3$ y $b = 4$), la respuesta del algoritmo es exacta; pero no existe un algoritmo que pueda dar en notación decimal una respuesta exacta de la $\sqrt{2}$, por cuanto su secuencia de decimales es irrepitible e infinitamente larga.

La dependencia funcional de las reglas dadas induce que R_3 depende para dar los resultados funcionalmente de R_2 ; a su vez, R_2 depende funcionalmente de R_1 para poder calcular los valores de la hipotenusa en función de la raíz; pero finalmente el resultado o información de salida, en cuanto a la ejecución de la primera regla (R_1), también depende de los datos de entrada.

La R_i reglas mencionadas en la resolución del problema dado son las estructuras lógicas básicas que hacen que el cálculo de valor de la hipotenusa sea posible; o lo que es equivalente, son los constructores lógicos de entrada u obtención de datos (R_1), cálculo o proceso de datos (R_2), y finalmente salida de la información de la hipotenusa (R_3).

1.3 Concepto de sistema

1.3.1 Sistema

El concepto de *sistema* se desarrolla en el marco de la Teoría General de Sistemas (TGS). La TGS centra su fundamento en un enfoque multidisciplinario que sirve para estudiar las propiedades de las entidades del universo. Una “entidad” es algo identificable cuyas características se pueden definir en términos de la racionalidad humana y es entendible por sus propiedades para la comprensión del hombre. Este término fue acuñado en el siglo XX con el trabajo de Ludwing von Bertalanffy, a cuyo nombre se deben las investigaciones de sistemas abiertos. Con base en lo anterior, una entidad es una organización (el Estado, la universidad, la familia) que conforma una estructura social. El funcionamiento de las estructuras sociales se basa en personas (hombres y mujeres con consciencia racional e inteligentes) que dentro de la organización desempeñan unas funciones de una forma organizada según la estructura de la organización; y en tal sentido son funcionarios que dentro de la estructura del Estado realizan funciones de gobierno y defensa de un país; o dentro de la organización universidad hacen el trabajo académico, administrativo y de investigación del colectivo que se une en la diversidad; o en la familia desempeñan funciones de rol de padre y de madre, quienes conjuntamente guían a los hijos.

Las organizaciones, por lo tanto, dependen de las personas para su funcionamiento; y a su vez, las personas, como seres animales racionales, dependen de las plantas para la supervivencia de sus células biológicas, siendo la célula un sistema abierto.

Se identifica un ordenamiento jerárquico con complejidad decreciente en las entidades anteriormente relacionadas, lo cual se concreta en unidades de información, a saber:

1. Sistema Social: La entidad “organización” tiene un NIT dentro de la sociedad. Ej. NIT de la Universidad.
2. Sistema Humano: Las entidades “el hombre” y “la mujer” se identifican por su nombre en cualquier sociedad, lo cual se complementa con la cédula de ciudadanía, los números del seguro social o los números de los pasaportes. Ejemplo: Pedro Jesús Blanco, con cédula de ciudadanía 72’425.777; y Debra Kimberly Hausser, con número de pasaporte AFDK-1418-20.
3. Sistema Genético Vegetal: La entidad “planta”, la cual puede ser clasificada según varios sistemas de clasificación, como son los casos de los sistemas de clasificación de angiospermas de APG¹. Ejemplo: clasificación del olivo o aceituno como planta oleaginosa.

¹“Angiosperm Phylogeny Group” (APG). Disponible en:
http://es.wikipedia.org/wiki/Clasificaci%C3%B3n_de_los_organismos_vegetales

4. Sistema Celular: La entidad “célula”, la cual puede ser clasificada en células animales y vegetales. Ejemplo: célula eucariota, aquella que teniendo un núcleo, contiene su información genética en “una doble membrana, la envoltura nuclear la cual delimita un núcleo celular”.²

Las entidades anteriormente relacionadas tienen unidades de información, las cuales se pueden expresar tanto sintáctica como semánticamente a través de números y letras. Ejemplo: código numérico de un alumno de una universidad: 20110213044, cuya semántica se puede interpretar como 2011, año de ingreso, 02, ingreso en el segundo semestre, 13 es el número de la unidad académica a la cual pertenece o Programa de Ingeniería Industrial, y finalmente 044 es el alumno posicionado 044 en la secuencia de ingreso al programa mencionado.

Teniendo en cuenta el marco de la Teoría General de los Sistemas, se intentará concretar algunas definiciones del término “sistema”.

Con base en los objetos como partes componentes de un sistemas y los atributos asociados a los objetos, Hall (Johansen, 1992, p. 55)

(...) define un sistema como un conjunto de objetos y sus relaciones, y las relaciones entre los objetos y sus atributos.

Una definición a nivel genérico del concepto de sistema es:

... Cualquier o alguna cosa del mundo real capaz de ser vista como un sistema. Cualquier referente de sistema debe ser capaz de ser representado como un todo o como una colección de partes relacionadas. Cualquier parte de un sistema debe ser capaz de ser vista como un sistema en sí mismo. Igualmente, un sistema debe ser capaz de ser absorbido por un sistema mayor como una de sus partes. (Myers & Kaposi, 2004, p. 38)

Desde el punto de vista del tipo de sistema social, económico, ambiental, técnico o informático:

... Sistema se refiere a sistemas técnicos tales como un sistema de transmisión electrónica. Algunas veces se refiere un sistema como interactuando con componentes, algunas de las cuales no son técnicas. Sistema dirigido principalmente en la interacción de componentes institucionales y técnico... Sistema entonces significa la interacción de componentes de diferentes formas, tales como técnicas o institucionales, también como diferentes valores. (Coutard, 1999, p. 248)

El término “sistema” puede ser referido también a la confiabilidad en procesos predictivos, cuando el sistema está integrado por componentes; una de los cuales puede ser un conjunto de algoritmos que soportan el software de un proceso determinado, y se expresa el concepto de sistema como

²Célula eucariota. Disponible en: http://es.wikipedia.org/wiki/C%C3%A9lula_eucariota

Cualquier equipo / producto compuesto de unidades de subsistemas / componentes, los cuales deben ser especificados por sus configuraciones funcionales y límites. Esta información ayudará a hacer los diagramas de bloques de los productos/equipos funcionales para evaluar la confiabilidad. La falla de los componentes en un bloque hará que el sistema falle dependiendo de su configuración. (Mishra & Sandilya, 2009, p. 117)

Los diferentes enfoques del término “sistema” hace que su variabilidad conceptual, por su polisemia, sea difícil de definir concretamente; luego, el concepto de sistema que se maneja en este libro, dentro de un enfoque informático soportado por la teoría de algoritmos, es el siguiente:

Un sistema es un conjunto de entidades que interactúan entre sí de una forma organizada, con el fin de desempeñar una función y generar una información útil al usuario, dentro de unos límites de operación y funcionalidad.

Se resaltan en la definición las siguientes categorías conceptuales: *i*) Conjunto de entidades: las entidades son las partes componentes del sistema, lo cual genera los conceptos de supra sistemas, sistema y subsistemas. Ejemplo: Suprasistema “carro”; integrado por los sistemas mecánico, neumático y eléctrico; a su vez, el subsistema mecánico, compuesto por el motor, la dirección, el acelerador y los frenos del carro. *ii*) Interacción de entidades: la entidades dentro de los sistemas generan procesos de conectividad y comunicación; dichos proceso hacen que los sistemas sean estáticos o dinámicos. Una estructura geométrica, como es el caso de un cubo sobre una mesa en la tierra, por la gravedad tiende a ser un sistema estático, y una estructura esférica sobre un plano inclinado en un ángulo de 45 grados tiende a ser dinámica. *iii*) Interacción organizada: la organización es contraria al caos; y es precisamente la interacción de una forma controlada de las partes de un sistema la que genera información. En el suprasistema “carro” mencionado, el punto de apoyo del carro es el sistema neumático; ahora, el sistema neumático ha de interactuar de una forma controlada con el sistema eléctrico, el cual permite el arranque del carro, y a su vez, dichos sistemas deben interrelacionar de una forma organizada con el sistema mecánico, generando así el sistema de desplazamiento de vehículo. Contraria a la organización antes mencionada, si el carro en su sistema de desplazamiento, estando en bajada, se arranca en una quinta velocidad y a una máxima aceleración, el sistema puede salirse de control. *iv*) Generar información útil al usuario: son los resultados del sistema para el cual fue concebido o por lo cual interactúan sus partes empleando unidades energéticas y espacio temporales, entre otras. En tal sentido, dentro del sistema de desplazamiento del carro, cuando el vehículo está en movimiento, la utilidad del sistema es la movilidad para el usuario de un punto geográfico inicial a un punto final; pero, a su vez, se generan en el desplazamiento otras unidades de información, tales como: la velocidad del carro en su desplazamiento, número de kilómetros/hora; el consumo de gasolina número de galones/kilómetro; la temperatura del motor, entre otras informaciones. *v*) Contorno del sistema: de acuerdo con los resultados de información, el sistema debe funcionar en un entorno determinado o área de operación; ello hace que el sistema tenga unos límites preestablecidos, contorno que diferencia un sistema de otro, pero, a su vez, es el punto de interacción

1.3. Concepto de sistema

entre subsistemas; así, el sistema “neumático” (llantas) tiene su contorno, muy diferente al sistema mecánico (motor); pero ambos, llantas y motor, interactúan para darle al carro el desplazamiento. Algorítmicamente, es necesario primero accionar el sistema eléctrico del carro, que enciende el motor, para poder accionar a través del eje de transmisión del carro la energía que da fuerza de desplazamiento a las llantas, para que finalmente estas sean controladas por el acelerador y el freno del carro.

Una de las características fundamentales de la definición de sistema es la generación de información. La información es una cantidad mensurable que debe ser medida en una escala de valoración cuantitativa o cualitativa. La valoración de la información requiere de un procesamiento de las unidades primarias de información, que son los datos; por lo tanto, el concepto de información se asocia también a un conjunto organizado de datos derivados de un sistema en operación. Organizar los datos y evaluarlos requiere necesariamente de un sistema de posicionamiento o de clasificación de datos y de un sistema de valoración de los mismos. En el sistema digital, los números en el intervalo cerrado $[0;9]$ son base para los sistemas de codificación; un sistema de codificación digital de k dígitos ($k = 4$) y de n códigos ($n = 5$) requiere para su organización posicionamiento; en tal sentido, los códigos digitales representativos de las referencias de partes de una compañía de producción especificados como 4584, 9872, 3210, 1400, 7777 son fácilmente valorables en la escala digital por su posicionamiento, de lo cual se puede concluir que la menor referencia es la 1400, y la mayor referencia es la 9872, soportada la conclusión por el sistema de posicionamiento de las referencias en su orden: 1) 1400, 2) 3210, 3) 4584, 4) 7777 y 5) 9872. Las referencias fueron fácilmente organizadas para la compañía referidas a un sistema de inventarios; pero, si tanto los códigos del sistema de codificación aumentan, supuesto $k \geq 10$ dígitos por referencia, y el número de dígitos aumenta, supuesto $n \geq 1$ millón, entonces la organización de los datos vía a convertirlos en información (referencias clasificadas) útil al usuario requiere necesariamente de un sistema de procesamiento electrónico de datos basado en un computador; para ello se requiere entonces construir un algoritmo que ordene las referencias del sistema de inventarios mencionado.

1.4 Concepto de algoritmo en el marco de la Lógica

Teniendo como fundamentos que la lógica es entendida como un conjunto de reglas que conforman un sistema de razonamiento y, a su vez, siendo el algoritmo un conjunto finito de reglas bien conformadas en su lógica de control y factibles de ser automatizadas mediante un lenguaje de programación, el análisis de la relación entre algoritmo y lógica se basará en la Teoría de la Información; justificado por el hecho de que es precisamente la operación de la lógica de control del algoritmo la que permite el procesamiento de los datos para generar información y, en últimas, conocimiento útil en el contexto de producción de un sistema informático.

La información es un recurso de decisión en la sociedad digital, y sirve para desa-

rollar el conocimiento humano. El desarrollo histórico del lenguaje como medio de comunicación entre humanos y vía de interacción en la relación hombre-computador, ligado al desarrollo histórico de la teoría de alfabetos, ha hecho posible que los fenómenos del mundo real sean representados en el mundo imaginario del hombre mediante símbolos. Un símbolo, en el contexto del lenguaje, es una secuencia de caracteres con una sintaxis y una semántica asociada que pertenecen a un alfabeto. Así, la secuencia 1010 es una sarta de dígitos binarios cuyos caracteres pertenecen a un alfabeto definido como el conjunto de imágenes pictográficas llamadas “símbolos, vocabulario, o cualquier conjunto finito de símbolos” (Hopcroft & Ullman, 1969, p. 1); secuencia de bits 1010 que pertenecen entonces al alfabeto binario o en base dos igual a $b_2 = \{0, 1\}$ en su estructura sintáctica; y en su semántica o significado, de acuerdo con el sistema posicional binario, corresponde al número decimal $(12)_{10}$. Con base en lo anterior, la palabra “algoritmo” es una sarta de nueve símbolos, compuesta por cuatro vocales y cinco consonantes que pertenecen al alfabeto latino.

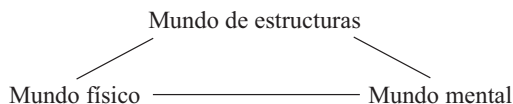
La representación en el mundo real de un algoritmo a nivel fáctico es un conjunto de reglas lógicas. Las reglas lógicas que componen el algoritmo forman un imaginario en la mente del sujeto o alumno. En el imaginario del sujeto, es la organización de la lógica propia de la mente de este la que permite simbolizar la solución de un problema determinado utilizando la teoría de algoritmos. En tal sentido se unen dos sistemas de operación: por un lado, las reglas lógicas que sirven para construir los algoritmos y, por otro lado, las reglas lógicas (desarrolladas o no) presentes en la mente del sujeto. De resultar, si el sujeto tiene una estructura de pensamiento formal, que le sirve para pensar en complejos, es más fácil para el aprendiz de algoritmos construir rápidas y efectivas soluciones algorítmicas. Ello indica que la lógica formal tiene una estrecha relación con la lógica algorítmica; y a mayor nivel de desarrollo de la lógica formal del sujeto se generará un mayor nivel de estructuración formal en las reglas algorítmicas y, consecuentemente, una mayor posibilidad de automatización de un problema a ser solucionado con base en la algoritmia y la computación.

El paso del mundo real (reglas algorítmicas) al mundo imaginario (lógica del constructor del algoritmo) del sujeto produce como consecuencia la representación simbólica de la solución del problema en su mente a través de su lógica. La representación simbólica del algoritmo en la mente del sujeto no es aún la solución del problema a nivel informático utilizando la teoría de algoritmos; hasta ahora se ha cubierto la etapa de la solución del problema a través de la lógica humana, cuya representación simbólica es la lógica algorítmica. La lógica algorítmica, concretada en un algoritmo, necesita la utilización de un lenguaje de programación, con el fin de que la representación del algoritmo pueda codificarse en un conjunto de símbolos que pertenecen no ya a un lenguaje humano, sino a un lenguaje de programación, que puede ser ejecutado en una máquina de procesamiento electrónico de datos o computador. Este es precisamente el punto en el cual la lógica humana al interactuar con la lógica algorítmica se convierte en lógica de programación, justificando así la relación entre algoritmo y lógica; y llegando así a convertirse en lógica de programación.

El análisis anterior está soportado por la llamada “Triada existencial del mundo”

1.4. Concepto de algoritmo en el marco de la Lógica

(o estructura global del mundo), que tiene la forma



(Burgin, 2009, p. 60).

El mundo de las estructuras contiene sus tipos, tales como: estructuras algorítmicas, estructuras de datos, estructuras de lenguajes de programación, entre otras, en un contexto informático. Con el conglomerado del mundo de estructuras se modela el mundo físico; o sea, en el mundo físico se encuentra tanto el problema que se va a solucionar como las reglas algorítmicas, ambas son fácticas. La triada se cierra cuando las estructuras algorítmicas solucionan el problema al pasar por el mundo mental del alumno en su lógica humana racional, regresando al mundo de las estructuras organizadas, donde se convierten en estructuras lógicas algorítmicas. El calcular el problema de la función factorial ($n!$) en su raíz pertenece al mundo de las estructuras matemáticas; ello indica que el factorial es una función representada por $!$, cuyos elementos matemáticos son el 1 como módulo de la multiplicación y el \times como operador de la operación de multiplicación. Con base en la estructura matemática de soporte a la función mencionada, para calcular la función del mundo real $n!$ (de aplicación en el coeficiente binomial $(n!/(k!(n-k)!))$) utilizando algoritmos, en el mundo real se encuentran las estructuras de control algorítmico; estas estructuras indican en la realidad que si el sujeto en su mundo mental organiza una secuencias de multiplicaciones desde un punto inicial (1) hasta un punto final (5) en la búsqueda de la respuesta del $5!$, obtiene la respuesta de la función para dicho valor, o sea 120; ya en este punto del resultado se regresa al mundo de las estructuras, lugar en el cual se encuentran las estructuras lógicas algorítmicas o de lógica de programación si el algoritmo se codifica. Se enfatiza que si la mente del sujeto no logra integrar las estructuras matemáticas con las estructuras de control algorítmico a través de su mundo mental, puede no tener una respuesta correcta.

Es de la mayor importancia resaltar que concordando con la triada anteriormente expuesta, en la mente del sujeto o en su mundo imaginario también está la triada “real - imaginario - simbólico” (Burgin, 2009, p. 67); luego, lo que realmente genera la lógica del sujeto o su sistema de razonamiento es la representación en símbolos algorítmicos de la resolución del problema.

Con base en los tres mundos mencionados y la triada referenciada, la estructura algorítmica (mundo físico) en la mente del sujeto (mundo mental) debe regresar al mundo de las estructuras (estructuras de control algorítmico base para las estructuras de lenguajes de programación (lenguaje C o Java)), con el fin de que el algoritmo sea codificado en un lenguaje de programación. Las reglas algorítmicas convertidas a un lenguaje de programación se identifican como un programa de computador; y este código de programación al correrse o funcionar en un computador es el que produce información; o sea, da el resultado al usuario en el sentido de que $5! = 120$.

Capítulo 1. Lógica de programación

Este proceso corresponde a la transformación de los datos en información (dato 5, que al aplicar el factorial es información 120). Esta información ya hace parte del conocimiento humano, que se puede utilizar para una determinada decisión, como es el cálculo del Coeficiente Binomial $\binom{n}{k} = n!/k!(n-k)!$, que representa el número de subconjuntos que se pueden formar a partir de un determinado conjunto.

En tal sentido, los datos son materia prima para producir información; y la información es la base para producir conocimiento. El conocimiento como tal no se genera del simple procesamiento del dato a través del algoritmo; pues en su procesamiento, el dato es influenciado no solo por los operadores aritmético/lógicos dentro del flujo de control del programa cuando se ejecuta en el computador que transforman el dato en información, sino el dato procesado que ya es información también recibe la influencia de la(s) acción(es) de la información adicional que se le agregue a los resultados dados por el programa de computador, y que completa la interpretación del usuario de la información dada por la máquina en las estructuras de conocimiento que pertenecen al usuario en la interpretación del proceso completo de transformación del dato a información y de información a conocimiento humano.

Se justifica lo anterior por el hecho de que

Datos, bajo (acción) la influencia de información adicional, llegan a ser conocimiento (Burgin, 2009, p. 200).

Por lo tanto, únicamente si se genera información los datos son convertidos en conocimiento; más aun, si y solo si la información generada por el programa es validada en la condición de verdad a través del sistema de razonamiento del usuario, la información llega realmente a ser útil al usuario, y toda la energía gastada en la resolución del problema a través de la construcción del algoritmo y la codificación del programada no se ha perdido tanto en tiempo como en esfuerzo.

El análisis del concepto de algoritmo en su relación con la lógica conduce a identificar que la resolución de problemas factibles de ser tratados por máquinas de procesamiento electrónicas de datos o computador inicia en el sistema de razonamiento humano, pero también termina en el conjunto de reglas del razonamiento humano. Las reglas algorítmicas y de programación, que generan la lógica de programación, son solo un paso, en este caso electrónico, para lo cual el computador es una herramienta de cálculo, pero la esencia de la resolución de problema es en sí la lógica humana.

Una vez desarrollados los conceptos de algoritmo, sistema y el concepto de algoritmo en el marco de la lógica, el paso siguiente es explorar conceptualmente las máquinas de procesamiento electrónico de datos.

1.4. Concepto de algoritmo en el marco de la Lógica

1.5 Máquinas de procesamiento electrónico de datos

1.5.1 Sistema Computacional

Teniendo en cuenta que un sistema es un conjunto de entidades que al interrelacionar entre sí de una forma controlada dan una respuesta de utilidad para el usuario, el sistema computacional se basa en un conjunto de partes (el computador y la red) que interactúan de una forma organizada para dar resultados al usuario. Luego, un sistema computacional basado en el computador posibilita los procesos de conectividad, de comunicaciones y de cálculo de símbolos pertenecientes a un alfabeto.

1.5.2 Concepto de Computador

Es una máquina de procesamiento electrónico de datos, que permite recibir datos de entrada, procesar los datos recibidos con acciones aritméticas y lógicas para dar como resultado una información útil al usuario.

El procesamiento de los datos requiere una lógica de programación, la cual se basa en estructuras lógicas que se representan mediante los algoritmos. Estos algoritmos se codifican en un lenguaje de programación que genera un programa (Código C, Java), y los programas se corren en un computador con el fin de obtener la respuesta requerida por el usuario.

El término “computador”, “computadora” u “ordenador” se asocia también a las siguientes definiciones:

Una computadora es un dispositivo electrónico, utilizado para procesar información y obtener resultados, capaz de ejecutar cálculos y tomar decisiones a velocidades millones o cientos de millones más rápidas que puedan hacerlo los seres humanos. (Joyanes, 2008, p. 4).

Una computadora es una máquina y, como otras máquinas, como un automóvil o una podadora, debe encenderse y luego conducirse, o controlarse, para hacer la tarea que se pretende realizar. En un automóvil, por ejemplo, el control es proporcionado por el conductor, quien se sienta en su interior y lo dirige. En una computadora, el conductor es un conjunto de instrucciones llamado programa. (Bronson, 2007, p. 2).

Es importante destacar en tales definiciones las siguientes categorías conceptuales con relación al término “computador”: *i*) Es una máquina o dispositivo, requiere un control de encendido, operación y apagado. *ii*) Necesita un control; a la computadora la controla un programa. *iii*) El programa de computadora permite el procesamiento de lo requerido por el usuario. *iv*) El programa recibe datos, los procesa con operadores aritmético y lógicos, y da la respuesta al usuario convertida en información;

que a su vez, al entrar nuevamente al computador, se vuelve a procesar para generar nuevas informaciones. *v*) Con la información se pueden tomar decisiones; en tal sentido, la información procesada por el computador debe ser útil al usuario. *vi*) El procesamiento requiere de una lógica de programación. *vii*) La lógica de programación se basa en estructuras mentales o lógicas que sirven para diseñar el flujo de control en el programa. *viii*) Las estructuras lógicas que dan control al programa conforman algoritmos. *ix*) Los algoritmos son entonces conjuntos de reglas que realizan cálculos; y *x*) Los algoritmos requieren de un lenguaje de programación para ser codificados, y así convertirse en un programa de computadora.

El computador, por ser una máquina, requiere de un *hardware*, y para su funcionamiento requiere de un *software*. El hardware son las componentes metálicas del computador y el software es el lenguaje de programación que hace que el computador funcione. El computador funcionando en el interior de un sistema computacional cuando hace procesos de conectividad y comunicaciones necesita el *comware*. Este es una combinación de elementos que permiten al computador conectarse y comunicarse formando una red, con el fin de transmitir información entre un computador fuente (o el generador de la información) y un computador destino (o el receptor de la información), como es el caso de la red Internet.

1.5.3 Arquitectura de un computador

La computadora como máquina o dispositivo con base en sus funciones de entrada (E), proceso (P) y Salida (S) se compone de tres partes:

- Las unidades de entrada (U/E), tales como el teclado, el ratón (*mouse*) o la pantalla táctil.
- La unidad central de procesamiento (UCP o CPU (*Central Processing Unit*)).
- Las unidades de salida (U/S), como el caso de la pantalla del computador o la impresora.

La CPU (abreviación de *Central Processing Unit*) es la unidad que desempeña la función primaria de un sistema computacional; está compuesta por la unidad de control, la memoria y la unidad aritmético/lógica.

La unidad de control es la parte del computador digital que realiza la secuencia de operaciones e interrupciones de las instrucciones de un programa codificadas en un lenguaje de programación; y a su vez envía las señales apropiadas a los circuitos del computador con el fin de realizar las operaciones aritméticas o lógicas necesarias en la ejecución de las instrucciones del programa almacenado en la memoria del computador.

La memoria es el dispositivo de almacenamiento utilizado por el computador digital con el fin de albergar en ella los datos de entrada, los programas y los resultados. La memoria tiene dos características principales: las direcciones de memoria

1.5. Máquinas de procesamiento electrónico de datos

y el contenido que se almacena en las direcciones de memoria. Luego, similar a una distribución urbanística de una ciudad: en la dirección Calle 96 n.º 51-32 se encuentra viviendo José Jesús Estrella, siendo la persona en mención el contenido de identidad digital de la vivienda en mención.

Se debe tener en cuenta que existen dos tipos de memoria: la memoria de acceso random y la memoria de solo lectura. La primera, identificada como RAM por sus siglas en inglés (*R*andom *A*ccess *M*emory), es una memoria volátil o temporal que contiene el computador y se llama “memoria principal”, cuya función es almacenar los datos que se deben procesar, el programa del usuario y los resultados de información obtenidos por el computador para el usuario; los contenidos de la memoria RAM son volátiles, por cuanto se pierden al apagar el computador. La segunda se identifica como ROM, que significa en inglés *R*ead *O*nly *M*emory, o memoria de solo lectura, la cual tiene como principal función almacenar los datos, los programas y los resultados del usuario de una forma permanente, o lo que es equivalente: cuando el computador se apaga las informaciones almacenadas en la ROM no se pierden, siendo, por lo tanto, su almacenamiento permanente.

La unidad aritmético/lógica (U A/L) es la parte del computador que se encarga de hacer las operaciones aritméticas y lógicas requeridas por las instrucciones del programa almacenado en el computador, o lo que es equivalente: “... es la sección de “manufactura” de la computadora” (Deitel & Deitel, 2004, p. 5). La U A/L tiene como función el tratamiento matemático de la lógica formal del programa mediante el uso de un conjunto de símbolos que representan cantidades y relaciones, que en últimas se ejecutan a través de circuitos lógicos que se encargan de hacer las operaciones lógicas propias de la Lógica booleana, tales como la operación AND, OR o la operación NOT, entre otras. Estas operaciones en sus circuitos digitales posibilitan la utilización de dígitos binarios con el fin de ejecutar las instrucciones contenidas en el programa del usuario, y de esta forma resolver a través del programa los problemas de este factibles de tratar computacionalmente a través de una máquina de procesamiento electrónico de datos.

Esquemáticamente, la arquitectura de un computador digital se representa en la figura 1.1.

Con base en la figura 1.1 suponga que el usuario va a realizar la operación matemática $C = A + B$. Esta ecuación matemática para poder ser realizada en una lógica de control algorítmica³ necesita tres primitivas de control: 1) Leer a, b , lleva al computador las variables de entrada A y B . 2) Asignar el resultado a la variable C , que en notación algorítmica es $c \leftarrow a + b$ (lo cual significa que a la variable C se está llevando el cálculo matemático de la suma de $A+B$). 3) Imprimir c , variable que contiene el resultado para el usuario. Una vez se construyen las tres primitivas de control, que forman el algoritmo para sumar dos números, el algoritmo se codifica en un lenguaje de programación (C, C++, Visual C++, Java, ...), y se tiene el programa de sumar dos números en un computador digital. Cuando el programa se

³Las estructuras de la lógica de control algorítmica se desarrollan en el capítulo 2 de este texto, con el nombre de “Estructuras básicas para la construcción de algoritmos”.

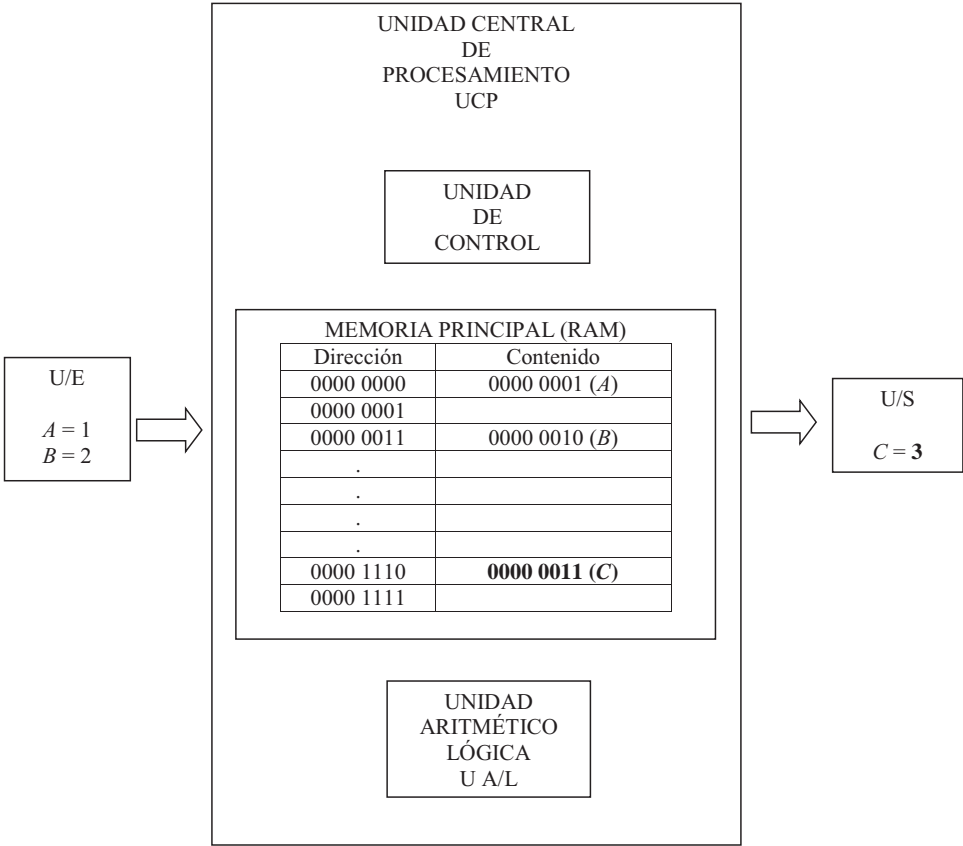


Figura 1.1 Arquitectura de un computador

1.5. Máquinas de procesamiento electrónico de datos

corre en el computador, se ejecuta de la siguiente forma: 1) Se ejecuta primero la lógica de control de lectura; o sea, lee las variables A y B ; suponga que los valores de las variables sean $A = (1)_{10}$ en base decimal y $B = (2)_{10}$, también en base diez; entonces el computador convierte el contenido de las variables dadas en base decimal (10) por el usuario a contenidos en base binaria o base dos (2); luego $A = (1)_{10}$ es igual a $(0001)_2$ y $B = (2)_{10}$ es igual al valor $(0010)_2$ en base binaria. Se debe tener en cuenta que los datos se almacenan en las direcciones de memoria del computador; para el caso de la figura 1.1, la variable A = está almacenada en la dirección binaria de la memoria igual a 0000 y la variable B = está contenida en la dirección de memoria igual a 0011.

Teniendo en cuenta los datos contenidos en las variables almacenadas en mención, la unidad aritmético/lógica realiza los cálculos matemáticos; o sea, suma en binario los valores $0001 + 0010 = 0011$ en base dos y asigna el resultado a la variable C ; se debe tener en cuenta que la variable C , que contiene el resultado del usuario, se encuentra en la dirección de memoria 1110. Finalmente, la primitiva de control Imprima C realiza para el usuario la operación de convertir el contenido de la variable $C = (0011)_2$ de base dos, lo cual es igual a $(3)_{10}$ en base diez, que es el resultado o información útil al usuario que da como resultado el programa de computador ejecutado en el computador; o lo que es equivalente, la unidad de salida, consola del computador, dará como resultado el número decimal **3**, que ya es **información**.

Teniendo en cuenta que para producir la información el computador tiene que transformar el código fuente o el programa escrito en un lenguaje de programación, que es la instrucción en lenguaje C igual a $c = a + b$, instrucción que se transforma a lenguaje objeto o lenguaje binario ($0011 = 0001 + 0010$) en el computador, se hacen necesarios los conceptos de almacenamiento de la información en el computador y codificación de la información, lo cual se desarrolla a continuación.

1.5.4 Almacenamiento de datos en el computador

La unidad básica de almacenamiento de los datos para producir información en un computador es el *bit*. El término “bit” es la abreviatura de “dígito binario” (en inglés *binary digit*) y es una unidad de información correspondiente a una decisión binaria. La decisión binaria tiene dos estados: el estado uno, o sea, el bit es igual a 1, en cuyo caso se asocia con la decisión de SI, o lo que es equivalente, el encendido en un sistema de control de un dispositivo. El estado dos de la decisión mencionada es igual a cero (0), a cuyo estado se le asocia la decisión NO, lo cual significa que el estado del dispositivo es apagado si el bit está referido a su sistema de control de encendido o apagado.

La unión de una secuencia de ocho bits recibe el nombre de “byte”. Luego el byte es un término que indica una secuencia de bits consecutivos tratados como una única entidad de información. La unión de varios bytes (cada uno de ellos con secuencias de 8 bits) recibe el nombre de “palabras en código binario”. Luego una palabra de 2 bytes es igual a 16 bits, que representados en cuartetos de bits para

$\text{Palabra}_{4 \text{ bytes}} = \overset{b_7}{1001\ 0010}_{\text{byte1}}\ \overset{b_0}{1100\ 0011}_{\text{byte2}}\ 1001\ 0010_{\text{byte3}}\ 1100\ 0011_{\text{byte4}}$

Donde b_7 es el octavo bit del primer byte contado a partir del bit cero b_0 de derecha a izquierda (\leftarrow)

Con base en las figuras 1.1 y 1.2, la memoria principal (RAM) de un computador es un conjunto de celdas organizadas de una forma contigua, cada una de las cuales tiene una sola dirección de memoria que identifica la celda, y en la cual se va a almacenar la información. Luego, teniendo en cuenta lo anterior, la capacidad de memoria de un computador puede contener millones de bytes. Estas capacidades de almacenamiento se miden utilizando los múltiplos del byte, que sin hacer una enumeración exhaustiva son kilobyte (1 KB = 1024 bytes); megabyte (1 MB = 1024KB), gigabyte (1 GB = 1024 MB) o terabytes (1 TB = 1024 GB).

Para generar la información útil al usuario, el dato se debe almacenar y procesar en el computador; luego, en síntesis, el objetivo fundamental de todos los programas (generados a partir de la construcción de un algoritmo) es almacenar y procesar datos para generar informaciones como resultados para el usuario. Los tipos de datos según el dominio pueden ser: numéricos (dominio del tacto), alfabéticos (dominio de la voz), de audio (dominio auditivo) o de video (dominio de la visión).

Los datos numéricos, alfabéticos o de otro tipo en su significado son realmente átomos unitarios o caracteres que pertenecen a otros tipos sistemas. En el caso de los datos numéricos, se habla de sistemas numéricos binarios $\{1, 0\}$, decimales $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ o hexadecimales $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$; es decir, el sistema binario tiene dos caracteres, 0 y 1. En el caso de datos alfabéticos, se habla del alfabeto latino $\{A, B, C, \dots, Z, a, b, c, \dots, z\}$. Luego, cuando un carácter de un sistema numérico o alfabético se introduce en un dispositivo (U/E o U/S) del computador, los caracteres se codifican.

Mancilla, Herrera, Alfonso. Diseño y construcción de algoritmos, Universidad del Norte, 2014. ProQuest Ebook Central, <http://ebookcentral.proquest.com/lib/bibunlasp/detail.action?docID=4183551>.
Created from bibunlasp on 2018-11-06 10:41:10.

Copyright © 2014. Universidad del Norte. All rights reserved.

Tabla 1.1 Tipos de datos

Dominio Cognitivo	Tipo de dato	Lenguaje C	Ejemplos	Almacenamiento	Intervalo	Declaración de variables en el lenguaje C
Lectura/Escritura Números enteros	Entero Cortos	short int	9, -5	2 bytes	[-32768;32767]	short int suma;
Lectura/Escritura Números enteros	Enteros	int	+98752, -999936	4 bytes	[-2147483648; 2147483647]	int promedio;
Lectura/Escritura Números Reales	Punto flotante o Real	float	+777.77, -0.47	4 bytes	[1.4013E-45; 3.4028E+38]	float gradiente;
Lectura/Escritura Números en Notación científica	Real doble precisión	double	9.4665E+307	8 bytes	[4.9406E-324; 1.7977E+308]	long double vel_estelar;
Lectura/Escritura Letras del alfabeto	Carácter	char	'Z', 'n'	1 byte	256 caracteres	char palabra_clave;
Lectura/Escritura Datos Booleanos	Lógico	Bool	1 o 0	1 byte	Falso = 0; Verdadero = Valor positivo	bool switch_control;

Fuente: adaptado de Bronson (2007, pp. 2-85).

Capítulo 1. Lógica de programación

La codificación de un carácter es la combinación de bits, a través de los cuales se representa dicho carácter. El código de codificación más utilizado es el Código Estándar Americano para el Intercambio de Información, conocido como ASCII por sus siglas en inglés: *American Standard Code for Information Interchange*. En tal sentido, el carácter numérico $(7)_{10}$ en base 10 se representa como $(00000111)_2$ en binario, lo cual corresponde a su código ASCII $(7)_{15}$ en hexadecimal; y la letra mayúscula A corresponde al código ASCII $(65)_{10}$, con su equivalente $(41)_{15}$ en hexadecimal; por lo tanto, la letra A, como carácter, se almacena en el computador como un byte y su contenido es $(01000001)_2$. Los caracteres ASCII con sus equivalentes en binario y hexadecimal se muestran en la tabla 1.2.

Teniendo en cuenta que los datos procesados por el computador son digitales, justificado por el hecho de que su contenido se construye a partir de dígitos binarios, para el correcto entendimiento de los datos que entran, se procesan y se convierten en resultados de información para el usuario es necesario el estudio de los sistemas de numeración, en el marco de la Teoría de la Codificación de la Información.

1.5.5 Sistemas de numeración

La justificación por la cual son necesarios los sistemas de numeración al procesar datos en el computador estriba en que los datos solicitados como entrada en los programas del usuario, y los caracteres numéricos y alfabéticos que se procesan en los programas, son el código fuente del programa. Este código es pasado por el computador a código objeto o en base dos, que es, en últimas, es el lenguaje que maneja la máquina. A su vez, la representación alfabética de los datos es única; pero la representación numérica no lo es; de hecho hay enteros cortos (*short int*), hay enteros (*int*) y hay también reales (*float*); pero a su vez, un número real tiene varias representaciones; por ejemplo, puede expresarse en notación científica. Así, el número 777.48459 es equivalente al número 7.7748459×10^2 (notación científica) o al número 777484.59×10^{-3} .

Los sistemas de numeración binario, octal, decimal y hexadecimal son los más utilizados para la representación de datos en el computador para pasar el lenguaje fuente a lenguaje objeto o base dos. Estos serán desarrollados en esta sección.

Sistema binario: el alfabeto binario está compuesto por dos cifras $\{0, 1\}$. Por lo tanto, un número binario corresponde a una cadena de bits, la cual expresada como $(a_n a_{n-1} a_{n-2} \dots a_0)$ forma un sistema numérico posicional, desde la posición 0 hasta la n -ésima posición de la cadena. Por ejemplo, en la cadena de bits $C = (1010)_2$, su sistema posicional es de cuatro bits, identificados así: $a_{n=3} = 1$; $a_2 = 0$; $a_1 = 1$ y $a_0 = 0$.

La importancia del sistema posicional consiste en que el número C se puede representar mediante la siguiente ecuación:

1.5. Máquinas de procesamiento electrónico de datos

Tabla 1.2 Tabla de caracteres ASCII⁴

ASCII	Hex	Símbolo	ASCII	Hex	Símbolo	ASCII	Hex	Símbolo	Hex	ASCII	Símbolo
0	0	NUL	32	20	(espacio)	64	40	@	96	60	,
1	1	SOH	33	21	!	65	41	A	97	61	a
2	2	STX	34	22	"	66	42	B	98	62	b
3	3	ETX	35	23	#	67	43	C	99	63	c
4	4	EOT	36	24	\$	68	44	D	100	64	d
5	5	ENQ	37	25	%	69	45	E	101	65	e
6	6	ACK	38	26	&	70	46	F	102	66	f
7	7	BEL	39	27	'	71	47	G	103	67	g
8	8	BS	40	28	(72	48	H	104	68	h
9	9	TAB	41	29)	73	49	I	105	69	i
10	A	LF	42	2A	*	74	4A	J	106	6A	j
11	B	VT	43	2B	+	75	4B	K	107	6B	k
12	C	FF	44	2C	,	76	4C	L	108	6C	l
13	D	CR	45	2D	-	77	4D	M	109	6D	m
14	E	SO	46	2E	.	78	4E	N	110	6E	n
15	F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[123	7B	{
28	1C	FS	60	3C	!	92	5C	\	124	7C	
29	1D	GS	61	3D	=	93	5D]	125	7D	}
30	1E	RS	62	3E	¿	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	-	127	7F	□

⁴Fuente original:

<http://www.sitiosargentina.com.ar/categorias/internet/formatos/ASCII.htm>

$$C = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \dots + a_12^1 + a_02^0 = \sum_{i=0}^{n-1} a_i2^i$$

Teniendo en cuenta esta ecuación, la cadena de dígitos binaria $C = (1010)_2$ tiene cuatro cifras binarias; luego a partir de cero, entonces $n = 3$, de derecha a izquierda; entonces el número binario dado es igual a la cantidad $C = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 2 = (9)_{10}$ en base diez. Se debe tener en cuenta que los bits con valor cero no contribuyen en este caso con el posicionamiento de las potencias de base binaria $2^n, 2^{n-1}, 2^{n-2}, \dots, 2^i, \dots, 2^3, 2^2, 2^1, 2^0$.

Con base en lo anterior, si se define la base b de un sistema de numeración y n el número de cifras de un número expresado en la base dada, entonces el número C está dado por la siguiente expresión matemática:

$$C = (a_{n-1} \dots a_1a_0)_b = \sum_{i=0}^{n-1} a_ib^i$$

donde cada uno de los a_i son las cifras del número C y puede ser una de las cifras b_0, b_1, \dots, b_{n-1} que corresponden a la base b .

Se debe tener en cuenta que los sistemas de numeración son bases para la codificación de los datos, y con base en los datos representados en dicha codificación se pueden hacer operaciones. En tal sentido, si se tienen dos operandos binarios, identificados como b_1 y b_2 , si se realiza la operación binaria de la suma de números binarios, entonces la realización de la operación se muestra en la tabla 1.3.

Tabla 1.3 Operación binaria de suma de bits

Operando uno b_1	Operador +	Operando dos b_2	Resultado	Acarreo
0	+	0	0	
0	+	1	1	
1	+	0	1	
1	+	1	0	1

Sistema octal: los dígitos que componen el alfabeto octal es el conjunto $\{0, 1, 2, 3, 4, 5, 6, 7\}$; de este modo, el número $(17)_8$ en base octal es igual a $C = 1 \times 8^1 + 7 \times 8^0 = 8 + 7 = (15)_{10}$.

Sistema decimal: el alfabeto que forma el sistema digital o en base 10 está compuesto por el conjunto de dígitos $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$; utilizando el sistema de posicionamiento decimal de derecha a izquierda sin hacer un enumeración exhaustiva de posiciones, un número está compuesto por unidades ($1 = 10^0$), decenas ($10 = 10^1$), centenas ($100 = 10^2$), unidades de mil ($1000 = 10^3$), decenas de mil ($10000 = 10^4$), centenas de mil ($100000 = 10^5$), unidades de millón ($1000000 = 10^6$), y así sucesivamente. La justificación de la cantidad del número $(9874)_{10}$ con base en el sistema de posicionamiento decimal es $9874 = 9 \times 10^3 + 8 \times 10^2 + 7 \times 10^1 + 4 \times 10^0 = 9 \times 1000 + 8 \times 100 + 7 \times 10 + 4 \times 1 = 9000 + 800 + 70 + 4 = 9874$.

1.5. Máquinas de procesamiento electrónico de datos

Sistema hexadecimal: por su parte, el sistema de números hexadecimal en cuanto a su alfabeto está compuesto por el conjunto de número y letras $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$, donde las letras en su equivalencia numérica son $A = 10, B = 11, C = 12, D = 13, E = 14, F = 15$. En este sentido, la cantidad $(FFFF)_{16}$ en base hexadecimal es igual a $15 \times 16^3 + 15 \times 16^2 + 15 \times 16^1 + 15 \times 16^0 = 15 \times 4096 + 15 \times 256 + 15 \times 16 + 15 \times 1 = 61440 + 3840 + 240 + 15 = (65535)_{10}$.

Ejemplo 1.2 Conversión de bases

- a) Convertir $(1111)_2$ a base decimal

$$\begin{aligned}(1111)_2 &= 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 1 \times 8 + 1 \times 4 + 1 \times 2 + 1 \times 1 = \\ &= (15)_{10}. \text{ Lo cual corresponde al sistema de posicionamiento binario } 8 \ 4 \ 2 \ 1, \text{ con todos los bits de posicionamiento en } 1.\end{aligned}$$

- b) Convertir $(1010)_2$ a base octal

$$\begin{aligned}(1010)_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\ &= 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 0 = (10)_{10} \\ (10)_{10} &= 10/8 = 1 \text{ con residuo } 2; \text{ para lo cual se toma el cociente } 1 \text{ y el residuo } 2; \text{ entonces } (10)_{10} = (12)_8.\end{aligned}$$

- c) Realizar la operación $(1001)_2 + (1210)_3 + (1001)_8 + (100A)_{16}$, efectuando la suma en decimal:

$$\begin{aligned}(1001)_2 &= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1 = (9)_{10}. \\ (1210)_3 &= 1 \times 3^3 + 2 \times 3^2 + 1 \times 3^1 + 0 \times 3^0 \\ &= 1 \times 27 + 2 \times 9 + 1 \times 3 + 0 \times 1 = (48)_{10}. \\ (1001)_8 &= 1 \times 8^3 + 0 \times 8^2 + 0 \times 8^1 + 1 \times 8^0 \\ &= 1 \times 512 + 0 \times 64 + 0 \times 8 + 1 \times 1 = (513)_{10}. \\ (100A)_{16} &= 1 \times 16^3 + 0 \times 16^2 + 0 \times 16^1 + 10 \times 16^0 \\ &= 1 \times 4096 + 0 \times 256 + 0 \times 16 + 16 \times 1 = (4112)_{10}.\end{aligned}$$

$$\text{Luego } (9)_{10} + (48)_{10} + (513)_{10} + (4112)_{10} = (4682)_{10}.$$

- d) Realizar $(10)_{20} + (9)_{10} + (14)_8 + (100)_2$, efectuando la suma en el sistema binario:

$$(10)_{20} = 1 \times 20^1 + 0 \times 20^0 = 1 \times 20 + 0 \times 1 = (20)_{10} = (10100)_2.$$

$(9)_{10} = 9/2 = 4$ (residuo 1), $4/2 = 2$ (residuo 0), $2/2 = 1$ (residuo es 0) $= (1001)_2$, de las divisiones sucesivas se toma el último cociente y los residuos en orden inverso.

$$(14)_8 = 1 \times 8^1 + 4 \times 8^0 = 1 \times 8 + 4 \times 1 = (12)_{10} = (1100)_2.$$

Luego, al realizar la suma en binario:

Capítulo 1. Lógica de programación

$$\begin{aligned}
& 11 (\leftarrow) \text{ Valor del acarreo} \\
(10)_{20} &= (10100)_2 \\
(9)_{10} &= (01001)_2 \\
(14)_8 &= (01100)_2 \\
(100)_2 &= (00100)_2 \\
\text{Resultado} &= (101101)_2 \\
&= (20)_{10} + (9)_{10} + (12)_{10} + (4)_{10} \\
&= 1 \times 32 + 0 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 \\
&= (45)_{10}.
\end{aligned}$$

Suponga que un computador tiene x _Gb (2 gigabytes) de memoria real, una unidad de almacenamiento secundaria de y _Tb (1 terabyte) y n ($n = 4$ unidades portables de memoria, cada uno de ellas con z _Mb (1000 megabytes) de espacio de almacenamiento. Calcule la capacidad total la máquina en términos de bytes.

El espacio de cada unidad portable es igual a z _Mb, o sea, $z \times 1024 \text{ Kb} \Rightarrow z \times 1024 \times 1024$; por las n unidades de almacenamiento, el resultado es $n \times z \times 1024 \times 1024$ bytes.

El espacio de almacenamiento secundario es y _Tb = $y \times 1024 \text{ Gb} = y \times 1024 \times 1024 \text{ Mb} = y \times 1024 \times 1024 \times 1024 \text{ Kb} = y \times 1024 \times 1024 \times 1024 \times 1024$ bytes. El espacio de almacenamiento en memoria real es X _Gb = $x \times 1024 \text{ Mb} = x \times 1024 \times 1024 \text{ Kb} = x \times 1024 \text{ Mb} = x \times 1024 \times 1024 \times 1024$ bytes.

Luego el espacio total de almacenamiento del computador es igual a

$$(y \times 1024^4 + x \times 1024^3 + n \times z \times 1024^2) \text{ bytes}$$

En términos de datos concretos, los resultados de las operaciones son:

$$(1 \times 1024^4 + 2 \times 1024^3 + 4 \times 1000 \times 1024^2) \text{ bytes}$$

$$1 \times 1099511627776 + 2 \times 1073741824 + 4000 \times 1048576$$

$$1099511627776 + 2147483648 + 4194304000 = 1105853415424 \text{ bytes.}$$

1.6 Lógica de programación en Ingeniería

La lógica de programación para ingenieros es una herramienta que combina los campos de la lógica humana, la algoritmia y la Ciencia de la Computación en un

1.6. Lógica de programación en Ingeniería

sentido Informático, para consolidar las capacidades del ingeniero en identificar, formular y resolver problemas en Ingeniería.

El Consejo de Acreditación para la Ingeniería y la Tecnología *Accreditation Board for Engineering and Technology* (ABET), de los Estados Unidos de América (Wright, 1994, p. 25), la define como

La profesión en la cual el conocimiento de las ciencias matemáticas y naturales adquirido mediante el estudio, la experiencia y la práctica, se aplica con buen juicio a fin de desarrollar las formas en que se pueden utilizar, de manera económica, los materiales y las fuerzas de la naturaleza en beneficio de la humanidad.

La relación que existe entre la definición de Ingeniería dada por ABET y la lógica de programación es directa, justificada por el siguiente análisis:

La lógica humana, desde el mundo antiguo (los sumeros, los egipcios, los griegos, los romanos) hasta el mundo moderno, ha estado al servicio de la humanidad en obras asociadas a las diferentes ramas de la Ingeniería. Es así como las pirámides egipcias, el panteón como templo romano, o el Faro de Alejandría, construido por los griegos son obras de Ingeniería presentes en la edad actual, para las cuales las diferentes civilizaciones utilizaron el conocimiento matemático a fin de realizar las estructuras arquitectónicas con base en la interrelación matemático-geométrica, dentro de las cuales fueron bases de cálculo los sistemas numéricos y, consecuentemente, las reglas algorítmicas como conjunto organizado de pasos para realizar un proceso o proyecto determinado. La lógica humana para la realización de obras en cualquier área de la Ingeniería se soporta en las Matemáticas y las Ciencias Físicas y Químicas como fundamentos necesarios para entender los fenómenos de la naturaleza y, consecuentemente, en el mundo real, poder identificar y formular problemas en Ingeniería. La lógica humana tiene en la lógica algorítmica una herramienta que permite “en parte” automatizar la lógica humana; es así como cálculos de grandes cantidades en cualquier proyecto de Ingeniería son realizados más eficientemente y con mayor grado de precisión mediante el computador, para cuyo funcionamiento son necesarios los algoritmos, como paso previo a la codificación y, en últimas, a la obtención de un programa para funcionar en un computador digital. El paso de la lógica humana a la lógica algorítmica, y de esta a la lógica de programación en las fases de resolución de un problema en Ingeniería, requiere las mismas etapas de la definición de Ingeniería: el estudio, la experiencia y la práctica. Lo anterior significa que es necesario el estudio de las estructuras lógicas básicas para la construcción de algoritmos, la experiencia acumulada en la construcción de algoritmos y, finalmente, la práctica de la lógica de programación para obtener soluciones en lógicas programables aceptables y representativas de la práctica en Ingeniería en sus condiciones de utilidad al usuario.

Cualquiera de las lógicas en análisis (humana, algorítmica o de programación) no es útil si no es aplicable con buen juicio. Luego, el estudio, la experiencia y la práctica en el marco de la lógica de programación debe ser aplicado según el sistema de razonamiento lógico del ingeniero, o lo que es equivalente, con una condición de

Capítulo 1. Lógica de programación

verdad validada por el constructor del algoritmo o del programa de computador focalizado a beneficiar al usuario, tanto en el procesamiento de los datos como en las informaciones derivadas del programa de computador.

Con base en lo anterior, en el marco de una sociedad globalizada y basado en la generación de información y teniendo en cuenta el paradigma socio-técnico, se acepta el hecho que

... el uso de las máquinas computacionales ha llegado a ser amplio, y hay una necesidad entendida del alcance e impacto de lo que se llama la Revolución de la Información o la Edad de la Información Digital. (Committee for the Workshops on Computational Thinking & National Research Council, 2010, p. 7)

Luego, el trabajo del ingeniero en la sociedad del conocimiento, soportado por la sociedad de la información, requiere para el profesional como mínimo interactuar necesariamente con las máquinas generadoras de información digital (computador). Ahora bien, para construir proyectos ingenieriles soportados en computadores es necesario que el ingeniero conozca la lógica de programación; y la lógica de programación se basa necesariamente en la lógica algorítmica, porque dicha lógica se genera con base en la lógica humana. Las acciones de Ingeniería de teoría, abstracción y diseño en su relación con el computador dependen de la lógica de programación para beneficiar a los usuarios de cualquier profesión de la Ingeniería con la información. Así, las bases de datos de información, que constituyen el núcleo teórico de conocimientos en cualquier profesión humana, no hubieran sido posibles sin procesos de lógica algorítmica y de programación; porque se debe estructurar la base de datos, poblarla con datos primarios y mantenerla. La acción de abstracción en cualquier obra de Ingeniería requiere lógica; y la lógica de programación con base en la algoritmia facilita la abstracción de modelos representativos de fenómenos del mundo real; de hecho, casos como la construcción de un transbordador espacial, en el cual confluyen varias profesiones de la Ingeniería, no se realiza en su fase experimental sino a través de simulaciones de su construcción física; y los simuladores computacionales, al permitir abstraer condiciones de fenómenos reales, se soportan en la lógica de programación. El diseño de Ingeniería con base en la lógica de programación posibilita el avance de áreas tales como CAD/CAM/CAE (*Computer Aided Design/Computer Aided Manufacturing/Computer Aided Engineering*, respectivamente), debido a que

CAE significa Ingeniería Asistida por Computador y primariamente hace funcionar dos áreas: la Ingeniería de Software y la tecnología. Esta última, la manufactura la usa en conjunto con el Diseño Asistido por Computador (CAD) para hacer la ingeniería, el análisis y la optimización del diseño de producto. Diseño, análisis e Ingeniería basada en conocimiento (KBE⁵) que son aplicaciones que ayudan a los manufactureros en los conceptos de diseño a través de la simulación del comportamiento de la física de un producto en su ambiente de operación e incluyen el conocimiento y experiencia del diseñador en el proceso de manufacturación. (Mirman & McGill, 2004, p. 159)

⁵KBE: Knowledge-based engineering, o Ingeniería Basada en Conocimiento.

1.6. Lógica de programación en Ingeniería

Con base en lo anterior, es necesario en la formación de ingenieros incluir los conceptos relacionados con la construcción de algoritmos como herramientas que unidas a la lógica humana y al conocimiento experto asociado a cada profesión hacen posibles la construcción, solución y desarrollo de sistemas de información, que contruidos en software con base en la algoritmia y funcionando sobre dispositivos computacionales permiten aplicar los conocimiento de la matemáticas, la ciencia y la Ingeniería en el desarrollo de problemas neurálgicos para el desarrollo de la humanidad y para la conservación del planeta.

Por lo tanto, reconociendo la importancia de la lógica algorítmica en el desarrollo de la lógica humana y enfatizando la necesidad del desarrollo de la lógica de programación en función de la lógica algorítmica, y dada la dinámica de los desarrollos en Ingeniería en la sociedad del conocimiento, soportados por la Ciencia de la Computación, se hace necesario el entendimiento de la lógica algorítmica y la formación de ingenieros en Algoritmia y Programación, que sirven de sustento al desarrollo de proyectos informáticos en áreas tan neurálgicas para el desarrollo humano como la salud, el medio ambiente, la generación energética, en la óptica de que los proyectos informáticos desarrollados necesariamente por la dinámica de la ciencia y la técnica en Ingeniería han de ser multidisciplinarios, y una vez desarrollados se debe evaluar el impacto de las soluciones algorítmicas propuestas en el desarrollo de los proyectos en contextos nacionales, mundiales, económicos, ambientales y sociales, he ahí la importancia de la enseñanza y el aprendizaje de la lógica algorítmica para los actuales y futuros ingenieros al servicio de una sociedad información y global basada en el conocimiento humano pero con sentido humanista. Luego, la construcción de algoritmos para el desarrollo de los proyectos mencionados es el contenido central del próximo capítulo.

1.7 Conclusiones

Los temas desarrollados en este primer capítulo permiten concluir que:

- El desarrollo y avance de la ciencia y la técnica a través de la humanidad ha estado soportado por la Algoritmia, la Matemática y la Ciencia de la Computación (*Computer Science*).
- La algoritmia, en su función esencial de estudio de la construcción, el desarrollo, funcionamiento y avance de los algoritmos, está directamente correlacionada con la lógica humana; en tal sentido, un desarrollo del pensamiento del sujeto a nivel formal permite la construcción de lógicas formales; y a su vez, el desarrollo de estructuras algorítmicas permite el avance de la lógica formal de la persona. Ello implica la importancia de los algoritmos en el desarrollo presente y futuro de cualquier profesión al servicio de la sociedad.
- Los conceptos bases del funcionamiento de un algoritmo dependen de dos bases: la lógica humana como la lógica algorítmica aplicada solución a un pro-

blema representado en un conjunto de reglas organizadas hacia un fin útil o algoritmo; luego, el correcto conocimiento de las estructuras bases para la construcción de algoritmos soporta el éxito de la condición de verdad del algoritmo, unida simple al correcto ejercicio de la razón en el cumplimiento de la validez y utilidad del algoritmo, que con relación a la solución de un problema, el algoritmo debe necesariamente hacer lo que debe hacer, o sea, solucionarle el problema al usuario.

- El cumplimiento de la buena solución de un problema fundamentado en la utilización de las lógicas humana racional y algorítmica son bases para generar, de resultas, la lógica de programación, siendo un programa un algoritmo codificado en un lenguaje de programación entendible para una máquina electrónica de datos-computador.
- Las lógicas humana, algorítmica, de programación en su proceso integrado generan el software (*logiciel*) de los sistemas informáticos; software que necesita de dispositivos de hardware o el computador para su funcionamiento.
- La integración software-hardware se complementa con los procesos de comware o generación de acciones de conectividad y comunicaciones entre varios computadores que funcionan con piezas de software, lo cual conforma la red, en el marco del paradigma socio-técnico.
- La red (Internet) ha permitido la creación de grupos sociales; cada uno de ellos siendo usuarios de múltiples piezas de software y hardware interconectadas que son funcionales por la operación de lógicas de programación, y en síntesis, dichas piezas operan con bases en la lógica algorítmica.
- La red en el marco del presente y el desarrollo futuro de la Informática y la Ciencia de la Computación tiene un impacto en la política, la economía, la salud y la vida de la actual y futura sociedad del conocimiento predictivo, en el sentido de que el conocimiento es la base de las decisiones y, consecuentemente, el fundamento de los sistemas predictivos en función de informaciones almacenadas en series de tiempo representativos de fenómenos de la realidad humana.
- Por lo tanto, el recorrido realizado desde la Algoritmia a la Sociedad del Conocimiento induce el conocimiento fundamental que en la vida moderna ha de tener el sujeto de los conceptos que soportan la construcción de algoritmos; para lo cual los siguientes capítulos de este libro permiten al lector la construcción de sus propios algoritmos y el avance de su pensamiento formal y, consecuentemente, de su nivel inteligencia.

Bibliografía

Bronson, G. (2007). *C++ Para Ingeniería y Ciencias*. México, D.F: Thomson.

Bibliografía

- Burgin, M. (2009). *Theory of information: Fundamentality, diversity and unification*. SGP: World Scientific Publishing Co.
- Chabert, J.-L. & Barbin, E. (1999). *A history of algorithms: From the pebble to the microchip*. Berlin: Springer.
- Committee for the Workshops on Computational Thinking & National Research Council (2010). *Report of a workshop on the scope and nature of computational thinking*. Washington: National Academies Press.
- Coutard, O. (1999). *Governance of large technical systems*. Florence: Routledge.
- Dean, N. (2003). *Logic and language*. Gordonsville: Palgrave Macmillan.
- Deitel, H. & Deitel, P. (2004). *Cómo Programar en Java*. México: Pearson Education México, S. A. de C.V.
- Hopcroft, J. & Ullman, J. (1969). *Languages and their relation to automata*. USA: Addison-Wesley Publishing Company.
- Johansen, B. (1992). *Introducción a la teoría general de sistemas*. México, D.F: Limusa.
- Joyanes, L. (2008). *Fundamentos de programación*. Madrid: McGraw-Hill/Interamericana de España, S.A.U.
- Mirman, I. & McGill, R. (2004). CAD/CAM/CAE. En Geng, H., editor, *Manufacturing Engineering Handbook*. McGraw-Hill Professional Publishing, New York.
- Mishra, R. C. & Sandilya, A. (2009). *Reliability and quality management*. Daryaganj: New Age International.
- Myers, M. & Kaposi, A. (2004). *First systems book: Technology and management*. Singapore: Imperial College Press, 2 edition.
- Wear, L., Pinkert, J., Wear, L., & Lane, W. (1991). *COMPUTERS An introduction to Hardware and Software Design*. USA: McGraw-Hill, Inc.
- Wright, P. (1994). *Introducción a la Ingeniería*. Wilmington: Addison-Wesley Iberoamericana, S. A.

Bibliografía