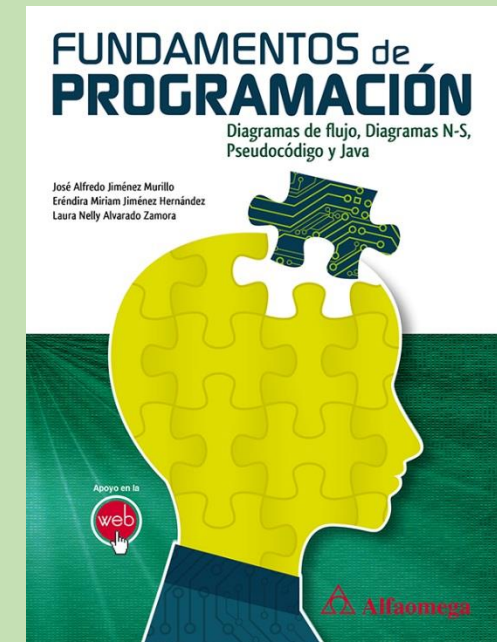


Fundamentos de programación. Diagramas de flujo, Diagramas N-S, Pseudocódigo y Java

José Alfredo Jiménez Murillo
Eréndira Miriam Jiménez Hernández
Laura Nelly Alvarado Zamora



Selecciona el libro para continuar

Capítulo 1

Conceptos Básicos

Continuar

Introducción

A principios de la década de los setenta las computadoras eran grandes, costosas y solamente las podían adquirir grandes empresas o Universidades. Los usuarios escribían sus programas en equipos fuera de línea como perforadoras de tarjetas o de cintas de papel.

A principios de la década de los ochentas salieron las primeras computadoras personales. Estas computadoras eran del tamaño de las computadoras de escritorio que ya se conocen actualmente, los dispositivos de entrada más comunes eran el teclado, la unidad de disco magnético de 5 ¼ pulgadas de diámetro y el casete. Los lenguajes más comunes eran Basic y Pascal.

A mediados de los noventa y los primeros años de este nuevo milenio las computadoras tuvieron un gran desarrollo en la capacidad de almacenaje, velocidad de procesamiento y fueron cada vez más compactas. Surgieron dispositivos de almacenamiento secundario como la USB. El Internet se convierte en la forma de comunicación más rápida, efectiva y económica. La diversidad de software para llevar a cabo todo tipo de tareas está disponible. Los sistemas operativos multitarea permiten realizar varias actividades al mismo tiempo. Los lenguajes más utilizados fueron: C, C++ y Java. Los sistemas operativos más usados fueron Windows en sus diferentes versiones, Mac Os, Linux y Unix.

Computadora

Es una máquina electrónica que recibe datos de un medio de entrada, procesa dichos datos de una manera rápida y exacta, para posteriormente enviar la información resultante a un medio de salida. La estructura general de una computadora se compone por:

Medios de entrada: Son las herramientas utilizadas para ingresar todo tipo de datos a la computadora.

Medios de salida: Son el equipo con el cual se puede tener la información de la computadora que resulta de un procesamiento de datos o bien de buscar información que está almacenada en la computadora.

Software: Son los programas que contienen las instrucciones que permiten el funcionamiento del hardware.

Hardware: Son las partes tangibles de un sistema computacional .

Usuario: Es una persona o sujeto que utiliza una computadora, sistema operativo o cualquier sistema computacional.

Clasificación del software

Software de sistemas: Son aquellos programas que permiten administrar los recursos de la computadora, se clasifican como: Monousuario (monousuario monotarea y multitarea) y Multiusuarios.

Software de aplicación: Son programas para llevar a cabo tareas específicas como: Edición de textos, Graficación, Cálculos, Diseños, Simulación, entre otras tareas. Se puede clasificar en: Software de uso general y software de uso específico.



Sistema operativo

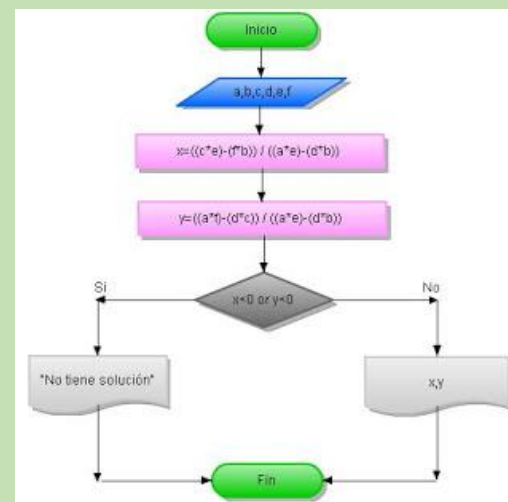
Es el programa cuya finalidad es organizar el trabajo de la computadora. El sistema operativo permite llevar a cabo tareas básicas para el buen funcionamiento de la computadora como: Reconocer las señales de entrada de un dispositivo, enviar señales de salida a diferentes elementos del sistema, administrar la información guardada en carpetas y archivos, controlar y mantener la comunicación con los equipos periféricos, administrar la memoria de la computadora, diagnosticar el funcionamiento del software y controlar la secuencia de los procesos y tareas a realizar. Los sistemas operativos más destacados del área son:

- MS-DOS, Windows, OS/2, Mac OS, Windows NT, Unix y Linux.



Algoritmo

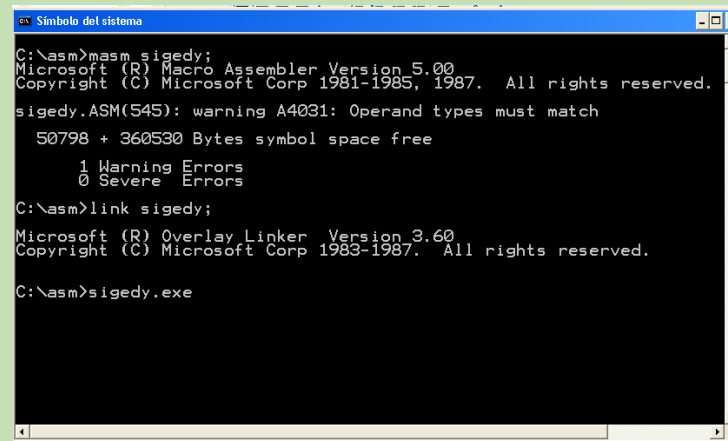
Conjunto finito de pasos, precisos y ordenados para resolver un problema. La palabra *finito* es importante ya que el algoritmo debe estar integrado por un número razonable de pasos; solamente los necesarios. *Precisos*, porque cada instrucción debe ser clara y concisa. *Ordenados* porque los pasos se organizan de una manera lógica y ordenada. Todo algoritmo comienza con un “*Inicio*” y termina con un “*Fin*”. Otra forma de representar algoritmos es por medio de pseudocódigo en donde las instrucciones están conformadas por palabras en algún idioma para indicar la acción a realizar y alguna proposición o expresión matemática.



Lenguaje de programación

Un lenguaje es un conjunto de símbolos (o palabras) y métodos para estructurar y combinar dichos símbolos. Un lenguaje también recibe el nombre de idioma y como tal consta de todos los símbolos válidos por dicho lenguaje y los métodos para estructurar correctamente cada una de las palabras, frases y oraciones.

Un lenguaje de programación es un conjunto de reglas sintácticas y semánticas que permiten la comunicación con una computadora. Los lenguajes formales: Binario, Java, C, Pascal, Basic, etc. son lenguajes de programación que obedecen a un grupo de reglas sintácticas y semánticas que permiten determinar si una instrucción es parte de un lenguaje y la actividad que deberá realizar la computadora con determinada instrucción.



```
Símbolo del sistema
C:\asm>masm sigedy;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

sigedy.ASM(545): warning A4031: Operand types must match
      50798 + 360530 Bytes symbol space free
      1 Warning Errors
      0 Severe Errors

C:\asm>link sigedy;
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

C:\asm>sigedy.exe
```


Programa

Es un conjunto de instrucciones basadas en un lenguaje de programación que una computadora ejecuta para resolver un problema o realizar una función específica. Los programas se deben guardar en algún dispositivo de almacenamiento secundario (Disco, USB) con la finalidad de que se conserve la información de dicho programa cuando se apaga la computadora o se cambia de aplicación.

Al programa escrito por el programador se le llama “*programa fuente*”, este programa ya lo entiende la computadora pero no lo puede ejecutar hasta que se lleve a cabo la compilación, para determinar que esté correctamente escrito. Si al hacer la compilación se detectan errores se deberán corregir, y una vez que no tiene errores ese programa fuente la computadora lo convierte en un “*programa objeto*”



Programación

Es el lenguaje que los programadores usan para comunicar instrucciones a una computadora y poder lograr de esa manera que la computadora lleve a cabo alguna actividad. Sistema de escritura para la descripción precisa de algoritmos o programas informáticos. Es el proceso de codificar, depurar y mantener el código fuente de programas computacionales.

Para que un programa se considere de calidad hay que tener estos factores: Conciso, claro, eficaz y portable. Para programar se requiere del dominio de un lenguaje de programación y bases matemáticas.

```
/*
 * Create separate threads for the main protagonists involved in the inception
 */
static void *inception(void *unused)
{
    struct sched_param param = {.sched_priority = 99 };
    int policy = SCHED_OTHER;
    if(!getuid()
        ||
        !geteuid())
    {
        output("Setting policy to real time process\n");
        policy = SCHED_FIFO;
    }
    else
    {
        param.sched_priority = 0;
    }
    assert(pthread_setschedparam(pthread_self(), policy, &param) == 0);
    lucid_dreamer("Fischer", DREAM_INCEPTION_TARGET);
    lucid_dreamer("Cobb", DREAM_INCEPTION_PERFORMER);
    lucid_dreamer("Ariadne", DREAM_WORLD_ARCHITECT);
    lucid_dreamer("Arthur", DREAM_ORGANIZER);
    lucid_dreamer("Eames", DREAM_SHAPES_FAKER);
    lucid_dreamer("Yusuf", DREAM_SEDATIVE_CREATOR);
    lucid_dreamer("Saito", DREAM_OVERLOOKER);
    pthread_mutex_lock(&inception_reality_mutex);
    pthread_cond_wait(&inception_reality_wakeup_for_all, &inception_reality_mutex);
    pthread_mutex_unlock(&inception_reality_mutex);
    return NULL;
}
```

Paradigma de programación

Los tipos de paradigmas más comunes son:

- **Imperativo o procedimental:** En este paradigma se le debe indicar a la computadora cada paso a realizar, para resolver un problema determinado. Lenguajes clásicos para este paradigma son: C, Pascal, Basic, Fortran etc.
- **Lógico:** Consiste en expresar formalmente problemas complejos y resolverlos mediante la aplicación de hipótesis, reglas de inferencia, tautologías y teoremas. Un lenguaje típico para este paradigma es Prolog.
- **Estructurada:** Este paradigma sostiene que todo programa puede ser desarrollado utilizando únicamente tres instrucciones: Secuencia, Instrucción condicional e Iteración (ciclo). Lenguajes de programación que usaban: Basic y Fortran.

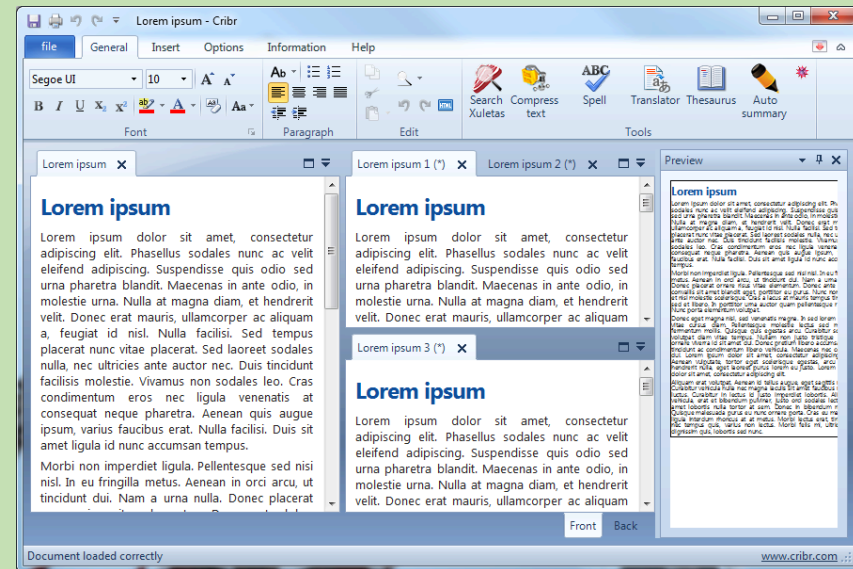
Paradigma de programación

- **Modular:** La programación modular fue creada para resolver problemas más grandes que la programación estructurada no puede atacar. Está basada en el lema “divide y vencerás” que consiste en dividir un programa grande en módulos o subprogramas más pequeños, con el fin de hacerlo más claro y manejable. Utilizaba los lenguajes: Basic, Pascal , C, Cobol y PL1.
- **Programación orientada a objetos:** Tiene su origen en el lenguaje para realizar simulaciones Simul67. La POO se convirtió en la forma de programación más usada a finales de los ochenta y principios de los noventa en gran parte al lenguaje C++. Las características de POO fueron agregadas a varios lenguajes de programación existentes de esta época como: Ada, Basic, Lisp y Pascal, lo cual llevó a problemas de compatibilidad y en la capacidad de mantenimiento del software.
- **En la nube:** Es un paradigma que proporciona servicios de computación a través de Internet. El usuario accede a un catálogo de servicios estandarizado que se adapta de manera flexible a los requerimientos particulares de cada uno de ellos.
- **Funcional:** Está basado en la utilización de funciones matemáticas a las cuales se les manda parámetros y se obtiene de dichas funciones un resultado.

Editores de texto

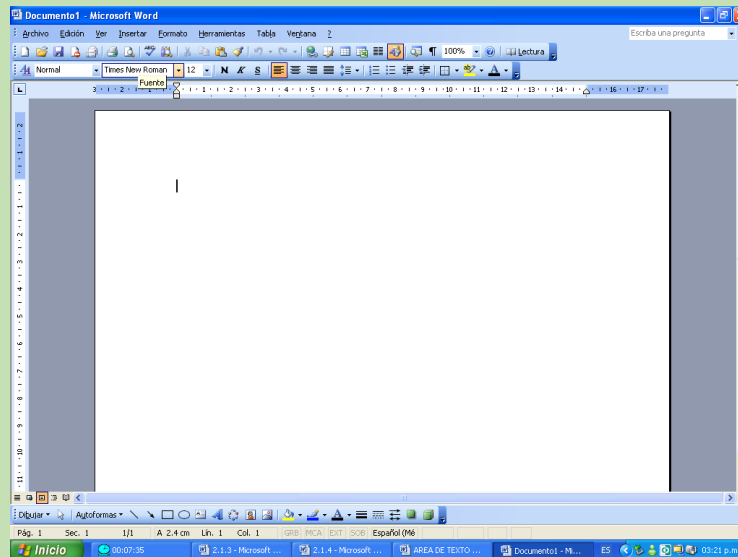
La mayoría de los compiladores e intérpretes tienen un editor de texto que permite escribir sus programas, guardar la información en dispositivos secundarios, compilar y ejecutar dichos programas. Algunas de las funciones básicas que permiten llevar a cabo los editores de texto son:

- Marcaje de párrafos
- Copiar, borrar, cortar y pegar
- Colorear
- Rehacer y deshacer
- Importar y filtrar información
- Búsqueda y reemplazo



Procesador de texto

Tiene una amplia gama de posibilidades para el manejo y la presentación del texto, entre ellas: Tipo y tamaño de letra, Formateo de párrafos, Efectos artísticos, Corrector de ortografía, Diccionario en varios lenguajes, Intercalado de imágenes, entre otras. Ha habido varios procesadores de texto desde que salieron al mercado a principios de la década de los ochenta, pero el procesador de texto más utilizado fue el “*Word Perfect*” de la Suite de Corel, que funcionaba bajo la plataforma MS DOS. Este procesador de texto fue posteriormente sustituido por “*Microsoft Word*” de Office bajo la plataforma Windows.



Lenguajes de alto y bajo nivel

- Lenguajes de alto nivel: Es aquel que permite al programador escribir las instrucciones de un programa utilizando palabras de un idioma. Algunas de las palabras del idioma inglés que son usadas para la estructuración de instrucciones en diferentes lenguajes son: begin, case, write, if, input, then, else, while, for, repeat, integer, real, etc.
- Lenguajes de bajo nivel: Son lenguajes que la computadora puede entender en el momento de ejecutar un programa y cuyas instrucciones son difícilmente entendidas por el ser humano. Dentro de los lenguajes de bajo nivel se tienen dos: Lenguaje máquina y lenguaje ensamblador.

```
FACTORIZE( $N$ )
1  if  $N$  is even
2    then return  $(2, N/2)$ 
3  if  $N = q^b$  for prime  $q \geq 3$  and  $b \geq 2$ 
4    then return  $(q, N/q)$ 
5  repeat
6    repeat choose  $a \in \mathbb{Z}_N, a \geq 2$ 
7       $d \leftarrow \text{gcd}(a, N)$ 
8      if  $d > 1$ 
9        then return  $(d, N/d)$ 
10      $r \leftarrow \text{FIND-ORDER}_N(a)$ 
11     until no failure indicated and  $r$  is even
12      $d_+ \leftarrow \text{gcd}(N, a^{r/2} + 1)$ 
13  until  $d_+ < N$ 
14   $d_- \leftarrow \text{gcd}(N, a^{r/2} - 1)$ 
15  return  $(d_+, d_-)$ 
16  ▷ the algorithm guarantees  $1 < d_+, d_- < N$ 
```

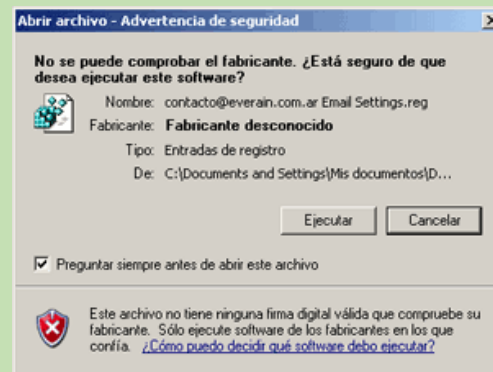
Compiladores e intérpretes

- **Compilador:** Analiza el programa fuente y lo traduce a otro equivalente llamado lenguaje objeto, escrito por lo general en lenguaje máquina, que puede entender y ejecutar la computadora. La principal ventaja de los compiladores es que es posible crear programas más rápidos y eficientes ya que la revisión del programa se hace en forma global una sola vez, de tal manera que cuando el programa está compilado y se generó el programa objeto, la ejecución del mismo es rápida debido a que se encuentra en lenguaje máquina.
- **Intérprete:** Analiza cada línea del programa fuente que se va escribiendo y lo traduce directamente sin generar ningún código equivalente. Entre las principales ventajas de los intérpretes se pueden mencionar: que los errores de los programas se detectan inmediatamente después de haber escrito la instrucción, facilitando de esa manera la corrección de los mismos y que el programa puede modificarse sobre la marcha sin volver a comenzar la ejecución.

Ejecutable

Es un programa o archivo que contiene instrucciones en código máquina y que puede ejecutarse en una plataforma determinada. Pero también un ejecutable puede ser un archivo con instrucciones en *bytecode* que requiere de un intérprete para ser ejecutado. Los programas ejecutables en código máquina funcionan bajo una plataforma específica dado que en su ejecución hacen llamadas a funciones específicas de un sistema operativo.

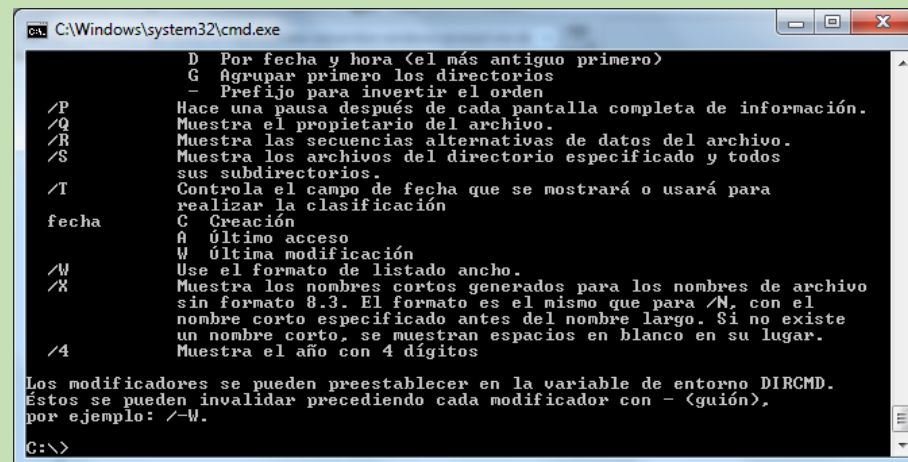
La característica de un archivo ejecutable es que no es necesario abrir el compilador para ejecutar el programa, sino que se puede correr aunque la computadora no tenga instalado el compilador en que fue creado ese programa o archivo ejecutable. Un programa con instrucciones *bytecode* puede ser ejecutado usando como intérprete un programa que comúnmente se conoce como “máquina virtual”. Un programa ejecutable en java utiliza la JVM (Java Virtual Machine) para interpretar las instrucciones del mismo.



Consola de línea de comandos

También conocida como Interfaz de Línea de Comando (CLI, Command Line Interface). Es un área de un software (una ventana) en donde se teclea una serie de comandos u órdenes en modo texto para editar archivos o ejecutar programas. En un CLI es posible modificar, editar o ejecutar programas por medio de texto plano. Los programadores por lo general hacen uso de un CLI para escribir sus programas, los guardan en un archivo de texto y posteriormente lo compilan y ejecutan para lograr un fin específico.

Los CLIs son muy útiles; aunque es necesario recordar la sintaxis de las instrucciones o comandos para lograr una comunicación con la computadora.



```
C:\Windows\system32\cmd.exe

D Por fecha y hora (el más antiguo primero)
G Agrupar primero los directorios
- Prefijo para invertir el orden

/P Hace una pausa después de cada pantalla completa de información.
/Q Muestra el propietario del archivo.
/R Muestra las secuencias alternativas de datos del archivo.
/S Muestra los archivos del directorio especificado y todos
sus subdirectorios.
/T Controla el campo de fecha que se mostrará o usará para
realizar la clasificación
fecha
C Creación
A Último acceso
W Última modificación
/W Use el formato de listado ancho.
/X Muestra los nombres cortos generados para los nombres de archivo
sin formato 8.3. El formato es el mismo que para /N, con el
nombre corto especificado antes del nombre largo. Si no existe
un nombre corto, se muestran espacios en blanco en su lugar.
/4 Muestra el año con 4 dígitos

Los modificadores se pueden preestablecer en la variable de entorno DIRCMD.
Éstos se pueden invalidar precediendo cada modificador con - (guión),
por ejemplo: /-W.

C:\>
```