Índice de Contenidos

1.1.- Concepto de ordenador y sistema operativo.

Un ordenador es una máquina creada por el hombre y por tanto no podrá realizar una tarea que no haya sido previamente determinada por él.

El ordenador no tiene inteligencia, todo lo que puede realizar son las siguientes operaciones básicas:

- Sumar y restar.
- Comparar dos valores (numéricos o alfanuméricos).
- Almacenar y recuperar información.

La combinación adecuada de estas tres operaciones permiten al ordenador realizar tareas muy complejas que aportan la solución a un determinado problema.

La potencia de cálculo de un ordenador se deriva de las características físicas que posee:

- Rapidez.
- Precisión.
- Memoria.

Estas características provienen de los componentes electrónicos:

- Velocidad de conmutación de los circuitos electrónicos.
- Rapidez de la transmisión de señales eléctricas.
- Fiabilidad de los circuitos.
- Gran capacidad de almacenamiento en el mínimo espacio posible.

El objeto del programador es, para un problema dado, diseñar una solución que pueda ser realizada por un ordenador.

Para ello necesitamos un lenguaje de programación que es una notación intermedia entre el lenguaje natural y el lenguaje del ordenador. Pero hay que tener en cuenta que todos los programas que diseñamos para resolver un problema determinado necesitan de otro programa que lo controle todo y se encargue de decirle al ordenador lo que tiene que hacer en cada momento. Estos programas se llaman “Sistemas Operativos”.

Por tanto podemos decir que el sistema operativo es el soporte base que hace que el ordenador pueda trabajar, es decir, leer datos, escribir datos, ejecutar instrucciones, manejar una impresora, etc...

Todos los demás programas se apoyan en los sistemas operativos para trabajar.

1.2.- Sistemas de procesamiento de la información.

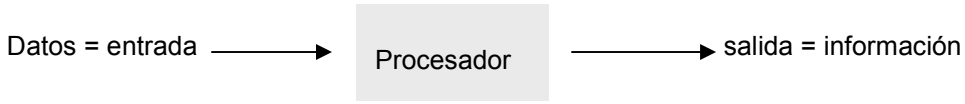
En el uso diario, datos e información son esencialmente sinónimos. Sin embargo los informáticos suelen hacer una diferencia:

- Datos se refieren a la representación de algún hecho, concepto o entidad real.
- Información implica datos procesados y organizados.

Un sistema en general se define como un conjunto de componentes conectados e interactivos, que tienen un propósito y una unidad total.

Sistema de procesamiento de información es un sistema que transforma datos brutos en información organizada y útil.

Los tres componentes básicos de un sistema de procesamiento de la información son: entrada, procesador, salida.



Es decir el procesador puede aceptar datos llamados entrada, procesarlos y producir una información denominada salida.

1.2.1.- La información y su representación.

La información es un concepto gracias al cual las personas representan acontecimientos y hechos. Con el paso del tiempo, el volumen de información ha crecido de forma desmesurada constituyendo procesos excesivamente repetitivos.

Al principio, el trabajo consistía en realizar manualmente las operaciones. En algunos sectores estas operaciones eran mínimas, pero en la mayoría de los casos las operaciones realizadas al día sobrepasan los límites.

Por todo esto podemos empezar a observar la utilidad de la informática como ciencia que se ocupa del tratamiento de los datos.

1.2.2.- Sistemas de numeración.

Un sistema de numeración es el conjunto de símbolos utilizados para representar cantidades así como las reglas que rigen dicha representación.

Un número se puede representar por un conjunto ordenado de símbolos diferentes para cada uno de los sistemas existentes.

Los sistemas de numeración actuales se definen como posiciónales, es decir que el valor que representa cada símbolo depende de su valor absoluto y de la posición que ocupa dicha cifra con respecto a la coma decimal.

La base nos permite diferenciar el sistema de numeración con el que estamos trabajando. Dicha base es el número de símbolos utilizados para representar las cantidades, sin tener en cuenta el punto decimal.

Existe un teorema que nos da la pauta para comprender el sistema de numeración, es el Teorema Fundamental de la Numeración, que dice: "El valor decimal de una cantidad expresada en otro sistema de numeración viene dado por la fórmula":

$$..... + X_4 * B^4 + X_3 * B^3 + X_2 * B^2 + X_1 * B^1 + X_0 * B^0 + X_{-1} * B^{-1} + X_{-2} * B^{-2} + X_{-3} * B^{-3} + \ldots$$

Siendo X_i cada una de las cifras que componen el número, y B la base.

Los sistemas de numeración más utilizados son:

- **Sistema decimal:** Utiliza 10 símbolos (0, 1, 2, ... , 9) para representar todos los posibles valores. El nombre usado para cada uno de estos símbolos es "dígito". Cuando combinamos varios dígitos, tenemos un número. Su valor depende no solo del valor de cada uno de ellos sino de la posición que tiene dentro del conjunto.

- **Sistema binario:** Es el sistema de numeración que utiliza internamente el hardware del ordenador. Es un sistema que utiliza 2 símbolos (0, 1), es decir su base es 2. Cada uno de estos símbolos se denomina BIT. (Es la unidad mínima de información).
  La asociación de un número determinado de bits de uso muy común en el mundo informático es el siguiente:

1 BIT	un dígito 0 , 1
1 BYTE	8 BITS
1 KILOBYTE (KB)	1.024 BYTES = 2^{10} BYTES
1 MEGABYTE (MB)	1.024 KILOBYTES = 2^{10} KILOBYTES
1 GIGABYTE (GB)	1.024 MEGABYTES = 2^{10} MEGABYTE

- **Sistema octal:** Es un sistema de numeración que utiliza 8 dígitos (0, 1, ...,7) es decir que su base de numeración es 8. La conversión a binario es la siguiente:

Binario	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

- **Sistema hexadecimal:** Es un sistema de numeración que utiliza 16 dígitos (0, 1, ... 9, A, B, C, D, E, F) es decir su base de numeración es 16.

  Un inconveniente que representa el sistema binario es la utilización de gran cantidad de bits. Para solucionar este inconveniente se utiliza el sistema Hexadecimal, que permite trabajar cómodamente con el sistema binario, puesto que cada cifra equivale a 4 dígitos.

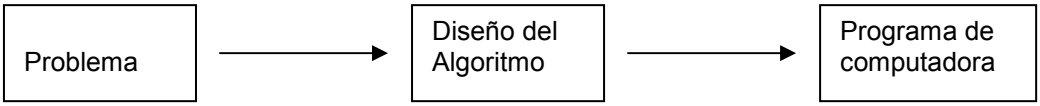
Hexadecimal	Binario	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

1.3.- Concepto de algoritmo.

El conjunto de instrucciones que especifican la secuencia de operaciones a realizar, en orden, para resolver un sistema especifico o clase de problema se denomina algoritmo.

"Algoritmo es la formula para la resolución de un problema".

Para realizar un proceso se debe suministrar al procesador un algoritmo adecuado.(Al cocinero una receta, al músico una partitura, ..etc.). Cuando el procesador es una computadora, el algoritmo ha de expresarse de una forma determinada que recibe el nombre de programa. Un programa se escribe en un lenguaje de programación y a la actividad de expresar un algoritmo en forma de programa se le denomina programación.
Cada paso en el algoritmo está expresado por medio de una instrucción en el programa. Por consiguiente, un programa consta de una secuencia de instrucciones, cada una de las cuales especifican las operaciones que debe realizar al computadora.



Los algoritmos son independientes tanto del lenguaje de programación en que se expresan como de la computadora que los ejecuta.

En la ciencia de la computación y en la programación los algoritmos son más importantes que los lenguajes de programación o las computadoras. Un lenguaje de programación es tan solo un medio para expresar un algoritmo y una computadora es solo un procesador para ejecutarlo. Tanto el lenguaje de programación como la computadora son los medios para obtener un fin: "Conseguir que el algoritmo se ejecute y se efectúe el proceso correspondiente".

Una vez que hemos diseñado el algoritmo, llegamos a la **Codificación.** Esto consiste en pasar de la descripción del algoritmo a un lenguaje de programación concreto, tal como Java. Una vez que hayamos elegido el lenguaje de programación concreto, sólo tendremos que ir siguiendo el algoritmo, y escribiendo con la sintaxis de ese lenguaje las sentencias o instrucciones que hacen lo que se indica en el algoritmo.

Por ejemplo, si el algoritmo indica:

```
nombre ⟵  "Juan"
Si (nombre = "Pepe")
        Escribir ( "El nombre del empleado es ", nombre)
Caso Contrario
        Escribir ("No lo conozco, pero se llama " , nombre)
Fin-Si
```

En Java deberemos escribir algo como lo que sigue:

```
String nombre = "Juan" ;
if (nombre.equals("Pepe")) {
        System.out.println("El nombre del empleado es "+ nombre) ;
}else {
        System.out.println("No lo conozco, pero se llama " + nombre) ;
}
```

Aún sin conocer todavía la sintaxis de Java ni la forma de describir el algoritmo en lenguaje algorítmico, vemos que existe bastante similitud entre la descripción más o menos formal, pero cercana a nuestra forma de hablar del algoritmo y el código escrito en lenguaje Java. La diferencia es que en el segundo tenemos que tener en cuenta una serie de detalles de sintaxis del lenguaje, tales como que las sentencias terminan con punto y coma, o las llaves para delimitar bloques de sentencias, o las palabras concretas que usamos para comprobar si se cumple o no una condición, o la forma de asignar un valor a una variable… En Java la sintaxis es esa y hay que escribirlo así, porque de otro modo, no funcionaría, el ordenador no lo entendería.

Por el contrario, en el algoritmo la sintaxis no es nada rígida. Podíamos haberlo expresado de otra forma, ya que el único requisito a cumplir es que quede claro para cualquier persona lo que hay que hacer en cada paso y el orden en el que se deben dar, sin ambigüedades.

Dada la importancia del algoritmo en la ciencia de la computación, un aspecto muy importante será su diseño. Las características fundamentales que debe cumplir todo algoritmo son:

- Un algoritmo debe ser **preciso** e indicar el orden de realización de cada paso.
- Un algoritmo debe estar **definido**. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
- Un algoritmo debe ser **finito**. Si se sigue un algoritmo, se debe terminar en algún momento, o sea debe tener un número finito de pasos.

1.4.- Aplicación informática.

Hoy en día la mayoría de los programas que utilizamos o elaboramos están formando parte de un conjunto de programas interrelacionados formando lo que se llama una aplicación informática.

Normalmente se trabaja con la aplicación completa aunque también se puede hacer independientemente con cada uno de los programas que la forman.

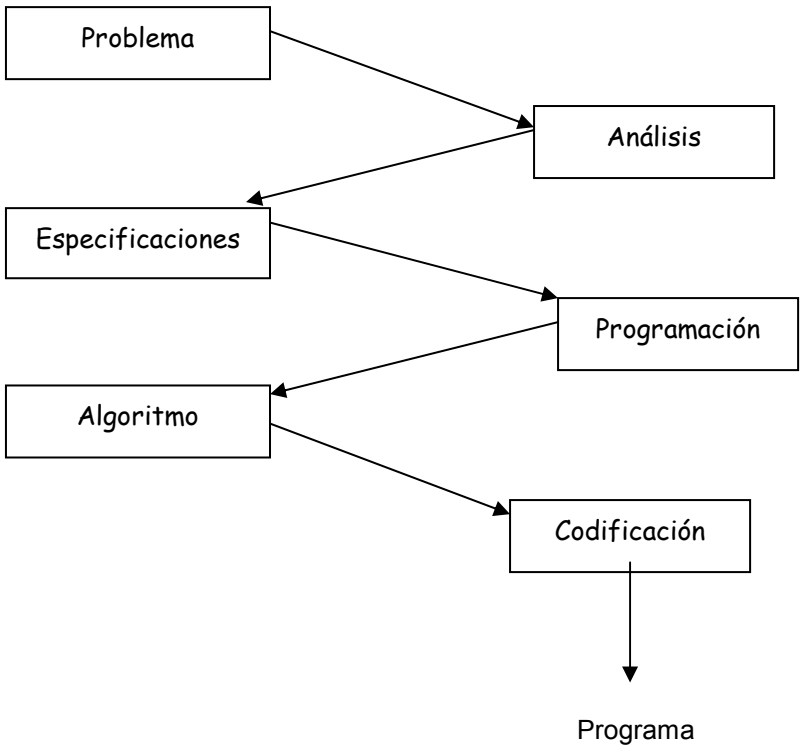
1.4.1.- Ciclo de vida de una aplicación informática.

El proceso que se sigue desde el planteamiento de un problema hasta que se tiene una solución instalada en el ordenador se denomina **ciclo de vida** de un sistema informático o **desarrollo** de una aplicación informática.

Este proceso consta de dos fases o etapas:

- **Fase de diseño o análisis:** En esta fase se comienza por identificar las necesidades y hacer un análisis de los requerimientos. Como resultado de este análisis obtenemos lo que se llama las especificaciones del problema.

  A partir de las especificaciones del problema se pasa a la fase de programación y se busca una solución al problema en forma de algoritmo. Seguidamente se transcribe el algoritmo a un lenguaje de programación en lo que se llama fase de codificación.



- **Fase de implementación o puesta a punto:** En esta fase se realiza la implementación de los programas (aplicación) en el entorno o sistema físico donde van a funcionar habitualmente y su puesta en marcha para obtener un funcionamiento normal de todo el sistema.

  Para completar el ciclo de vida hay que realizar las correcciones necesarias para subsanar errores y deficiencias del producto desarrollado.

1.5.- Características de los programas.

Un determinado problema puede tener uno o mas algoritmos que nos permitan llegar a una determinada solución. La elección del algoritmo mas adecuado se debe basar en una serie de requisitos de calidad que adquieren gran importancia a la hora de evaluar el coste de su diseño y mantenimiento.

Las características generales que debe reunir un programa son las siguientes:

- **Legibilidad:** tendrá que estar escrito de forma que sea fácil su lectura y comprensión.
- **Portabilidad:** su diseño debe permitir la codificación en distintos lenguajes de programación.
- **Modularidad:** para poder adaptarlos a una nueva situación.

- **Eficiencia:** aprovechando al máximo los recursos del ordenador.

- **Estructuración:** debe cumplir las reglas de la "programación estructurada" para facilitar la verificación y depuración del programa.

1.6.- Lenguajes de programación.

Un lenguaje de programación es una notación para escribir programas, a través de los cuales podemos comunicarnos con el hardware del ordenador y dar así las ordenes adecuadas para la realización de un determinado proceso.

Los distintos niveles de programación existentes nos permiten acceder al hardware de tal forma que, según utilicemos un nivel u otro, así tendremos que utilizar un determinado lenguaje. Una clasificación de los lenguajes desde el punto de vista de la programación de aplicaciones es:

- Lenguaje máquina.
- Lenguaje de bajo nivel (ensamblador).
- Lenguajes de alto nivel.

1.6.1.- Lenguajes máquina.

Son aquellos que están escritos en lenguajes directamente inteligibles por la máquina (ordenador), ya que sus instrucciones son cadenas binarias (0, 1) que especifican una operación y las posiciones (direcciones) de memoria implicadas en la operación se denominan "instrucciones máquina" o "código máquina".

El código máquina es el conocido código máquina.

Las instrucciones del lenguaje máquina dependen del hardware de la computadora y, por tanto, se diferenciara de un ordenador a otro.

➥ **Ventajas:** Posibilidad de cargar (transferir un programa a la memoria) sin necesidad de traducción posterior, lo que supone una velocidad de ejecución superior a cualquier otro lenguaje de programación.

➥ **Inconvenientes:**

- Dificultad y lentitud en la codificación.
- Poca fiabilidad.
- Dificultad grande a la hora de verificar y poner a punto los programas.
- Los programas solo son ejecutables en el mismo procesador.

Los inconvenientes superan ampliamente a las ventajas, lo que hace prácticamente no recomendables a los lenguajes máquina.

1.6.2.- Lenguajes de bajo nivel.

Estos lenguajes son más fáciles de utilizar que los lenguajes máquina, pero al igual que ellos dependen del ordenador, por lo que no hay un único lenguaje ensamblador.

El lenguaje de bajo nivel por excelencia es el lenguaje ensamblador, (assembler language).

Las instrucciones en lenguaje ensamblador son instrucciones conocidas como nemotécnicos (mnemonics).

Por ejemplo, nemotécnicos típicos de operaciones aritméticas son:

- En Ingles ADD, SUB, DIV
- En Español SUM, RES, DIV

Una instrucción típica de suma seria:

ADD M, N, P

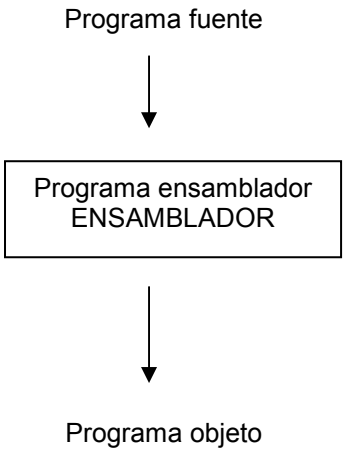
Esta instrucción podría significar: “Sumar el número contenido en la posición de memoria M al número almacenado en la posición de memoria N y situar el resultado en la posición de memoria P”.

Evidentemente es mucho mas sencillo recordar la instrucción anterior que su equivalente con código máquina:

0110 1001 1010 1011

Un programa escrito en lenguaje ensamblador no puede ser ejecutado directamente por la computadora, sino que requiere una fase de traducción al lenguaje máquina.

El programa original escrito en lenguaje ensamblador, se denomina programa fuente y el programa traducido en lenguaje máquina se conoce como programa objeto, y es directamente entendible por el ordenador.



- **Ventajas:** Las ventajas de estos lenguajes frente a los lenguajes máquina son:
  - Mayor facilidad de codificación
  - Mayor velocidad de cálculo.
- **Inconvenientes:**
  - Dependencia total de la máquina, lo que impide la transportabilidad de los programas (posibilidad de ejecutar un programa en diferentes máquinas).
  - La formación de los programadores es más compleja ya que exige no solo técnicas de programación, sino también el conocimiento del interior de la máquina.

1.6.3.- Los lenguajes de alto nivel.

Los lenguajes de alto nivel son los más utilizados por los programadores. Están diseñados para que las personas escriban y entiendan los programas de un modo mucho más fácil que los lenguajes máquina y ensambladores.

Otra razón es que un programa escrito en un lenguaje de alto nivel es independiente de la máquina, es decir que las instrucciones del programa de la computadora no dependen del diseño del hardware o de una computadora en particular. En consecuencia, los programas escritos en lenguajes de alto nivel son transportables, lo que significa que tienen la posibilidad de poder ser ejecutados con poca o ninguna modificación en diferentes tipos de computadoras.

- **Ventajas:**
  - El tiempo de formación de los programadores es relativamente corto comparado con otros lenguajes.
  - La escritura de programas se basa en reglas sintácticas similares a los lenguajes humanos.
  - Las modificaciones y puestas a punto de los programas son más fáciles.
  - Reducción del coste de los programas.
  - Transportabilidad.

- **Inconvenientes:**
  - Incremento del tiempo de puesta a punto al necesitarse diferentes traducciones del programa fuente para conseguir el programa definitivo.
  - No se aprovechan los recursos internos de la máquina que se explotan mucho mejor en lenguajes máquina y ensambladores.
  - Aumento de la ocupación de memoria.
  - El tiempo de ejecución de los programas es mucho mayor.

Al igual que sucede con los lenguajes ensambladores, los programas fuente tienen que ser traducidos por programas traductores, llamados en este caso compiladores e interpretes.
Los lenguajes de programación de alto nivel existentes en la actualidad son muy numerosos, entre ellos están BASIC, COBOL, C, C++, PASCAL, etc.

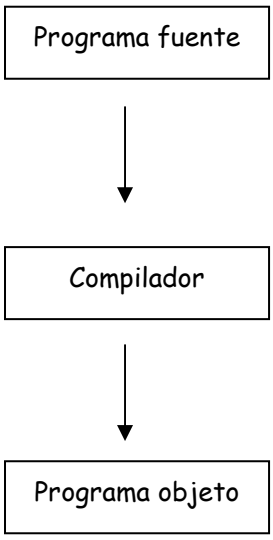
1.7.- Traductores del lenguaje.

Los traductores de lenguaje son programas que traducen a su vez los programas fuente escritos en lenguajes de alto nivel a código máquina. Los traductores se dividen en:

- Compiladores.
- Intérpretes.

1.7.1.- Compiladores.

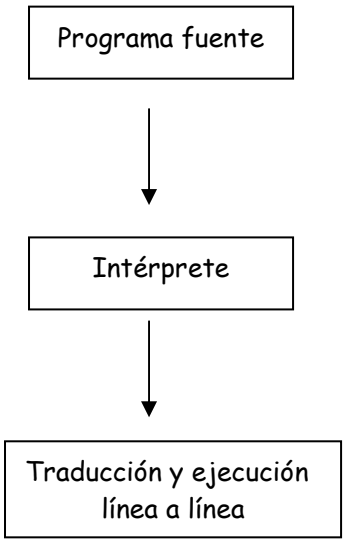
Un compilador es un programa que traduce los programas fuente escritos en lenguajes de alto nivel a lenguaje máquina.
El programa escrito en lenguajes de alto nivel se llaman programa fuente y el programa traducido programa objeto o código objeto. El compilador traduce sentencia a sentencia el programa fuente.



1.7.2.- Intérpretes.

Un intérprete es un traductor que toma un programa fuente, lo traduce y a continuación lo ejecuta.

Un lenguaje que soporte un traductor de tipo intérprete se denomina lenguaje interpretado. BASIC es el modelo por excelencia de lenguaje interpretado.



1.8.- La compilación y sus fases.

La compilación es el proceso de traducción de programas fuente a programas objeto.

El programa objeto obtenido de la compilación no ha sido traducido normalmente a código máquina sino a ensamblador.

Para conseguir el programa máquina real se debe utilizar un programa llamado montador o enlazador (linker). El proceso de montaje conduce a un programa en lenguaje máquina directamente ejecutable.

El proceso de ejecución de un programa en un lenguaje de alto nivel sigue por tanto las siguientes fases:

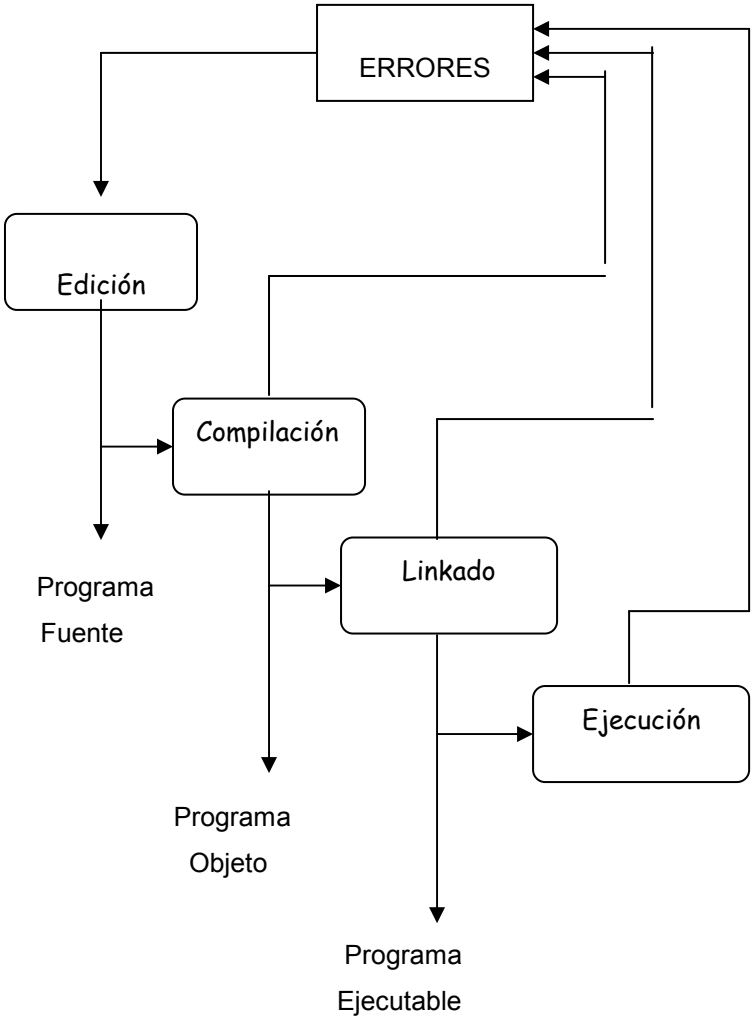
- **Edición:** Consiste en la escritura del programa (empleando un lenguaje de programación previamente seleccionado) y su posterior grabación sobre un soporte de almacenamiento permanente. La edición del programa debe realizarse mediante la utilización de un editor, que puede formar parte o no del compilador utilizado. En esta fase se obtiene el denominado programa fuente.

- **Compilación:** En esta fase se traduce el programa fuente a su equivalente código máquina, obteniendo en caso de que no se produzca ningún error el denominado programa objeto. En caso de producirse errores, el compilador los mostrará utilizando los mensajes correspondientes, que nos permitirán corregir el programa fuente y proceder de nuevo a su compilación.

- **Linkado:** Esta fase también recibe el nombre de montaje y consiste en unir o enlazar el programa objeto obtenido en la fase de compilación con determinadas rutinas internas del lenguaje y, si el método de programación es modular, se enlazan los distintos módulos para obtener así el programa ejecutable.

- **Ejecución:** Esta fase consiste en la llamada del programa ejecutable a través del sistema operativo. Inicialmente se debe comprobar el buen funcionamiento del programa mediante el uso de unos juegos de pruebas que especifican los resultados que se desean obtener en función de unos determinados datos de entrada.

Esquema del proceso de ejecución de un programa en un lenguaje de alto nivel



Los principales errores en la ejecución de un programa son:

- Datos de entrada incorrectos que producen una parada del sistema (por ejemplo, introducir un dividendo con valor cero en una operación de división).
- Bucles mal definidos que producen un funcionamiento continuo del programa (por ejemplo, un bucle sin fin o bucle infinito).
- Datos de salida incorrectos, producidos por un mal desarrollo del programa o ambigüedad en las especificaciones del usuario.

Objetivos

Introducción a la programación, tiene como objetivo presentar al alumno los conceptos básicos de la programación con el fin de que se familiarice con los términos, materiales y finalidades del módulo completo. Es una unidad con un matiz claramente conceptual que pretende ofrecer una visión global del contenido que será desarrollado durante todo el módulo. El alumno alcanzará los siguienes objetivos:

- Entender la necesidad de abordar la solución a los problemas de una forma sistemática.
- Conocer de forma genérica las fases a seguir para solucionar un problema usando ordenadores (solución informática)
- Conocer y comprender cada una de las tareas fundamentales en que se subdivide cada una de las fases de la solución de un problema.
- Introducirle en la terminología básica usada en la solución informática de los problemas.
- Distinguir entre los principales tipos de lenguajes que se usan en programación.
- Distinguir entre los principales tipos de traductores existentes, enumerando sus principales ventajas e inconvenientes.

Enlaces de interés

☞ En el siguiente enlace, aparece un artículo en el que se ejemplifica de forma más o menos general, los pasos que deben darse para solucionar un problema en general. La mayoría de ellos son los mismos que aplicamos cuando intentamos solucionar un problema informático.

Pasos para solucionar un problema
http://www.desarrolloweb.com/articulos/1597.php?manual=5

☞ En los siguientes enlaces encontrarás información algo más detallada acerca de las características de los compiladores y de los lenguajes compilados.

Características de los Compiladores
http://www.linux10.com.ar/Glosario/terminos/compilador.htm

Lenguajes Compilados
http://es.wikipedia.org/wiki/Lenguajes_compilados

☞ En este enlace aparece información exhaustiva sobre el funcionamiento de los compiladores.

Funcionamiento de los compiladores
http://www.monografias.com/trabajos11/compil/compil.shtml

☞ El siguiente enlace ofrece información adicional sobre las características de los intérpretes y los lenguajes interpretados.

Lenguajes interpretados
http://es.wikipedia.org/wiki/Lenguajes_interpretados

Glosario de términos

Algoritmo

Es una "fórmula" para resolver un problema. Un conjunto finito de acciones o secuencia de operaciones que ejecutadas en un determinado orden resuelven el problema. También puede definirse como un método para resolver un problema mediante una serie finita de pasos precisos y bien definidos. Ejemplos usuales de algoritmos son una receta de cocina, o el protocolo de actuación de un médico para atender y curar a un paciente que padece una determinada enfermedad. En el contexto de la informática, el algoritmo representa la secuencia de acciones o instrucciones que debe ejecutar el ordenador para solucionar un problema.

Código fuente

Se llama así al programa escrito en un lenguaje de alto nivel, normalmente contenido en un fichero de texto.

Código máquina

Lenguaje totalmente dependiente de las características físicas de un procesador concreto. Las instrucciones se representan en binario, con ceros y unos. Es el único lenguaje directamente ejecutable por el ordenador, por lo que cualquier programa escrito en cualquier otro lenguaje de programación debe ser traducido a código máquina para poder ser ejecutado.

Código objeto

Es el programa una vez compilado, traducido a código máquina y contenido en un fichero, que es el resultado de la compilación. Este código se puede ejecutar cada vez que se quiera sin tener que volver a traducir el programa.

Compiladores

Programas traductores capaces de traducir un programa escrito en un lenguaje de programación de alto nivel a código máquina. Se traduce todo el código fuente del programa, obteniéndose como resultado un fichero con el programa traducido a código máquina, que es ejecutable tantas veces como se quiera sin tener que volver a traducir.

Compilar

Traducir un programa escrito en lenguaje de alto nivel a código máquina usando un compilador.

Datos

En la vida común se usa como sinónimo de información, pero en informática el término dato es más restrictivo que el de información. Se define como un conjunto de símbolos que representan valores, hechos, objetos o ideas de forma adecuada para ser objeto de tratamiento.

Información

Yuxtaposición de símbolos con los que se representan valores, hechos, objetos o ideas. Normalmente en la vida común se usa como sinónimo de la palabra dato, si bien en informática se entiende como algo más amplio que los datos. Normalmente la información que maneja un programa incluye datos e instrucciones. Estas últimas indican tareas a realizar, mientras que los datos son el objeto de tratamiento.

Intérpretes

Programas traductores capaces de traducir un programa escrito en un lenguaje de programación de alto nivel a código máquina. Se analiza una a una cada una de las instrucciones del programa escrito en lenguaje de alto nivel, se traduce a código máquina y ejecutándola inmediatamente. La traducción no se guarda en ningún sitio, por lo que cada vez que queramos volver a ejecutar el programa tenemos que volver a traducirlo.

Java

Un lenguaje de programación de alto nivel, entre cuyas características fundamentales se encuentra su alta portabilidad. Los programas escritos en java siguen siendo portables incluso después de haber sido compilados. Otra característica importante es que el Kit de desarrollo básico en java (JDK o Java Development Kit) puede obtenerse gratuitamente en Internet, junto con toda una librería de clases, que resuelven problemas concretos, y que son directamente utilizables por los programadores java sin coste adicional.

Lenguaje de programación

Conjunto de símbolos y reglas sintácticas y gramaticales que permiten escribir instrucciones o sentencias válidas, comprensibles para un ordenador y aptas para ser ejecutadas en un ordenador.

Lenguaje máquina

Significa lo mismo que código máquina. Lenguaje totalmente dependiente de las características físicas de un procesador concreto. Las instrucciones se representan en binario, con ceros y unos. Es el único lenguaje directamente ejecutable por el ordenador, por lo que cualquier programa escrito en cualquier otro lenguaje de programación debe ser traducido a código máquina para poder ser ejecutado.

Lenguajes de alto nivel

Son lenguajes muy cercanos al lenguaje humano (lenguaje natural), pero evitando las ambigüedades del mismo. Son más estructurados, y con una sintaxis algo más rígida que el lenguaje natural, pero mucho más fáciles de entender que el código máquina. Son totalmente portables, es decir, cuando escribo un programa en un lenguaje de alto nivel, ese programa o código es el mismo para cualquier ordenador en el que quiera ejecutarlo. No obstante, necesitaré un traductor (compilador o intérprete) distinto para traducir al código máquina concreto de ordenadores con distintos procesadores.

Lenguajes de bajo nivel

Lenguajes totalmente dependientes de las características físicas de un microprocesador concreto. Por tanto, no son portables. Son lenguajes de bajo nivel el código máquina y el ensamblador. Ambos permiten programar usando directamente el repertorio de instrucciones del microprocesador, por lo que su uso requiere un profundo conocimiento técnico de los detalles de funcionamiento interno del mismo. Sin embargo permiten aprovechar de forma óptima los recursos del microprocesador, por lo que se usan para programas que requieren gran rapidez y eficiencia.

Portabilidad

Característica de un programa asociada a la posibilidad de ejecutarlo en distintas plataformas, es decir, en ordenadores con distintos procesadores, de distintas características y fabricantes. Puede definirse como independencia del programa con respecto a las características físicas del ordenador en el que se vaya a ejecutar.

Programador

Persona que confecciona programas de ordenador, indicándole al ordenador las tareas y sentencias que tiene que realizar y en qué orden para solucionar un problema.