

## **INTRODUCCIÓN**

Dada la dificultad de comunicación insalvable entre el computador y el programador, pronto aparecieron nuevos lenguajes de programación que hacen posible la comunicación con el microprocesador, utilizando términos y símbolos relacionados con el tipo de problema que se debe resolver, mediante el empleo de herramientas que brinda la informática. Se denominan lenguajes de alto nivel.

Estos lenguajes permiten, por un lado, escribir las operaciones que son necesarias realizar para resolver el problema de un modo parecido a como se escribiría convencionalmente (es decir, redactar adecuadamente el algoritmo de resolución del problema) y, por el otro, se encarga de traducir el algoritmo al lenguaje máquina (proceso conocido como compilación) con lo que se le confiere al programa la capacidad de corre (ser ejecutado) en el ordenador. El ordenador es en realidad tan sólo una máquina virtual, capaz de resolver todos los problemas que los usuarios seamos capaces de expresar mediante un algoritmo (programa).

En la actualidad hay muchos tipos de lenguajes de programación, cada uno de ellos con su propia gramática, su terminología especial y una sintaxis particular. Por ejemplo, existen algunos creados especialmente para aplicaciones científicas o matemáticas generales (BASIC, FORTRAN, PASCAL, etc.); otros, en cambio, se orientan al campo empresarial y al manejo de textos y ficheros, es decir, son en realidad fundamentalmente gestores de información (COBOL, PL/1, etc. ), o muy relacionados con el lenguaje máquina del ordenador (como el C y el ASSEMBLER).

Antes de dar inicio a la exposición del tema, debemos aclarar que en este trabajo no se encuentran todos los lenguajes de programación que en la actualidad existen, como por ejemplo DBASE Y CLIPPER, debido a que la información que se halló de ellos no era completa ni muy clara.

Después de presentar la introducción, y haber hecho algunas aclaraciones, daremos inicio al desarrollo del trabajo.

## **LENGUAJE DE PROGRAMACION**

Se ha descrito el programa como el conjunto de instrucciones, ordenes y símbolos reconocibles por autómata, a través de su unidad de programación, que le permite ejecutar la secuencia de control deseada. Al conjunto total de estas instrucciones, ordenes y símbolos que están disponibles se le llaman lenguaje de programación del autómata.

El programa esta formado por un conjunto de instrucciones, sentencias, bloques funcionales y grafismos que indican las operaciones a realizar.

Las instrucciones representan la tarea más elemental de un programa: leer una entrada, realizar una operación, activar una salida, etc.

La sentencia representa el mínima conjunto de instrucciones que definen una tarea completa: encontrar el valor de una función lógica en combinación de varias variable, consultar un conjunto de condiciones, etc.

El bloque funcional es el conjunto de instrucciones o sentencias que realizan una tarea o función compleja: contadores, registros de desplazamientos, transferencias de información, etc.

Todos estos elementos están relacionados entre sí mediante los símbolos o grafismos.

Los Lenguajes de programación se clasifican en:

## **LENGUAJES DE BAJO NIVEL:**

Visto a muy bajo nivel, los microprocesadores procesan exclusivamente señales electrónicas binarias. Dar una instrucción a un microprocesador supone en realidad enviar series de unos y ceros espaciadas en el tiempo de una forma determinada. Esta secuencia de señales se denomina código máquina. El código representa normalmente datos y números e instrucciones para manipularlos. Un modo más fácil de comprender el código máquina es dando a cada instrucción un mnemónico, como por ejemplo STORE o JUMP. Esta abstracción da como resultado el ensamblador, un lenguaje de muy bajo nivel que es específico de cada microprocesador.

Los lenguajes de bajo nivel permiten crear programas muy rápidos, pero que son, a menudo, difíciles de aprender. Más importante es el hecho de que los programas escritos en un bajo nivel sean altamente específicos de cada procesador. Si se lleva el programa a otra máquina se debe reescribir el programa desde el principio.

## **LENGUAJES DE ALTO NIVEL:**

Por lo general se piensa que los ordenadores son máquinas que realizan tareas de cálculos o procesamiento de texto. La descripción anterior es sólo una forma muy esquemática de ver una computadora. Hay un alto nivel de abstracción entre lo que se pide a la computadora y lo que realmente comprende. Existe también una relación compleja entre los lenguajes de alto nivel y el código máquina.

Los lenguajes de alto nivel son normalmente fáciles de aprender porque están formados por elementos de lenguajes naturales, como el inglés. En BASIC, el lenguaje de alto nivel más conocido, los comandos como IF CONTADOR=10 THEN STOP pueden utilizarse para pedir a la computadora que pare si CONTADOR es igual a diez. Por desgracia para muchas personas esta forma de trabajar es un poco frustrante, dado que a pesar de que las computadoras parecen comprender un lenguaje natural, lo hacen en realidad de una forma rígida y sistemática.

## **LENGUAJE ENSABLADOR (le):**

Este intenta desflexibilizar la representación de los diferentes campos. Esta flexibilidad se consigue no escribiendo los campos en binario y aproximando la escritura al lenguaje.

A continuación, hablaremos de algunos lenguajes de programación que existen en estos momentos:

### **FORTRAN**

FORTRAN que originalmente significa Sistema de Traducción de Fórmulas matemáticas pero se ha abreviado a la FORMula TRANslation, es el más viejo de los establecidos lenguajes de alto-nivel, fue diseñado por un grupo en IBM, durante los años 50 (1954). El idioma se hizo tan popular en los años 60, FORTRAN 66 se volvió el primer idioma en ser regularizado oficialmente en 1972.

A mediados de los años setenta se proporcionaron virtualmente cada computadora, mini o mainframe, con un sistema FORTRAN 66 normal. Era por consiguiente posible escribir programas en FORTRAN en cualquier sistema y estar bastante seguro que éstos pudieran moverse para trabajar en cualquier otro sistema bastante fácil. Esto, y el hecho que pudieran procesarse programas de FORTRAN muy eficazmente.

La definición normal de FORTRAN se puso al día en 1970 y una norma, ANSI X3.9-1978, pero recién normalmente conocido como FORTRAN 77 (desde que el proyecto final realmente se completó en 1977) y es ahora la versión del idioma en su uso extendido. FORTRAN 77 es muy difícil de representar datos estructura suintamente y la falta de cualquier medio de almacenamiento dinámico que todas las series deben tener un tamaño fijo que no puede excederse; estaba claro de una fase muy temprana, más moderno, el idioma necesitó

ser desarrollado. El trabajo empezó en los 80 en un conocido como FORTRAN 8X. El trabajo tomó 12 años en parte debido al deseo de guardar FORTRAN 77 un subconjunto estricto pero el idioma no se compuso. Idiomas como Pascal, Ada y Algol son muy fáciles de usar pero no pueden igualar la eficacia de FORTRAN.

FORTRAN 90 es un desarrollo mayor del idioma pero no obstante incluye todos los de FORTRAN 77 como un subconjunto estricto valido para FORTRAN 90.

FORTRAN 90 permite expresar los programas de maneras que se satisfacen más a un ambiente de la informática moderna y han quedado obsoletos muchos de los mecanismos que eran apropiados en FORTRAN 77. En FORTRAN 90 algunos rasgos de FORTRAN 77 han sido reemplazados por rasgos mejores, más seguros y más eficaces, muchos de éstos fueron quitados del idioma FORTRAN 95.

En los últimos años el idioma basado en FORTRAN 90 conocido como High performance Fortran (HPF) se ha desarrollado. Este idioma contiene todo de FORTRAN 90 y también incluye otras extensiones que son muy deseables. FORTRAN 95 incluirá muchos de los nuevos rasgos de HPF.

El FORTRAN tiene la ventaja de ser un lenguaje compacto que sirve muy bien para satisfacer las necesidades de los científicos y los estadísticos de los negocios.

### **LISP**

LIS Processing (Procesamiento de Listas), creado en 1960 por John McCarthy, es un lenguaje de programación de alto nivel, orientado a la generación de listas, utilizado principalmente para manipular listas de datos. Lenguaje interpretado que suele utilizarse en las investigaciones y esta considerado como el lenguaje standard en proyectos de inteligencia artificial.

### **APL**

Sus siglas significan (A Programming Language). Un Lenguaje de Programación. Este programa fue desarrollado por Kenneth Inversion en el año 1961 para resolver problemas matemáticos. Este lenguaje se caracteriza por su brevedad y por su capacidad de generación de matrices y se utiliza en el desarrollo de modelos matemáticos.

### **ALGOL**

ALGOrithmic Lenguaje (Lenguaje ANGOL algorítmico). Primer Lenguaje de Programación de proceso estructurado de alto nivel, fue creado en 1961 y orientado al uso de quienes participan en proyectos científicos y matemáticos.

Como observación podemos decir que un programa descendente del Algol es el Pascal y otro lenguaje en su línea es el Ada.

### **PL/I**

Programing Language (Lenguaje de Programación I) de alto nivel. Está diseñado para combinar las principales virtudes del Fortran, Cobol y Algol, se trata de un lenguaje de programación complejo. Compilado y estructurado, es capaz de gestionar errores y de procesar multitareas, y se emplea en entornos académicos y de investigación. Fue puesto en marcha en la década de los 60 (1964). PL/I es un lenguaje flexible y sofisticado. El elemento básico de este programa es el enunciado que termina en punto y coma. Los enunciados se combinan en procedimientos. Un procedimiento puede representar por completo a un programa pequeño o un bloque de construcción o modulo de un programa más complejo.

## **RPG**

Sus siglas en inglés significan Report Program Generator, lo que es un Programa Generador de Reportes. Este programa fue desarrollado por IBM en 1964 y diseñado en para generar informes comerciales o de negocios.

En 1960 RPG es creado para la familia 1400, pero no es hasta en 1964 que es lanzada la versión final para la IBM 360, a sido actualizado un par de veces la última actualización que se ha realizado hasta la fecha es el RPG/IV en 1995, la cual contienen el ambiente de integración de lenguaje (ILE) este programa estaba disponible con AS/400.

Report Program Operator fue introducido en 1960 como un lenguaje para duplicar rápidamente el enfoque de proceso utilizado con un equipo de tarjeta perforada. Su uso está aun limitado sobre todo para las aplicaciones de negocios que son procesadas en pequeñas computadoras. Como su nombre lo sugiere, el RPG está diseñado para generar los reportes de salida que resultan del proceso de aplicaciones de negocios.

A pesar de las capacidades de actualización de archivos, el RPG es un lenguaje de propósito limitado porque los programas objeto generados por el compilador de RPG siguen sin desviación, un ciclo de procesamiento básico.

Una ventaja del RPG es la relativa facilidad para aprenderlo y usarlo. Dado que la lógica de la programación es fija, existen menos reglas formales que en otros lenguajes.

## **BASIC**

BASIC es un lenguaje de programación de alto nivel y es la abreviación de Beginners All Purpose Symbolic Instruction Code (Código de Instrucciones Simbólicas multipropósito para principiantes) es un sistema desarrollado en la Universidad de Dartmouth en 1964 bajo la dirección de J. Kemeny y T. Kurtz. Se llevó a cabo para los GE-225. Esto significa ser un idioma muy simple para aprender y también que sería fácil de traducir. A los diseñadores desearon que los estudiantes aprendieran más adelante los idiomas más poderosos como FORTRAN o ANGOL.

En los 70 cuando Allen convenció a Gates a ayudarle a desarrollar un Idioma Básico para él. El futuro de BASIC y la PC comenzó. Allen y Gates autorizaron su BASIC a M.I.T.S. para su Altair. Esta versión tomó un total de 4K. De memoria incluía el código y los datos que se usaron para el código fuente.

Allen y Gates pusieron a babor a BASIC en otras plataformas y se mudaron a su lugar de origen en Seattle en donde ellos habían asistido a la escuela primaria juntos. En ese momento la Corporación de Microsoft comenzó su reinado en el mundo de PCC. Más tarde en los 70, BASIC se había puesto en las plataformas como Apple, Comodoro y Atari y ahora era tiempo para el DOS de Bill Gates que vino con un intérprete de BASIC. La versión distribuida con MS-DOS era GW-BASIC y corrió en cualquier máquina que podría ejecutar DOS. No había ninguna diferencia entre BASIC-A y GW-BASIC, el A proporcionado por las computadoras de IBM.

Code es un lenguaje interactivo muy popular que tiene una aceptación debido a la facilidad de su uso. Un lenguaje interactivo permite la comunicación directa entre el usuario y el sistema de computo durante la preparación y uso de los programas. Aquella persona que tenga que resolver un problema usando una computadora y con un poco o ningún conocimiento de las computadoras o de la programación puede aprender muy pronto a escribir programas en BASIC en una terminal remota o en un teclado o microcomputadora.

## **VISUAL BASIC**

Este programa fue creado por Microsoft. Es un programa moderno que da apoyo a las características y

métodos orientados a los objetos. Las metas de la programación orientada al objeto son mejorar la productividad de los programadores haciendo más fácil de rehusar y extender los programas y manejar sus complejidades. De esta forma, se reduce el costo de desarrollo y mantenimiento de los programas. En los lenguajes orientados al objeto los datos son considerados como objetos que a su vez pertenecen a alguna clase. A las operaciones que se definen sobre los objetos son llamados métodos.

### **COBOL**

Nació como el deseo de desarrollar un lenguaje de programación que fuera aceptado por cualquier marca de computadora, reunió en Estados Unidos, en mayo de 1959, una comisión (denominada CODASYL: Conference on Data Systems Languages) integrada por fabricantes de computadoras, empresas privadas y representantes del Gobierno. Dando lugar a la creación del lenguaje COBOL (COmmon Business Oriented Language– Lenguaje Simbólico de Programación Orientado a Aplicaciones Comerciales) orientado a los negocios. Llamándose ésta primera versión COBOL–60, por ser éste el año que vio la luz.

COBOL, estaba en constante evolución gracias a las sugerencias de los usuarios y expertos dando lugar a las revisiones de 1.961, 1.963 y 1.965. La primera versión standard nació en 1968, siendo revisada en 1.974, llamadas COBOL ANSI ó COBOL–74, muy extendidas todavía. En la actualidad es COBOL–85 la última versión revisada del lenguaje COBOL, estando pendiente la de 1.997.

¿Porqué se hablaba de fabricantes de computadoras y no de S.O., como en la actualidad?, porque en aquellos años no existían Sistemas Operativos abiertos, sino que cada fabricante tenía su propio S.O. y por lo tanto cada COBOL debería valer para cada computadora.

COBOL es un lenguaje compilado, es decir, existe el código fuente escrito con cualquier editor de textos y el código objeto (compilado) dispuesto para su ejecución con su correspondiente runtime. Cuando se ve un programa escrito en COBOL saltan a la vista varios aspectos:

Existen unos márgenes establecidos que facilitan su comprensión. Esta estructurado en varias partes, cada una de ella con un objetivo dentro del programa. Nos recuerda mucho al lenguaje inglés, puesto que su gramática y su vocabulario están tomados de dicho idioma.

COBOL solo dispone de comandos para realizar los cálculos más elementales, suma, resta, multiplicación y división, sino que su empleo es apropiado para el proceso de datos en aplicaciones comerciales, utilización de grandes cantidades de datos y obtención de resultados ya sea por pantalla o impresos. Con COBOL se pretendía un lenguaje universal, sin embargo, se han ido incorporando retoques y mejoras aunque son mínimas.

Con la llegada del Sistema Operativo Windows, son muchos los que intentan proveer al COBOL de esa interfaces gráfica, Objective Cobol, Visual Object Cobol de Microfocus, fijitsu Powercobol, Acucobol–GT, Vangui y Cobol–WOW de Liant (RM), etc., que están consiguiendo que éste lenguaje siga estando presente en moda visual de ofrecer los programas. Sin embargo, son muchas las empresas que siguen dependiendo del COBOL–85 tradicional para sus proyectos debido principalmente a la estructura de su sistema informática.

### **LOGO**

Derivado del griego Logos, que significa palabra. Este lenguaje de programación fue creado en 1968 por Seymour Paper, y frecuentemente se utiliza en la enseñanza de lenguaje de programación a niños. Presenta un sencillo entorno de dibujo y varias presentaciones de mayor nivel del Lenguaje Lisp. Este lenguaje es fundamentalmente educativo.

### **PILOT**

Programmend Inquiry Language Or Teaching (Consulta, lenguaje o aprendizaje de investigación programada) creado en 1969.

Este lenguaje de programación es utilizado fundamentalmente para crear aplicaciones destinadas a instrucciones asistidas por computadoras. Se caracteriza por utilizar un mínimo de sintaxis.

### **SMALLTALK**

SMALLTALK, Lenguaje de Programación orientado a objetos integrados con un entorno de desarrollo multiventana. SMALLTALK no es solo un hermoso lenguaje de computación orientado a objetos. El entorno de desarrollo merece similar valoración y ha sido copiado muchas veces, desde el Sistema Operativo de Apple MS Windows y Borland Pascal (en una memoria extensión). Muchos conceptos de SMALLTALK como los browsers y las técnicas de browsing han encontrado hoy su rumbo en muchas herramientas de desarrollo de la generación X, desarrollado por SMALLTALK poseen un factor divertido-de-usar. Los cambios se graban instantáneamente y los mismos pueden probarse rápidamente.

SMALLTALK fue desarrollado dentro del Grupo de Investigación del Aprendizaje en el Centro de Investigación de Xerox en palo Alto a comienzos de los 70. Las principales ideas de SMALLTALK se le atribuyen generalmente a Alan Kay con raíces en Simula, LISP y SketchPad. Dan Ingalls escribió el código de las primeras ventanas solapables, los pop-up menús y la clase BitBlt. Adele Goldberg y Dave Robson escribieron los manuales de referencia para SMALLTALK y fueron miembros clave del equipo de desarrollo. Un programa de licenciamiento de Xerox y Xerox Special Information Systems. Sin embargo la distribución generalizada a la comunidad de desarrollo no sucedió hasta la fundación de una nueva compañía llamada ParcPlace Systems Inc. , Dirigida por Adele Goldberg.

Un segundo SMALLTALK (SMALLTALK 4) fue desarrollado por Digitalk en los Angeles California. Este SMALLTALK estaba dirigido a cubrir la necesidad de un producto pequeño, de alta velocidad, basado en PC.

Object Technology International Inc. (OTI) desarrolló un conjunto de herramientas para proveer el control de inversiones y el manejo de configuraciones en grandes proyectos. IBM desarrolló la familia de productos VisualAge para SMALLTALK en colaboración con Object Technology (antiguamente ParcPlace-Digitalk) e IBM permanecen como los distribuidores dominantes de entornos de desarrollos en SMALLTALK. Algunos nuevos SMALLTALK se hallan en etapa de desarrollo.

### **FORTH**

Lenguaje de cuarta generación, creado en 1970, es un lenguaje estructurado e interpretado de fácil ampliación y ofrece una alta funcionalidad en un espacio reducido. Es un lenguaje de alto nivel del cual derivan en la actualidad casi todos los lenguajes empleados en los robots.

### **PASCAL**

El lenguaje de programación PASCAL de alto nivel, fue desarrollado originalmente por Niklaus Wirth, desarrolló Pascal para proporcionar rasgos que estaban faltando en otros idiomas en ese entonces. Los principales objetivos para PASCAL eran que el lenguaje sea eficiente para llevados a cabo y correrse los programas, permita bien el desarrollo de estructuras y también organizar programas. PASCAL que se nombró gracias al matemático Blaise Pascales un descendiente directo de Algol 60, que ayudó a su desarrollo. Pascal también tomó componentes de Algol 68 y Algol-W. El original idioma de PASCAL aparecido en 1971 con última revisión publicada en 1973. Fue diseñado para enseñar las técnicas de programación y otros temas a los estudiantes de la universidad y era el idioma de opción de los años 60 al 80, y se emplea generalmente en ingeniería, artes gráficas y ofimática. Al igual que el C, se trata de un Lenguaje de Programación estándar para microcomputadoras.

## **LENGUAJE C**

El lenguaje C reúne características de programación intermedia entre los lenguajes ensambladores y los lenguajes de alto nivel; con gran poderío basado en sus operaciones a nivel de bits (propias de ensambladores) y la mayoría de los elementos de la programación estructurada de los lenguajes de alto nivel, por lo que resulta ser el lenguaje preferido para el desarrollo de software de sistemas y aplicaciones profesionales de la programación de computadoras.

En 1970 Ken Thompson de los laboratorios Bell se había propuesto desarrollar un compilador para el lenguaje Fortran que corría en la primera versión del sistema operativo UNIX tomando como referencia el lenguaje BCPL; el resultado fue el lenguaje B (orientado a palabras) que resultó adecuado para la programación de software de sistemas. Este lenguaje tuvo la desventaja de producir programas relativamente lentos.

En 1971 Dennis Ritchie, con base en el lenguaje de B desarrollo NB que luego cambio su nombre por C; en un principio sirvió para mejorar el sistema UNIX por lo que se le considera su lenguaje nativo. Su diseño incluyó una sintaxis simplificada, la aritmética de direcciones de memoria (permite al programador manipular bits, bytes y direcciones de memoria) y el concepto de apuntador; además, se buscó que generase códigos que pudiese correr en cualquier máquina. Logrados los objetivos anteriores, C se convirtió en el lenguaje preferido de los programadores profesionales.

En 1980 Bjarne Stroustrup, inspirado en el lenguaje Simula67 adicionó las características de la programación orientada a objetos incluyendo la ventaja de una biblioteca de funciones orientadas a objetos) y lo denominó C con clases. Para 1983 dicha denominación cambio a la de C++. Con este nuevo enfoque surge la nueva metodología que aumenta las posibilidades de la programación bajo nuevos conceptos.

## **ADA**

Nombrado en honor de la primera persona programadora de computadoras del mundo, Augusta Ada Byron King, Condesa de Lovelace, e hija del poeta inglés Lord Byron. Ada es un idioma de la programación de alto nivel pensado para las aplicaciones en vías de desarrollo donde la exactitud, seguridad, fiabilidad y manutención son las primeras metas. Ada es un fuertemente del tipo orientado a objeto. Se piensa que trabaja bien en un ambiente del multi-lenguaje y ha standarizado los rasgos para apoyar la unión a otros idiomas. La razón de ADA proporciona una descripción de los rasgos principales del idioma y sus bibliotecas y explicaciones de las opciones hecha por los diseñadores del idioma.

Este lenguaje ADA deriva del Pascal y está patrocinado por el Departamento de Defensa de los Estados Unidos para su uso en servicios militares. Este lenguaje fue presentado por su equipo de diseño en 1979. Los críticos lo llamaron inflexible e ineficiente, en tanto que sus favorecedores lo consideraban un gran avance en la tecnología del software.

## **LENGUAJE C++**

Se pronuncia ce plus plus. Fue desarrollada por Bjarne Stroustrup en los Bell Laboratories a principios de la década de los 80. C++ introduce la programación orientada al objeto en C. Es un lenguaje extremadamente poderoso y eficiente. C++ es un super conjunto de C, para aprender C++ significa aprender todo de C, luego aprender programación orientada al objeto y el uso de éstas con C++.

## **MODULA-2**

MODular Language-2, diseñado como base secundaria del Pascal (diseñador Niklaus Wirth) en el año 1980, Lenguaje que hace hincapié en la programación modular. Es un lenguaje de alto nivel que se caracteriza por la ausencia de funciones y procedimientos estandarizados.

## **DELPHI**

Es un entorno de programación visual orientado a objetos para desarrollo rápido de aplicaciones (RAD) de propósito general, incluyendo aplicaciones cliente/servidor.

Delphi es la versión de Delphi para 32 bits (delphi 3), es decir son casi los mismos, con la única diferencia que Delphi 3 es mucho más mejorado, por ejemplo contiene un TeeChart, que sirve para los gráficos de negocio.

Delphi tiene las siguiente características:

- Rendimiento – con el mejor y más rápido compilador del mundo.
- Empresa e Internet – soluciones cliente y servicio
- Desarrollo de aplicaciones rápidas (RAD).
- Reusabilidad de componentes, un verdadero entorno orientado a objetos.
- Manejo de Base de Datos escalables.
- Arquitectura multinivel abierta y dimensionable.
- Diseminación de información de base de datos en la Web a una gran velocidad.

## **JAVA**

Es un lenguaje de programación para crear programas seguros, portátiles, orientados a objetos interactivos, para mejorar la entrega de información a través de internet, etc.

## **JAVASCRIPT**

Este lenguaje de programación originalmente fue llamado LIVESCRIPT, pero luego fue renombrado con el nombre de JAVASCRIPT, con la idea de capitalizar la fama de Java, lenguaje desarrollado por Sun Microsystems. Éste es un complemento ideal del lenguaje HTML, al permitir a la página realizar algunas tareas por si misma, sin necesidad de estar sobrecargando el servidor del cual depende; JAVASCRIPT es un lenguaje diseñado especialmente para ejecutarlo en internet.

Entre estas tareas, puede estar, por ejemplo, realizar algunos cálculos simples, formatear un texto para que sea leído por distintas personas de manera distinta, proveer de un medio de configurar la visualización de una página, realizar un prechequeo de validación en formulario antes de enviarlo, etc.

## **HTML**

El lenguaje HTML, sirve para realizar esas atractivas páginas Web. Se trata de un sistema de marcas que permite enlazar al mismo tiempo texto, sonidos y gráficos dentro del mismo documento, con otros dentro del servidor o incluso con otros servidores WWW. Es decir, es un editor para combinar textos, imágenes e incluso sonido y ahora también imágenes en movimiento. Es, en definitiva, la forma de manejar y presentar la información en la red.

Para escribir documentos de hipertexto se ha desarrollado un nuevo formato de datos o lenguaje llamado Hyper Text Markup Language (HTML). Este lenguaje permite dar indicaciones precisas al programa cliente de cómo debe presentarse el documento en pantalla o al ser impreso.

El lenguaje HTML es el usado actualmente para escribir textos Hypermediales en el web.

Tres normas fundamentales:

## 1.- HTML simplemente texto

Lo primero es saber que un documento HTML es un archivo de texto simple, luego, se puede editar con cualquier editor de textos.

## 2.- No importan los Tabs ni los saltos de línea

Los interpretes HTML no toman en cuenta las tabulaciones, los saltos de líneas ni los espacios en blanco extra. Esto tiene ventajas o desventajas. La principal ventaja es que permite obtener resultados uniformes y de buena presentación de manera bastante fácil. La principal desventaja es que un documento HTML, por lo menos se debe usar los comandos `<P>... </P>` o `<BR>` para evitar que quede todo el texto en una sola línea.

## 3.- Existen 3 caracteres especiales:

- < menor que, se usa para indicar el comienzo de un comando HTML
- >mayor que, se usa para indicar el término de un comando HTML.
- & Ampersand, se usa para escribir caracteres especiales (símbolos matemáticos, comerciales, así como el signo menor que el mayor que entre otros) en un documento.

Lo primero es conocer los comandos que debe contener todo documento HTML de más de una línea de largo:

### **EL COMANDO PÁRRAFO:**

El comando `<P>... </P>` se utiliza como un delimitador de párrafo en HTML. Inserta automáticamente un quiebre de línea al final del párrafo, y produce un espaciamiento conveniente entre los diferentes párrafos de un documento. También en forma adicional permite alinear el texto al centro, a la izquierda o a la derecha.

### **EL COMANDO QUIEBRE DE LÍNEAS:**

El comando `<BR>` permite hacer un quiebre (salto) de línea.

### **LOS COMANDOS DE ENCABEZADO:**

Los textos en HTML poseen seis niveles de encabezado. Por ejemplo el nivel 1 se usa para las divisiones mayores de texto, el nivel de encabezado 6 se usa para las divisiones más chicas de texto.

### **ESTRUCTURA DE HYPERTEXTO:**

Existen dos partes fundamentales de un documento HTML.

### **ENCABEZADO:**

Se inicia mediante el comando `<HEAD>` y se termina con `</HEAD>`. Por lo general se incluyen aquí el título del documento, mediante el comando `<TITLE>... </TITLE>`.

### **CUERPO:**

Se inicia mediante el comando `<BODY>` y se termina con el comando. Dentro del cuerpo del documento se incluyen cualquier carácter imprimible. Además es importante incluir el comando `<ADDRESS>... </ADDRESS>` al final del cuerpo pero dentro de él. Dentro del ADDRESS se escribe el nombre del autor del documento, la organización a la que pertenece, su dirección del correo electrónico y otra información que se considere relevante.

## ÍNDICE

<i>1.– INTRODUCCIÓN .....</i>	<i>2–4</i>
<i>2.– LENGUAJE DE PROGRAMACIÓN .....</i>	<i>5–6</i>
<i>3.– TIPOS DE LENGUAJES DE .....</i>	<i>7–24 PROGRAMACIÓN</i>
<i>4.– BIBLIOGRAFÍA .....</i>	<i>25</i>