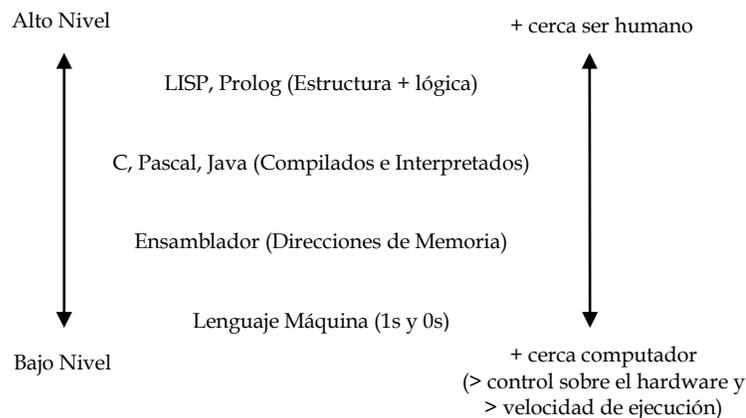




LENGUAJES DE PROGRAMACIÓN

Los lenguajes de programación están conformados por un conjunto de símbolos (léxico) y un conjunto de reglas (sintáctica y semántica).

Los computadores son máquinas que solo entienden un **lenguaje binario**, es decir, solo entienden dos estados (encendido y apagado). Conceptualmente, se puede considerar que el computador está compuesto por millones de interruptores donde cada interruptor puede tomar los valores de 1 ó 0, donde 1 significa que pasa corriente por ese interruptor y 0 significa que no pasa. Sin embargo, los seres humanos usamos un **lenguaje natural** que cuenta con muchos más símbolos y reglas, que sin embargo, no es entendido por el computador. Con base en lo anterior, existen varios tipos de lenguajes de programación:



1. Lenguajes de Bajo Nivel:

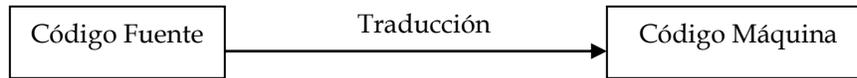
Son aquellos que están próximos al lenguaje que entiende el computador. Estos son:

- ▶ **Lenguaje Máquina:** Lenguaje de 1s y 0s que es directamente entendible por el computador. Es engorroso, fácil de cometer errores y difícil de detectarlos. Depende del hardware.
- ▶ **Lenguaje Ensamblador:** Lenguaje de etiquetas, donde se sustituye cada instrucción máquina por una etiqueta más fácil de recordar que 1s y 0s.
 - A diferencia de los lenguajes de alto nivel, no requieren de un compilador. Las instrucciones en lenguaje ensamblador son colocadas en memoria para que el microprocesador las tome directamente.
 - Presenta repetición constante de grupos de instrucciones y dificultad para encontrar errores en los programas (bugs).



2. Lenguajes de Alto Nivel:

Están próximos al lenguaje natural del computador. No son entendibles directamente por el computador, debido a que es necesario hacer una traducción.



Según la forma en que se haga esta traducción existen:

- ▶ **Lenguajes Interpretados:** (Basic, Perl, Java)
 - La traducción se realiza cada vez que se ejecuta el programa.
 - Son más lentos.
 - Para ejecutar el programa se precisa el intérprete.

- ▶ **Lenguajes Compilados:** (Pascal, C, C++)
 - La traducción se hace una sola y única vez.
 - Son más rápidos.
 - Se obtiene un ejecutable en lenguaje máquina.
 - Si se cambia el código fuente, hay que volver a compilar.

- ▶ **Ventajas de los Lenguajes de Alto Nivel:**
 - Son más flexibles y fáciles de aprender.
 - Independencia de la máquina y del SO.
 - Portabilidad a otras plataformas.

El siguiente, es un ejemplo de la misma fracción de algoritmo, pero en diferentes lenguajes, según lo descrito arriba:

<u>LENGUAJE C</u>	<u>LENGUAJE ENSAMBLADOR</u>	<u>LENGUAJE MÁQUINA</u>
suma=0;	CAR SUM,0	1111 0101 0000 0000
for (i=0; i<10; i++)	CAR i,0	1111 0111 0000 0000
suma+=i;	BUCLE: ADD SUM, i	1111 0101 0000 0000
printf ("%d",i);	ADD i, 1	1110 0111 0000 0001
	CAR AUX, i	1111 1111 0000 0101
	RES AUX, 9	0111 1111 0000 1001
	STZ SIGUE	0001 1111 1010 1111
	STI BUCLE	0010 1111 1010 0011
	SIGUE SAL i	0110 0111 0000 0000



3. INTRODUCCIÓN A C/C++

- ▶ El lenguaje C es un derivado del lenguaje B, realizado por Dennis Ritchie de los laboratorios Bell en 1972.
- ▶ Al principio fue muy conocido como lenguaje de desarrollo del Sistema Operativo Unix.
- ▶ La mayoría de los sistemas principales están escritos en C/C++.
- ▶ C es independiente del hardware.
- ▶ Es portable hacia la mayoría de las computadoras, si se hace con cuidado el diseño.

POR QUÉ C++?

- ▶ El lenguaje C++ es una evolución del lenguaje imperativo C (se llama imperativo porque implementa los programas como una serie de órdenes que la máquina tiene que cumplir).
- ▶ Fue diseñado por Bjarne Stroustrup para hacer un mejor C, con un aporte distintivo y es el de orientación a objetos (un objeto es un componente de software reutilizable que modela elementos del mundo real).
- ▶ Fue diseñado de forma que fuese compatible con el lenguaje que le precede; todo programa en C es un programa en C++, pero no a la inversa.
- ▶ Existen dos corrientes en los cursos de iniciación a la programación:
 1. Usar para la descripción de algoritmos un lenguaje parecido a cualquier lenguaje de programación, pero que esté más cercano al habla natural del estudiante. Es cuando se habla del pseudocódigo, lo que trae como ventaja las instrucciones en español y cuyo objetivo es abstraerse de un lenguaje concreto para diseñar algoritmos sin depender de las especificaciones de un lenguaje de programación.
 2. El uso directo de un lenguaje de programación, con suficiente nivel de abstracción para el diseño de algoritmos. Este es el caso de C/C++, el cual además es muy extendido en el entorno industrial y es idóneo para el aprendizaje de la programación.



4. PRIMER PROGRAMA EN C:

Para realizar en C, lo que se tiene a la izquierda en pseudocódigo, se debe hacer de la siguiente forma:

Inicio	<code>#include <stdio.h></code>
Escriba “Hola Mundo”	<code>main()</code> <code>{</code> <code> printf (“Hola mundo”);</code>
Fin	<code>}</code>

A continuación, la explicación de todas las líneas de código:

- ▶ `main()`
 - Forma parte de todo programa en C. Los paréntesis después de main indican que este es una función.
 - Los programas en C contienen una o más funciones, una de ellas debe ser el main.
 - Todos los programas en C empiezan a ejecutarse en la función main.

- ▶ `{`
 - Indica el inicio del cuerpo de cada función.

- ▶ `}`
 - Da por terminada la función.

- ▶ `printf (“Hola mundo”);`
 - Instruye a la computadora para que ejecute una acción, en este caso, escribir en pantalla el mensaje descrito por las comillas.
 - Toda esa línea se conoce como enunciado. Todo enunciado debe terminar en ;

- ▶ `#include <stdio.h>`
 - Esto se refiere a la inclusión de bibliotecas, que son unas funciones que proporcionan el acceso a elementos muy importantes de cualquier programa, como salida por pantalla, operaciones aritméticas, entrada por teclado, etc.
 - Las instrucciones de inclusión de estas bibliotecas de funciones van al principio del programa. Esto permite que el compilador lea una serie de definiciones que estarán disponibles para su uso y a partir de entonces, es como si los contenidos de esa biblioteca lo incluyera en el código.
 - Se puede digitar `#include <...>` ó `#include “...”`. Lo que va en los puntos suspensivos se llaman ficheros.



5. UTILIZANDO DEV C++

- ▶ Ingrese al DEV C++ en su computador. Luego de click en Archivo/ Nuevo/ Archivo Fuente e ingrese el siguiente código:

```
#include <stdio.h>

main()
{
    printf ("Hola mundo");
}
```

- ▶ De click en Archivo / Guardar Como y guarde el archivo con el nombre deseado.
- ▶ Luego de click en Ejecutar / Compilar (Así verifica que no haya errores).
- ▶ Y por último, de click en Ejecutar / Compilar y Ejecutar. Le debe aparecer en su pantalla, Hola mundo.

REFERENCIAS

Deitel/Deitel. Cómo programar en C/C++. Editorial Prentice Hall.

Xharfa, F., Vásquez, P., Marco, J., Molinero, X., Martin, A. Programación en C++ para Ingenieros. Editorial Thomson.