

REPRESENTACION INTERNA DE CARACTERES

Letras y símbolos. Ej.: A, E, Z, @, \$, *, etc.

- Se representan internamente como enteros sin signo.
- Dos métodos más conocidos:
 - ASCII → “American Standard Code for Information Interchange”
 - EBCDIC → “Extended Binary Code Decimal Interchange Code”

Ej.:		ASCII	EBCDIC
A	→	$(65)_{10} = (1000001)_2$	$(193)_{10} = (11000001)_2$
\$	→	$(36)_{10} = (0100100)_2$	$(91)_{10} = (01011011)_2$

LENGUAJES DE COMPUTADORAS

Programas son hechos de maneras diferentes:

- Hablamos el lenguaje máquina ó
- Hablamos a través de un traductor.

Lenguaje Máquina → “Machine Language”.

- Series de bits que identifican el código.
- Cada CPU tiene su propio lenguaje.

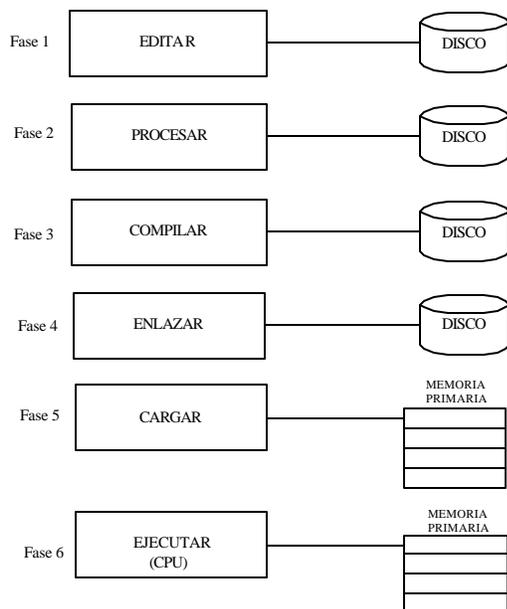
Lenguaje Ensamblador → “Assembly Language”

- ML → Complicado y poco práctico.
- Instrucciones en inglés.
- Necesitamos un interpretador (Ensamblador).

Lenguaje de Alto Nivel → “High Level Language”

- Proliferación cibernética, se comercializa la producción de programas.
- Una instrucción logra varias operaciones.
- Necesita un traductor (Compilador).

FASES DE EJECUCION EN UN ENTORNO TIPICO DE PROGRAMACION



FORMA GENERAL PARA RESOLVER PROBLEMAS MEDIANTE PROGRAMAS DE COMPUTADORA

1) Estudio preliminar del sistema:

- Definición dd problema.
- Detectar fallas.

2) Análisis de requerimientos:

- Entradas/Salidas (I/O).
- Teorías y principios aplicables.
- Lista de relaciones matemáticas.

3) Diseño:

- Algoritmo (Procedimiento detallado).
- Código (En lenguaje de computadoras).

4) Implementación:

- Compilar, Ensamblar.
- Ejecutar.

5) Pruebas:

- Verificación de resultados.
- Comparar con solución manual (Prueba de escritorio).