

# Metodología de la Programación I



I.T.I Gestión-Sistemas  
Curso 2007-08

Tema 1: Lenguajes de Programación



## ÍNDICE

1. **Concepto e historia de los lenguajes de programación.**
2. **Paradigmas de Programación.**
3. **Clasificación. Lenguajes máquina, bajo y alto nivel.**
4. **Traductores: Compiladores e Intérpretes. Proceso de generación de código ejecutable. Compilación y Enlazado.**
5. **Programación Orientada a Objetos.**



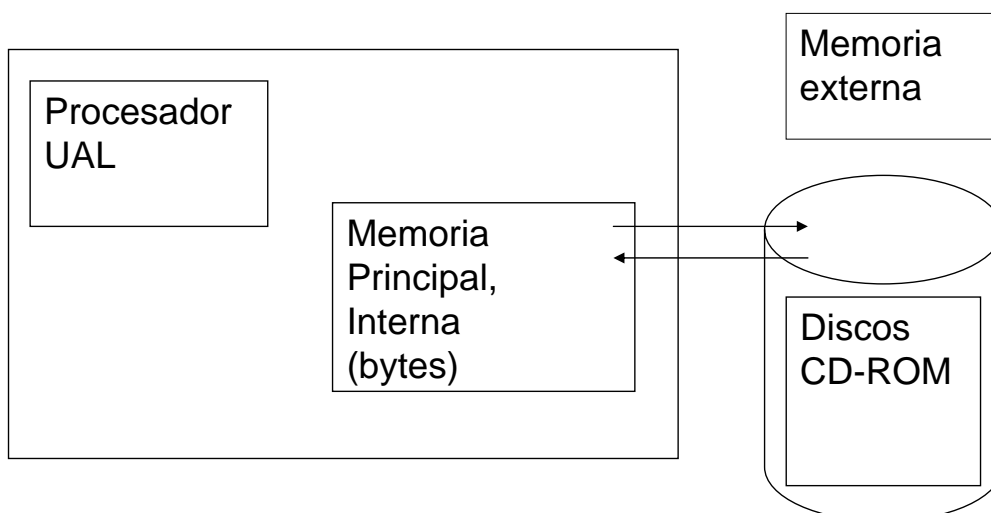
# 1. Concepto e historia de los lenguajes de Programación.

- **Informática** es la ingeniería que estudia la aplicación del tratamiento automático de la información. Los ordenadores se componen de hardware y software.
- **Hardware.** El hardware es el componente físico del ordenador. Los elementos hardware principales son cinco, la CPU (Central Processing Unit, procesador y unidad aritmético lógica), la memoria interna, la memoria externa, los dispositivos de entrada y los dispositivos de salida.

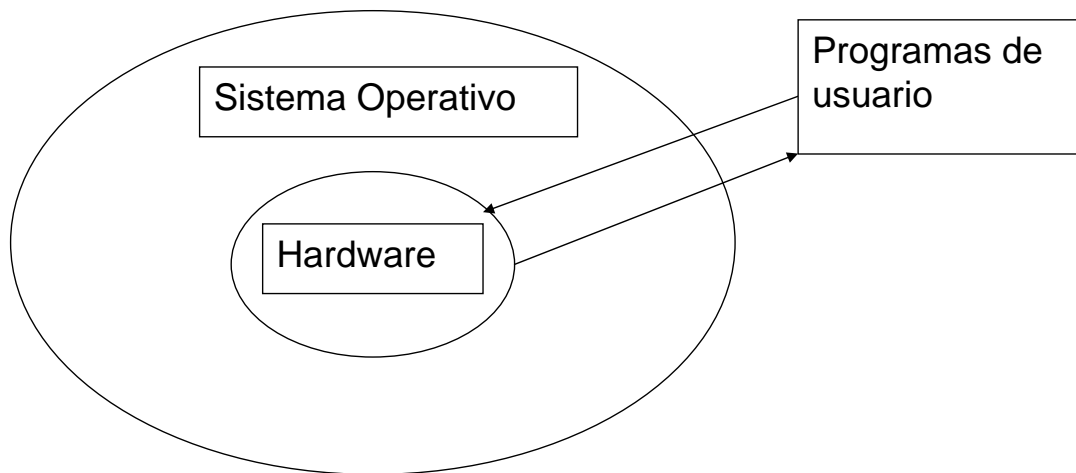
## 1. Concepto e historia de los lenguajes de Programación.

**Existen dos grupos de dispositivos externos:**

- **Dispositivos de entrada.**
- **Dispositivos de salida.**



- **Software.** Las operaciones que debe realizar el hardware vienen especificadas por una lista de instrucciones llamadas programas o software.



### Un lenguaje de programación es

- Un lenguaje que nos permite comunicarnos con el ordenador.
- Conjunto de reglas, símbolos y palabras para construir programas.

### Historia de los lenguajes de programación.

1950 Fortran, Algol, Cobol o LISP.

1960 Basic, PL/1.

1970 Pascal, C, Prolog.

1980 ADA, C++ , Modula-2 .

1990 Delphi (basado en Pascal), Visual Basic (basado en Basic), Visual C++ y C++ Builder (basados en C y C orientado a objetos).

1997 JAVA.

## 2. Paradigmas de Programación

- En el paradigma **imperativo**, las instrucciones se llevan a cabo como órdenes al estilo de una receta, calculando valores y asignándoselos a posiciones de memoria (denominadas variables).

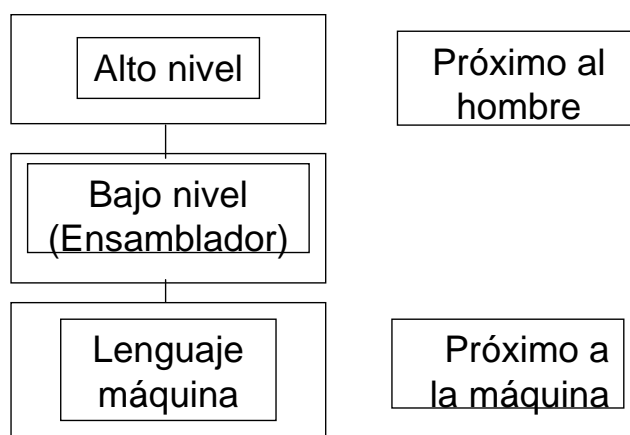
FORTRAN, C, MODULA-2, ADA o PASCAL.

- Los lenguajes **declarativos** no tienen órdenes, no describen secuencialmente el algoritmo de resolución del problema sino que se describe lo que se quiere hacer.

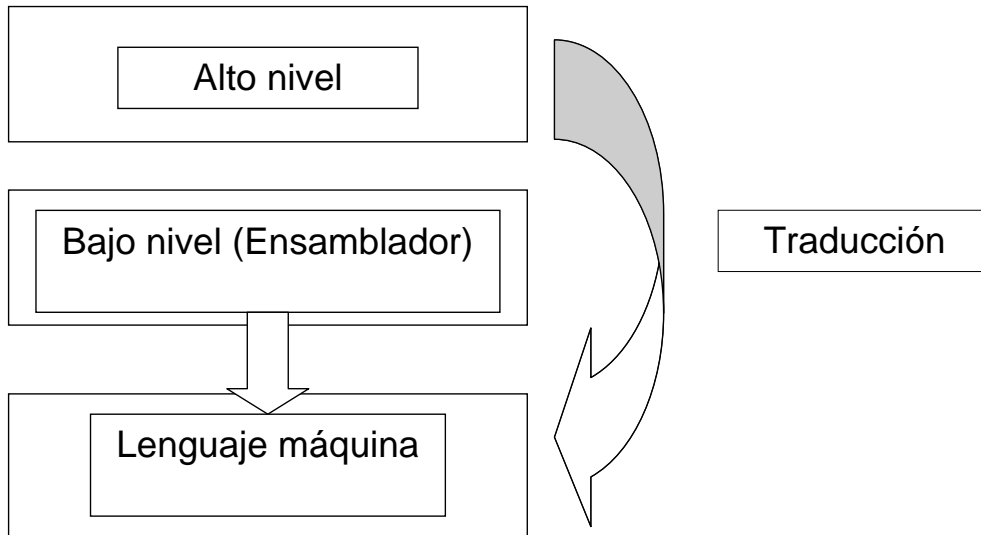
LISP, APL y PROLOG.

## 3. Clasificación. Lenguajes máquina, de bajo y alto nivel.

Según su **proximidad a la máquina**, los lenguajes de programación se clasifican en tres categorías: máquina, bajo nivel y alto nivel.

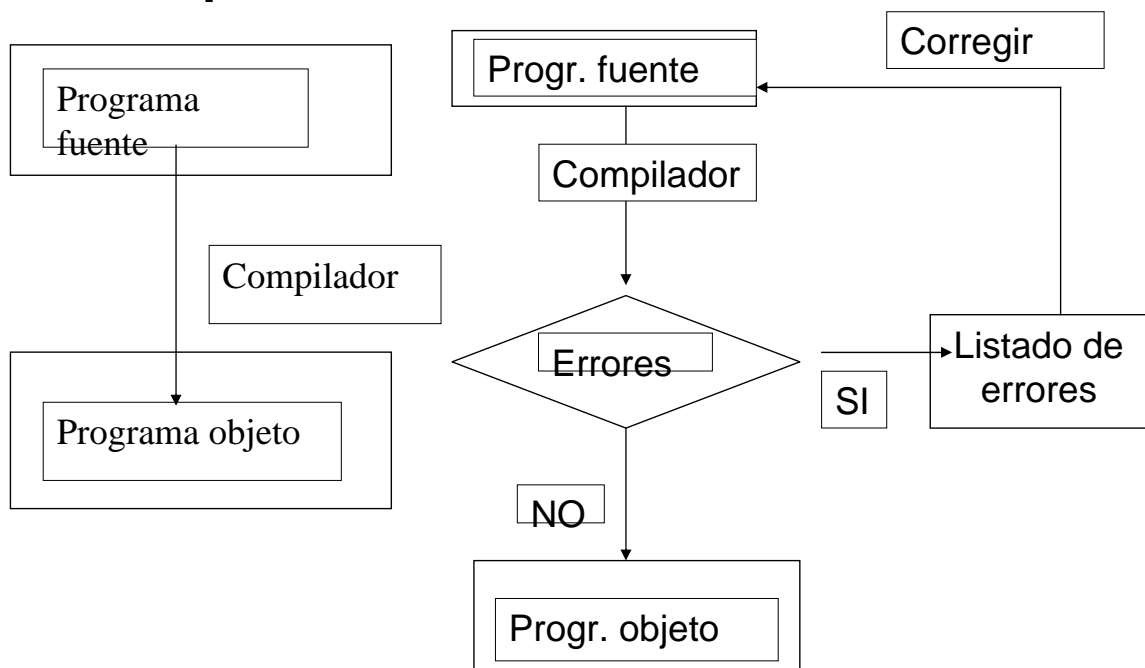


# 4. Traductores: Compiladores e intérpretes.



## 4. Traductores: Compiladores e intérpretes.

### 4.1. Compiladores.



## 4.2 Intérpretes.

Los intérpretes son programas que en lugar de traducir todo el código fuente en código máquina para su ejecución, lo que hacen es tomar, traducir y ejecutar una instrucción cada vez.

# 5.Programación Orientada a Objetos.

IDEA FUNDAMENTAL DE LA POO.

UNIR LOS DATOS Y LAS FUNCIONES QUE ESTAN RELACIONADAS CON ESOS DATOS.

## 5.Programación Orientada a Objetos.



VEHÍCULO.-

DATOS: Fabricante, modelo, color, fecha fabricación, número de plazas, número de bastidor, peso máximo...

FUNCIONES: Arrancar, frenar, repostar, acelerar, cambiar de velocidad...

VEHÍCULO sería una clase, un modelo o molde.

## CLASE Y OBJETOS

La **identidad** expresa que aunque dos objetos del mismo tipo tengan los mismos atributos , son distintos entre sí.

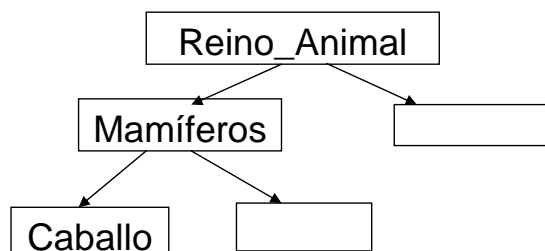
Podemos considerar tres **características** que incluye todo programa escrito con orientación a objetos:

- Encapsulación
- Polimorfismo
- Herencia

- **Encapsulación** es el mecanismo que agrupa el código y los datos que maneja este código y los mantiene protegidos frente a cualquier interferencia y mal uso.
- **Polimorfismo** es la cualidad que permite que un nombre se utilice para dos o más propósitos relacionados, pero técnicamente diferentes.

En C                                      abs() y fabs().                                      abs (-3); fabs(-3.2)  
 Sin embargo en C++ sólo abs()                                      abs (-3); abs(-3.2)

- La **herencia** es el proceso mediante el cual un objeto puede adquirir las propiedades de otro.



|                     |                               |
|---------------------|-------------------------------|
| POO.                | C++                           |
| Clase               | Clase                         |
| Objeto              | Objeto                        |
| Variables asociadas | Datos miembro                 |
| Método              | Función miembro               |
| Mensaje             | Llamada a una función miembro |