

# Unidad 0: Introducción a la programación

**Curso 2021 – 2022**



# Índice de la unidad

1. Introducción. Conceptos Previos
2. Concepto de programa
  1. Fases de creación de un programa
  2. Ejecución de un programa
3. Código fuente, código objeto y código ejecutable
4. Traductores: Compiladores, Intérpretes, Máquinas Virtuales



# I. Introducción

- # Informática
- # Sistema Informático
- # Aplicación Informática
- # Ordenador
- # Datos
- # Programa, tipos de lenguajes
- # Entorno de Desarrollo



# Informática

# Surge con el objeto de eliminar tareas repetitivas, para el **tratamiento automático de la información**.

# Proviene de dos palabras

**INFORmación+autoMÁTICA**

# Definición: Conjunto de conocimientos científicos y de técnicas que hacen posible el **tratamiento automático de la información** por medio de ordenadores



# Sistema Informático

# Conjunto de elementos necesarios para la realización y explotación de aplicaciones informáticas.

# Elementos de un sistema Informático:

- Sistema físico (Hardware)
- Sistema lógico (Software)
- Recursos Humanos:
  - Usuarios
  - Informáticos: Analista, Programador, Administrador, Planificador de trabajos, operador, formador....



# Aplicación Informática

- # Conjunto de programas, junto con la documentación asociada a los mismos, que permite la realización de un determinado tipo de trabajo (tratamiento de textos, facturación...)



# Ordenador

- ✦ Máquina compuesta de elementos físicos, principalmente de origen electrónico, capaz de:
  1. Aceptar unos datos de entrada. (ENTRADA)
  2. Realizar con ellos operaciones aritméticas y lógicas con gran velocidad y precisión. (PROCESO)
  3. Proporcionar los resultados a través de algún medio de salida. (SALIDA)

todo esto sin intervención humana y bajo el control de un programa almacenado en el propio ordenador.



# Datos

- # Conjunto de símbolos que representan una información de una forma aceptable para ser procesada por un ordenador.
- # Durante la ejecución de un programa se almacenan en **estructuras de datos** en la memoria del ordenador.





# Programa

- # Conjunto de órdenes o instrucciones que se le dan a una computadora, en un lenguaje comprensible por ella, para realizar un proceso determinado.
- # Durante la ejecución, el programa se carga en memoria desde donde irá ejecutando cada una de sus instrucciones.

# Tipos de lenguajes

- Lenguajes de programación:

Se trata de un lenguaje “artificial” diseñado para especificar las órdenes que ha de seguir un ordenador

- Lenguaje Máquina

- Es el lenguaje que entiende directamente el computador, tiene las siguientes características:
      - Las instrucciones se expresan en ALFABETO BINARIO
      - Los datos se referencian mediante la dirección de memoria donde se encuentran.
      - No aparecen nombres
      - No pueden incluirse comentarios
    - El Lenguaje máquina **depende del procesador del ordenador**, esto hace que un programa en lenguaje máquina para un determinado procesador no pueda ejecutarse en uno distinto (Baja portabilidad)

# Tipos de lenguajes

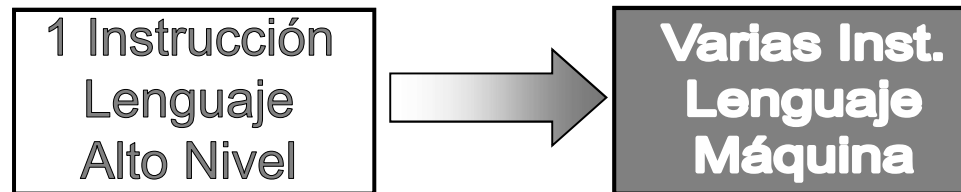
- Ensamblador

- Constituye el primer intento de sustituir el lenguaje máquina por un lenguaje más cercano al nuestro



- Alto nivel

- Son **INDEPENDIENTES DE LA ARQUITECTURA DEL COMPUTADOR**





# Entorno de Desarrollo

- # Un entorno de desarrollo integrado es una aplicación informática compuesta por un conjunto de herramientas que permiten desarrollar programas.



## 2. Concepto de programa

- # Un programa es un conjunto de instrucciones que permiten solucionar una tarea.
- # En el se reflejarán
  - # tanto las estructuras de datos que va manejar el programa → **DATOS**
  - # como las propias instrucciones del mismo.  
→ **ESTRUCTURAS DE CONTROL**




## 2.1. Fases de creación de un programa

1. Estudio del problema
2. Creación del algoritmo
3. Implementación del algoritmo en un Lenguaje de Programación concreto
4. Obtención del programa ejecutable



# Fase I: Estudio del problema

- # Consistirá en averiguar la estrategia para obtener los resultados esperados.
- # Por ejemplo: Para crear un **programa que transforme pesetas a euros**, deberemos saber la equivalencia entre las unidades y como hacerlo, es decir, 1000 pesetas son 6,01 euros porque:  
 $1000/166.386=6.01$

- 
- Estrategia: Si hacemos una regla de tres, podemos sacar la equivalencia, por ejemplo, ¿Cuántos Euros son 1000 ptas?:

1 Euro  $\rightarrow$  166,386 Ptas

x Euros  $\rightarrow$  1000 Ptas

Con lo que:

$$x = 1 * 1000 / 166,386$$

- Si en vez de 1000 fuera cualquier otra cantidad, haremos referencia a la estructura de datos que la contiene.



## Fase 2: Creación del Algoritmo

- # Un algoritmo es una descripción de pasos a dar sin asociarlo a un lenguaje de programación concreto.
  - # No solo se especifican los pasos, sino también las estructuras de datos necesarias.
  - # Por ejemplo:
    - Estructuras de datos: pesetas (nº entero), euros (nº real)
    - Leer pesetas (ENTRADA)
    - euros<- pesetas/166.386 (PROCESO)
    - Escribir pesetas, “ ptas son “, euros, “ euros” (SALIDA)
- Siendo **pesetas**, y **euros** las estructuras de datos donde se almacenan los mismos



## Fase 3: Implementación del algoritmo en un Lenguaje

- # Para poder realizar esta fase, deberemos de conocer las reglas del lenguaje de programación a utilizar.

# Fase 3: Implementación del algoritmo en un Lenguaje

## En Java:

```
import java.util.Scanner;
public class Ejemplo {
    public static void main(String[] args) {
        int pesetas;
        float euros;
        Scanner tcl = new
        Scanner(System.in);
        System.out.println("Dame
        cantidad en pesetas.");
        pesetas=tcl.nextInt();
        euros=pesetas/166.386f;
        System.out.printf("%d ptas=%f
        euros",pesetas, euros);
    }
}
```

## En Visual Basic:

Module Module1

Sub Main()

Dim ptas As Integer  
Dim euros As Double

Console.WriteLine("Dame pesetas")  
ptas = CInt(Console.ReadLine())

euros = ptas / 166.386

Console.WriteLine("{0} ptas={1} euros",  
ptas, euros)

Console.ReadLine()

End Sub

End Module

Definición de datos

Leer pesetas

Euros <- ptas / 166.386

Escribir euros



## Fase 4: Obtención del programa ejecutable.

- # En esta fase utilizamos un **programa traductor**
- # El programa traductor se encarga de transformar el programa (llamado programa fuente) escrito en la fase anterior a lenguaje máquina (llamado programa ejecutable).



## 2.2. Ejecución de un programa

- ✦ Antes de ver cual es la secuencia de ejecución de las distintas instrucciones de un programa escrito en lenguaje máquina, veamos brevemente como es un ordenador internamente.
- ✦ Hay que tener en cuenta que una vez traducido:
  - Una instrucción en lenguaje de alto nivel equivale a varias instrucciones escritas en lenguaje máquina.



## 2.2.1. Unidades funcionales del ordenador

# Memoria

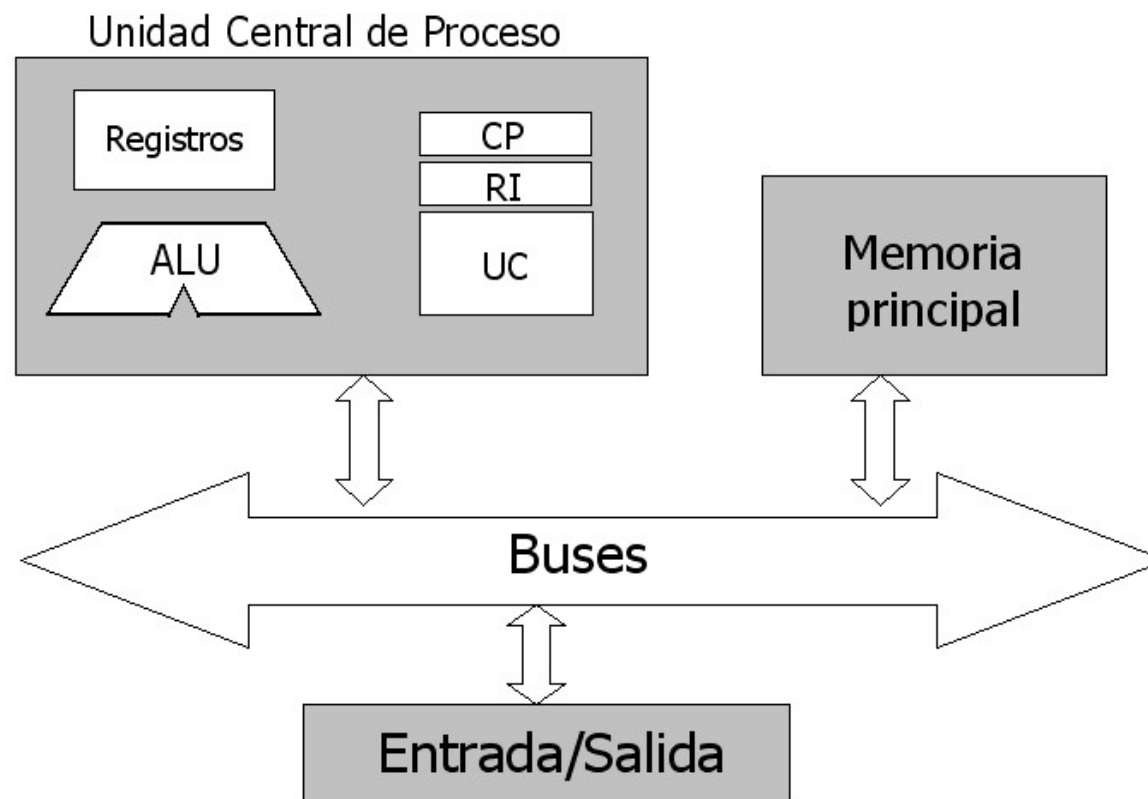
# UCP:

- Unidad de Control
- Unidad Aritmético Lógica
- Registros, entre otros:
  - Registro de instrucción
  - Contador de instrucciones

# Buses

# Dispositivos de entrada/salida

# Unidades funcionales





# Memoria

- # Su cometido es el almacenamiento de la información.
- # Durante la ejecución de un programa en ella se almacena:
  - Datos
  - Instrucciones
- # Está compuesta por un conjunto de celdas numeradas. Se accede a cada una de ellas a través de su Dirección de memoria.





# Unidad Central de Proceso

- # Es el Procesador. También conocida como la UCP o CPU, se encarga de ejecutar las distintas instrucciones de un programa.
- # Esta compuesta por:
  - Unidad de Control
  - Unidad Aritmético-Lógica
  - Registros



# Unidad de Control

- # Es el que se encarga de distribuir las órdenes para que se ejecuten las instrucciones.
- # Se encarga de gobernar el funcionamiento del ordenador y la transferencia de información entre los distintos elementos que componen el ordenador, incluyendo los dispositivos de entrada/salida.



# Unidad Aritmético-Lógica

- # Se encarga de realizar las operaciones
  - Aritméticas: sumas, restas, multiplicaciones...
  - Lógicas: comparaciones,...
- # Una vez la Unidad de Control ha dado las órdenes pertinentes para situar los datos en sus registros de entrada, le indica la operación a realizar.



# Registros del procesador

# Entre otros, los más importantes son:

- **Registro de instrucción (RI):** almacena la instrucción que se está ejecutando en ese momento.
- **Contador de Programa (CP):** contiene la dirección de memoria de la siguiente instrucción a ejecutar.



# Dispositivos de entrada / salida

## # Existen dos tipos:

- De comunicación: permiten el diálogo con el exterior.
- De almacenamiento auxiliar: son memorias auxiliares con capacidad de almacenamiento mucho mayor que la memoria principal.




# Buses

- # Se encargan de conectar las unidades funcionales del ordenador, para intercambiar información entre ellas.
- # Existen Buses de direcciones y buses de datos.



## 2.2.2 Ejecución de una instrucción


1. Lectura de la instrucción en memoria e incremento del CP.
2. Decodificación de la instrucción y búsqueda de operandos.
3. Ejecución de la instrucción (si es de salto se modifica el contenido del CP).
4. Vuelta al primer paso.



### 3. Código fuente, objeto, y ejecutable.

- # Para que un programa sea ejecutable, deberá de ser traducido al lenguaje entendible por el ordenador (el lenguaje máquina).
- # Esa traducción es realizada por unos programas especiales denominados traductores, cuyo cometido es transformar un programa escrito en un lenguaje de alto nivel, en instrucciones escritas en lenguaje máquina.



- 
- # Los programas de aplicación se apoyan no sólo en la CPU, sino que necesitan de la participación del sistema operativo para realizar muchas de sus labores.
  - # A veces, hay que incluir en esta lista otros programas como servidores de aplicaciones, sistemas gestores de bases de datos, librerías de código, etc.
  - # A este conjunto de requisitos necesarios para la ejecución de un programa se le suele llamar a menudo **plataforma**.



# Código fuente

- # También llamado programa fuente o fichero fuente.
- # Se trata de un fichero de texto que contiene el programa escrito en un lenguaje de alto nivel.
- # Dicho fichero se puede editar en un editor cualquiera, o en el que viene incluido en el entorno de desarrollo utilizado.
- # Como cualquier lenguaje, cada lenguaje de programación tiene sus propias reglas léxicas, sintácticas y semánticas válidas.



# Código objeto

- # También llamado fichero objeto.
- # El código objeto es un programa traducido a lenguaje máquina, pero de forma incompleta, ya que falta añadirle el código correspondiente a las macroinstrucciones utilizadas de los módulos de librería.
- # En los lenguajes de programación estructurados, el linkador lo tomaba como entrada para obtener el ejecutable.



# Código ejecutable

- # Es el lenguaje de la CPU.
- # Son secuencias binarias que la CPU ejecuta como instrucciones sencillas.
- # Una instrucción en un lenguaje de alto nivel puede dar lugar a varias instrucciones en lenguaje máquina-
- # Es el que se sitúa en la memoria del ordenador cuando ejecutamos un programa.
- # Cada CPU tiene su propio lenguaje máquina, aunque algunos de ellos son compatibles entre si.



## 4.Traductores

- # Son programas especiales que traducen programas escritos en un lenguaje de programación de alto nivel al lenguaje máquina.



# Tipos de traductores

# En función del lenguaje de programación, existe una forma diferente de traducirlo a lenguaje máquina:

- Compiladores
- Intérpretes
- Máquinas Virtuales



# Compiladores

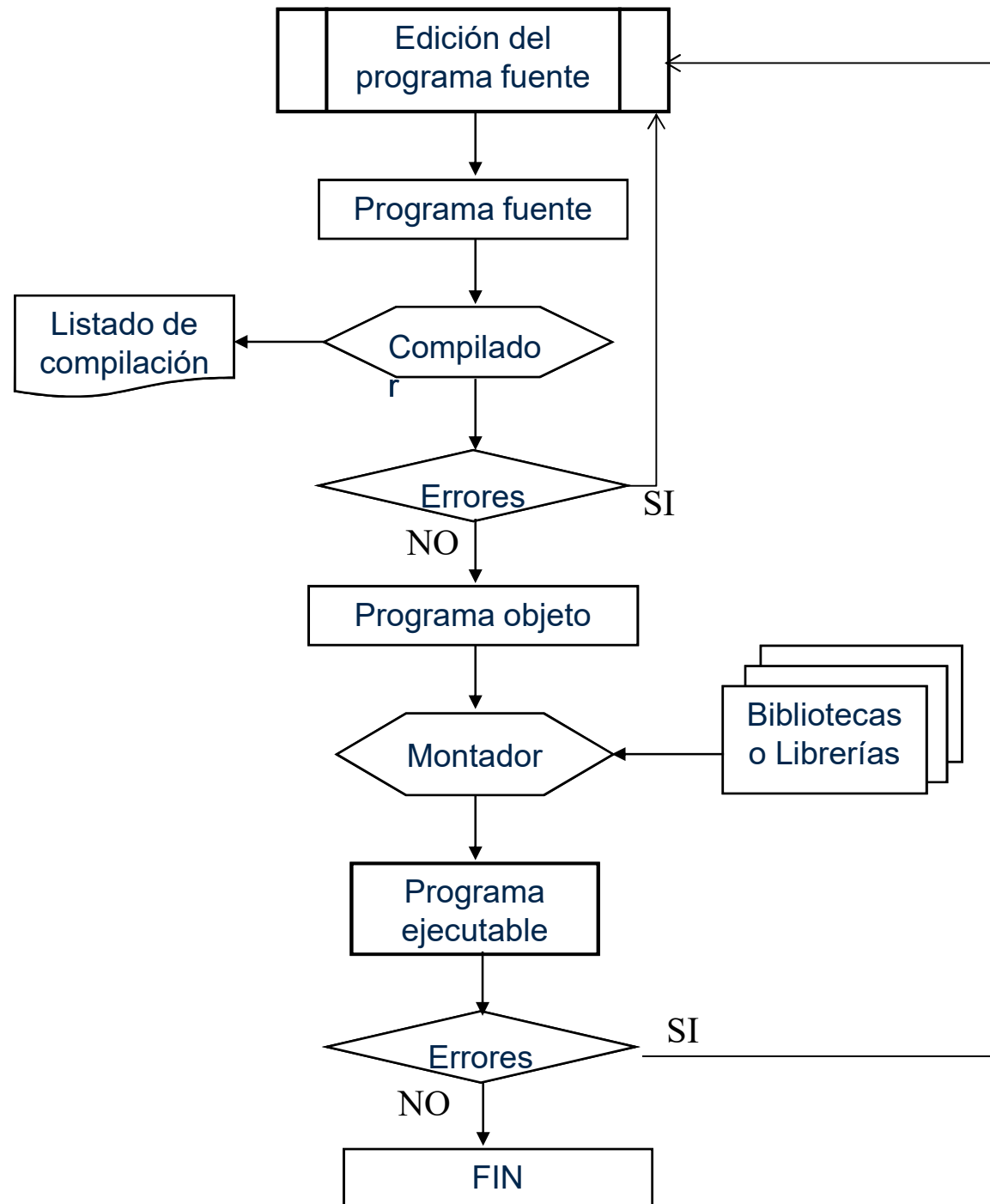
- # Traducen completamente el código fuente a código objeto.
- # El código objeto se obtiene después de corregir los errores léxicos, sintácticos y semánticos que se existen en el código fuente.
- # Al código objeto, en el proceso de linkado se le añaden el código de las macroinstrucciones de librería utilizadas, obteniendo el ejecutable.



# Compilación

- # Como consecuencia del proceso de compilación + el proceso de linkado se obtiene un fichero ejecutable, listo para utilizar.

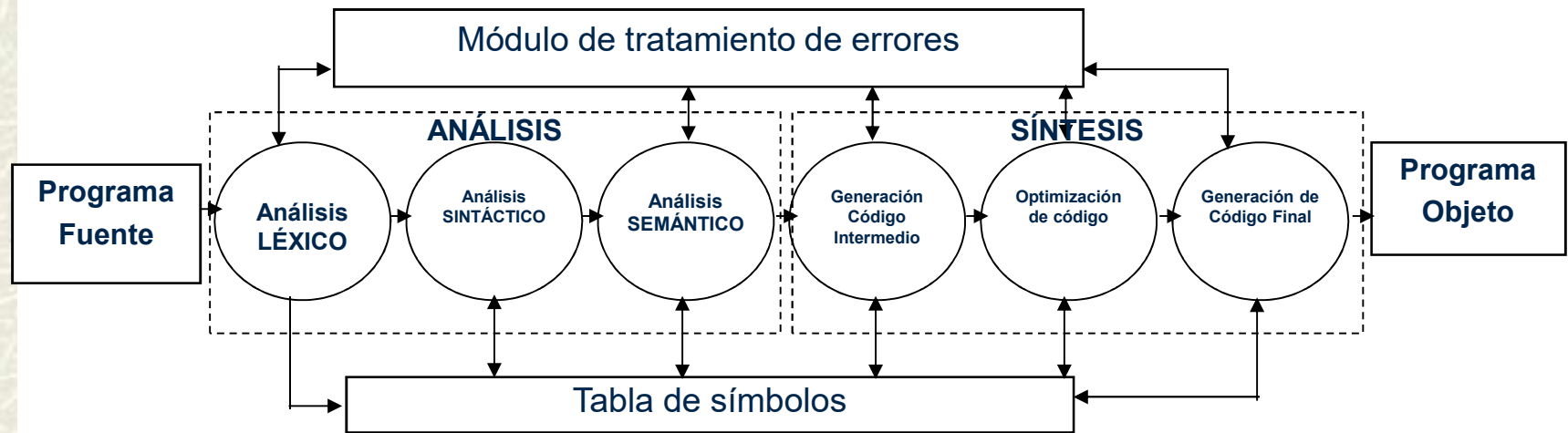






# El proceso de compilación

- # Es un proceso complejo que consume a veces un tiempo muy superior a la propia ejecución del programa.
- # Costa de dos etapas fundamentales:
  - Análisis del programa fuente
  - Síntesis del programa objeto.



# Tipos de errores

- # **Lexicográficos:** Tokens no reconocibles
- # **Sintácticos:** Al no reconocer una tira de tokens como un formato válido de sentencia
- # **Semánticos:** Sentencia sintácticamente correcta pero carente de sentido dentro del contexto.
- # **Lógicos:** Debidos a la utilización de un algoritmo o expresión incorrecta. No se obtienen los resultados previstos.
- # **De Ejecución:** Relacionados con desbordamientos, operaciones matemáticamente irresolubles: Dividir entre 0, raíz cuadrada de un  $n^\circ$  negativo,....bucles infinitos.
- # Errores que no perjudican al resto del proceso de compilación e incluso pueden permitir el funcionamiento del programa final:  
**WARNINGS.**

Ejemplo: Cuando se declara una variable que luego no se utiliza.



# Intérpretes

- # El programa fuente va traduciéndose y ejecutándose directamente sentencia a sentencia, sin crear ningún fichero objeto, hasta que encuentra un error o finaliza.
- # No se obtiene ningún ejecutable, el programa se interpreta cada vez.
- # Inconvenientes:
  - El proceso de análisis se realiza siempre con lo que es más lento que los lenguajes compilados



# Máquinas Virtuales

- # En este sistema de máquinas virtuales se realiza una compilación y posteriormente una interpretación.
- # Comenzó a utilizarse con la llegada de la plataformas Java de Sun Microsystems, y .NET de Microsoft.



# Funcionamiento de las máquinas virtuales

- # El código fuente se compila, detectando los errores sintácticos, y se genera una especie de ejecutable, con un código máquina (código intermedio, lenguaje intermedio, p-code, o **byte-code**) dirigido a una CPU imaginaria.
- # Como esa CPU imaginaria no existe, para poder ejecutar ese ejecutable, se construye un intérprete. Este intérprete es capaz de leer cada una de las instrucciones de código máquina imaginario y ejecutarlas en una CPU real. A este intérprete se le denomina máquina virtual.