Introducción a la Programación

Elementos de Programación

PROGRAMACIÓN

Hacer que las computadoras hagan lo que nosotros queremos que hagan.

Ámbitos de aplicación:

- **Control de procesos industriales, máquinas, herramientas.**
- **Electrodomésticos, comunicaciones.**
- **Gestión de negocios, oficina.**

Objetivo:
Definir los pasos a seguir para resolver un problema dado y conseguir un resultado esperado: CONSISTENTE, REPETIBLE Y CONFIABLE.

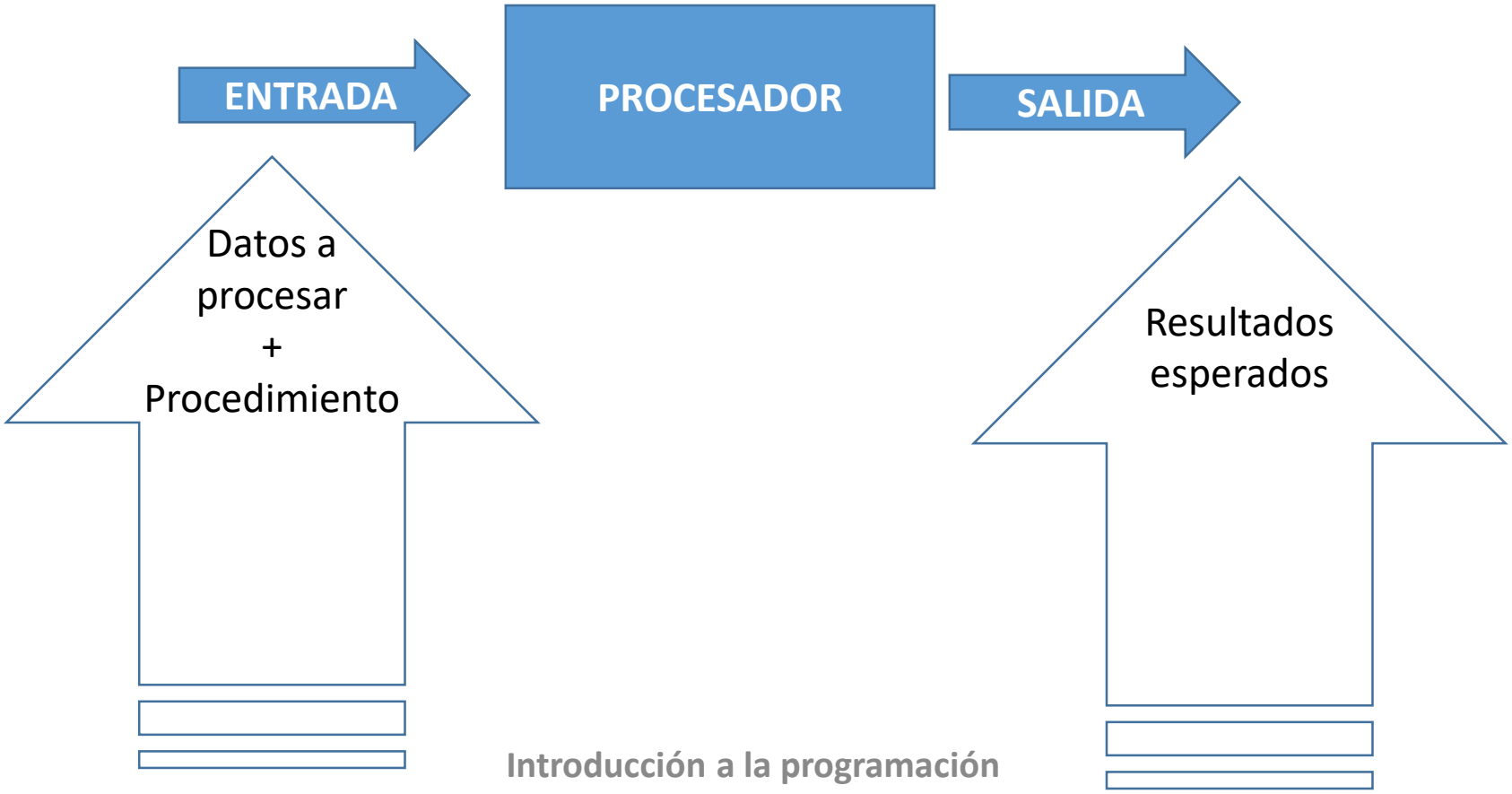
PROGRAMA

- **Son instrucciones ejecutables por el procesador que llevan a cabo un *algoritmo* que permiten resolver un problema dado.**

- **El algoritmo normalmente define SIEMPRE el mismo comportamiento para un conjunto de variables.**
- **El primer paso es encontrar el ALGORTIMO, es decir el mecanismo que – codificado en un cierto lenguaje- nos permita resolver el problema.**

- **Este mecanismo comprende 2 conceptos:**
  - **DATOS → Que representan "objetos" sobre los que operamos.**
  - **PROCEDIMIENTOS → Que son las reglas que definen cómo manipular los datos.**

PROCESADOR


ENTRADA
PROCESADOR
SALIDA
Datos a procesar + Procedimiento
Resultados esperados

PROCESADOR

- **Es quien "ejecuta" el programa y hace visible el resultado esperado del programa.**

MISMA ESTRUCTURA, distintos TAMAÑOS:

- **Marcapasos cardíaco, implantes inteligentes.**
- **Micros embebidos (PDA, celular, mobile, etc).**
- **Microprocesadores (Notebooks & PCS).**
- **Microprocesadores de alta gama (Midrange).**
- **Procesadores híbridos (Mainframe)**

EL LENGUAJE DEL PROCESADOR

SISTEMA BINARIO:

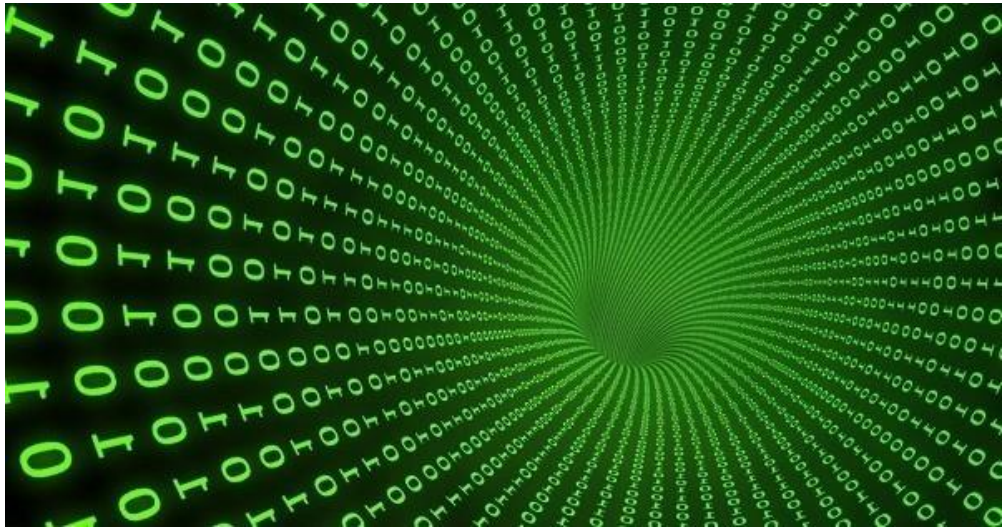
- **1101 1010 0100 1000 (DA68) → Activar motor #4.**

Originalmente especificado a través de diversos mecanismos:

- **Cintas/ tarjetas perforadoras (Jacquard)**
- **Llaves (On- off).**

Lenguajes:

- **Bajo nivel: Assembler → MOV A, X**
- **Alto nivel: COBOL, Java, Phyton → print PI * pow(x,2)/2**



CARACTERÍSTICAS

La mayoría de los lenguajes (para no decir todos) son capaces de realizar:

- **Operaciones aritméticas.**
- **Manipulación y análisis de texto.**
- **Loops: ejecutar grupo de instrucciones repetidamente.**
- **Condicionales: ejecutar grupo de instrucciones según criterio.**
- **Entrada/salida: recibir y entregar datos de y al mundo exterior.**
- **Funciones y procedimientos para manipular los datos.**
- **Utilizar librerías de funciones y procedimientos “prefabricados”.**
- **Organizar programa combinando estructuras simples para formar otras complejas.**

CÓDIGO FUENTE

- **Se conoce como "código fuente" (source code) al documento de texto que contiene las instrucciones en el lenguaje correspondiente:**

```
__author__ = "Mark Pilgrim (mark@diveintopython.org)"
__version__ = "$Revision: 1.2 $"
__date__ = "$Date: 2004/05/05 21:57:19 $"
__copyright__ = "Copyright (c) 2004 Mark Pilgrim"
__license__ = "Python"

def fibonacci(max):
    a, b = 0, 1
    while a < max:
        yield a
        a, b = b, a+b

for n in fibonacci(1000):
    print n,
```

EJECUCIÓN DEL CÓDIGO

El código fuente contiene el programa en un formato legible, en el lenguaje de alto nivel correspondiente (ej: COBOL, Java, Phyton)

Existen varios mecanismos para que el procesador EJECUTE las instrucciones del código:

- **INTERPRETADO: el código fuente se lee y ejecuta instrucción por instrucción, con poca ó ninguna optimización.**
- **COMPILADO: el código fuente es procesado por un compilador el cual genera:**
  - **Objeto: ejecutable (binario): optimizado por la plataforma de hardware y sistema operativo correspondiente.**
  - **Bytecodes: es un paso intermedio, que luego será interpretado por una máquina virtual VM.**

COMPILADOR VS. INTÉRPRETE

COMPILADOR

Genera un objeto binario:

- **Muy compacto.**
- **Optimizado.**
- **Específico para el sistema operativo/ procesador objetivo.**

INTÉRPRETE

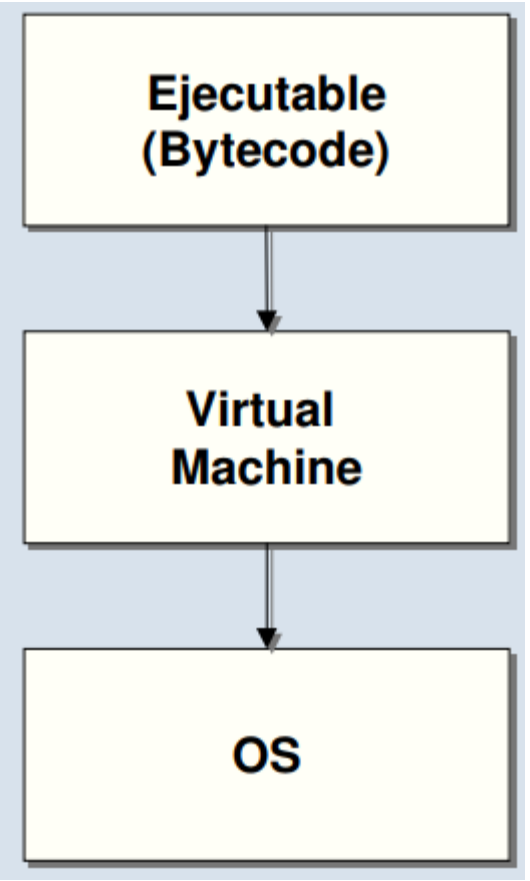
Ejecuta funciones paso a paso (script)

- **Muy flexible, se cambia el código y se corre.**
- **Se interpreta en el momento.**
- **No está optimizado.**

MÁQUINA VIRTUAL

Para que el ejecutable se independiente del sistema operativo (portable) se ejecuta en una MV.

- **Ésta "máquina" interpreta el programa compilado (bytecode).**
- **Cada sistema operativo tiene una versión de la MV, pero todas son iguales de cara al programa compilado.**



CICLO DE VIDA DEL SOFTWARE

- ✓ **Diseñar el algoritmo de solución.**
- ✓ **Escribir el código fuente y generar el objetivo ejecutable (script/bytecode/exe).**
- ✓ **Ejecutar la solución y verificar el resultado.**
- ✓ **Debug.**