

Tema 2. Funcionamiento del computador

1. Programación

<http://lorca.act.uji.es/ig09/>

Sergio Barrachina

Rafael Mayo

Índice General

| | | |
|----------|---|-----------|
| 1 | Concepto de <i>Software</i> | 3 |
| 1.1 | Características | 4 |
| 1.2 | Aplicaciones | 5 |
| 1.3 | Definición de programa de computador | 7 |
| 1.4 | Resolución de problemas | 8 |
| 1.5 | Algoritmos | 9 |
| 1.6 | Instrucciones y datos | 11 |
| 1.7 | Ejemplos de algoritmos | 13 |
| 2 | Lenguajes de programación | 15 |
| 2.1 | Compilación y ensamblado | 16 |
| 2.2 | Ventajas de los lenguajes de alto nivel | 18 |

1 Concepto de *Software*

Hardware vs. Software

Hardware Término utilizado originalmente para designar utensilios o herramientas de metal. En informática se utiliza para designar al conjunto de elementos *físicos* que componen un computador (tornillos, caja, ventilador, circuitos integrados, cables, ...)

Software Aparece por contraposición al término *hardware*. Es la parte intangible de un sistema de computación (no es físico).

1.1 Características

- Se *desarrolla*, no se fabrica en el sentido clásico:
 - ⇨ Al igual que el hardware:
 - ➡ Buen diseño → buena calidad.
 - ⇨ Al contrario que el hardware:
 - ➡ Errores de construcción fácilmente corregibles (o no hay).
 - ➡ Distinta relación entre cantidad de gente dedicada ↔ trabajo realizado.
 - ➡ Costes prácticamente sólo en la fase de ingeniería.
- No se estropea (¡pero conforme se modifica, se deteriora!).
- Una gran parte se realiza a medida (esta situación está cambiando rápidamente gracias a la programación orientada a objetos).

1.2 Aplicaciones

- **Software de sistemas** Sirve a otros programas, interactúa con el *hardware* de la computadora (compiladores, gestores, ...)
- **Software de tiempo real** mide/analiza/controla sucesos del mundo real conforme ocurren.
- **Software de gestión** Procesa información comercial.
- **Software de ingeniería o científico** “Manejo de números”.


1.2 Aplicaciones (II)

- **Software empotrado** Controla productos y sistemas de los mercados industriales y de consumo (hornos industriales, lavadoras, ...).
- **Software de PC** Procesamiento de textos, hojas de cálculo, bases de datos, gráficos, entretenimiento, ...
- **Software de inteligencia artificial** Sistemas expertos o basados en el conocimiento. Reconocimiento de patrones, habla, ...

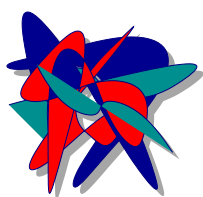
1.3 Definición de programa de computador

El software está formado por[2]:

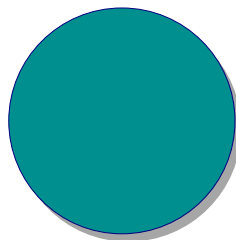
- instrucciones que al ser ejecutadas por un computador realizan una determinada función (*programa*),
- estructuras de datos que facilitan la manipulación de la información,
- (la documentación que describe la operación y uso de los programas.)

 **Programa de computador** Secuencia de instrucciones susceptibles de ser ejecutadas por un computador para realizar una determinada función.

1.4 Resolución de problemas



Problema



Algoritmo



Programa

1.5 Algoritmos

Algoritmo[3] Solución de un problema expresada con la suficiente precisión como para poder ser descompuesta en acciones realizables por el computador.

- Acciones bien definidas.
- Secuencia finita de operaciones.
- Debe acabar en un tiempo finito.

1.5 Algoritmos (II)

| | |
|--------|------|
| 857036 | 472 |
| 3850 | 1815 |
| 0743 | |
| 2716 | |
| 356 | |

1. Tantas cifras del dividendo como tenga el divisor
2. Buscar multiplicador
3. Multiplicar y restar
4. Si quedan cifras, bajar la siguiente y volver a 2

1.6 Instrucciones y datos

| Algoritmo | | Programa |
|---------------|---|---------------|
| ¿Qué hacer? | ⇒ | Instrucciones |
| ¿Sobre quién? | ⇒ | Datos |

- Instrucciones:
Seleccionar cifras, buscar multiplicador, bajar cifra, ...
- Datos:
Dividendo, divisor, resto, cociente, ...

1.6 Instrucciones y datos (II)

Se denomina **dato** a cualquier objeto manipulable por el computador[3]. (carácter leído del teclado, información almacenada en un disco, un número en memoria, ...)

Tipos de datos básicos:

- **Entero sin signo.** (p.e. números del 0000 al 9999)
- **Entero.** (p.e. números del -999 al $+999$)
- **Reales** (en coma flotante). (p.e. del $-0,999 \cdot 10^{-99}$ al $0,999 \cdot 10^{+99}$)
- **Valores lógicos:** *verdadero* o *falso*. (p.e. comparación)
- **Caracteres.** (p.e. letras 'a' a la 'z', 'A' a la 'Z', el '?', ...)

1.7 Ejemplos de algoritmos

Problema: Encontrar si un número positivo (> 0) es par.

➤ Algoritmo:

1. leer N
2. si $N == 2$ entonces escribe “Es par”; fin
3. si $N == 1$ entonces escribe “Es impar”; fin
4. $N = N - 2$
5. ir a 2

Coste del algoritmo: $(N + 1)/2$ iteraciones

1.7 Ejemplos de algoritmos (II)

Problema: Encontrar si un número positivo (> 0) es par.

➤ Algoritmo II:

1. leer N
2. $M = 2 \cdot \text{int}(N/2)$ // *int es la función parte entera*
3. si $M == N$ entonces escribe “Es par”
4. si no escribe “Es impar”
5. fin

2 Lenguajes de programación

- ▶ Lenguajes de bajo nivel (1ª generación)
 - ↳ Lenguaje máquina
 - ↳ Lenguaje ensamblador
- ▶ Lenguajes de alto nivel
 - ↳ Fortran, Cobol, Algol y Basic. (2ª generación)
 - ↳ PL/1, Pascal, Modula-2, C y Ada. (3ª propósito general)
 - ↳ C++, Objective-C, SmallTalk y Eiffel. (3ª propósito general OO)
 - ↳ Lisp, Prolog, APL y Forth. (3ª propósito específico)
 - ↳ Lenguajes de petición y generadores de programas (4ª generación).

2.1 Compilación y ensamblado

Lenguaje de
alto nivel (en C)

```
swap(int a, int b) {  
    int temp;  
    temp = b;  
    b = a;  
    a = temp; }
```

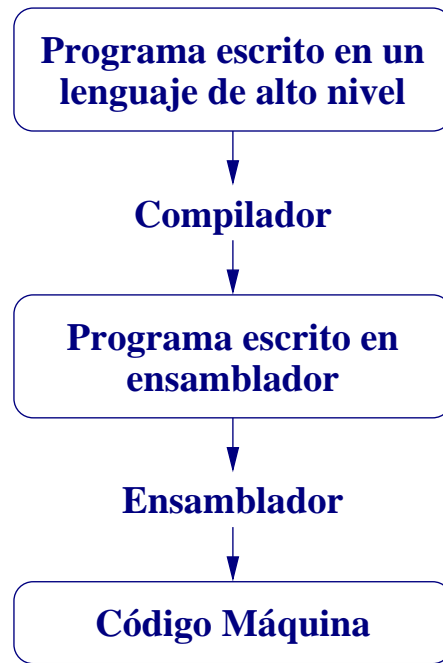
Lenguaje
ensamblador
(para MIPS)

```
swap: lw  $15,0($4)  
      lw  $16,0($5)  
      sw  $16,0($4)  
      sw  $15,0($5)  
      jr  $31
```

Lenguaje
máquina
(para MIPS)

```
10001100100011110000000000000000  
10001100101100000000000000000000  
10101100100100000000000000000000  
10101100101011110000000000000000  
00000011111000000000000000000100
```


2.1 Compilación y ensamblado (II)



2.2 Ventajas de los lenguajes de alto nivel

- Permiten al programador pensar en un lenguaje más natural:
 - ⇨ Lenguajes diseñados para un determinado campo de aplicación.
- Mejora la productividad del programador:
 - ⇨ Menos tiempo de desarrollo cuando se utilizan lenguajes que requieren menos líneas para expresar una idea.
- Los programas son independientes de la máquina en la que se desarrollan:
 - ⇨ Los compiladores y ensambladores pueden traducir el programa a las instrucciones máquina de cualquier computador.

Referencias

- [1] David A. Patterson y John L. Hennessy (1994). *Computer organization & design. The hardware/software interface*. Morgan Kaufmann Publishers.
- [2] Roger S. Pressman (1998). *Ingeniería del Software. Un enfoque práctico*. McGraw-Hill, 4^a edición.
- [3] Alberto Prieto, Antonio Lloris y Juan Carlos Torres (1995). *Introducción a la informática*. McGraw-Hill.