

Computación, Ordenadores y Python

Fundamentos de Computación

Grado en Ingeniería en Tecnologías Industriales

Universidad de Cantabria

24/9/2018

- **INFORMÁTICA** : conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de computadoras.
- **COMPUTADOR, COMPUTADORA U ORDENADOR** es una máquina que acepta información de entrada y efectúa con ella operaciones lógicas y aritméticas. Ejecutando ordenadamente una secuencia de instrucciones previamente almacenadas en la propia máquina, a la que llamaremos **programa**. Proporcionando la información resultante a través de un medio de salida.
Consecuentemente el ordenador está compuesto:

- **Hardware**: equipo electrónico.
- **Software**: conjunto de programas.

- Si disponemos de las horas de entrada y de salida de miles de vehículos que han atravesado un túnel, es muy sencillo encargar a un ordenador que calcule parámetros estadísticos relativos a la velocidad
- Si tenemos cientos de fotografías en una tarjeta de memoria, la tarea de comprimirlas a un formato más reducido es un cálculo que podemos automatizar mediante un programa.

- La raíz cuadrada de un número x es y tal que $y * y = x$
- Receta para deducir la raíz cuadrada de un número x (16)
 - 1 Conjeturamos un número, g
 - 2 Si $g * g$ es suficientemente próximo a x , paramos y decimos que g es la respuesta.
 - 3 En otro caso, realizamos una nueva conjetura haciendo la media entre g y x/g
 - 4 Usando la nueva conjetura, repetimos el proceso hasta encontrar una suficientemente próxima.

g	g^2	x/g	$(g+x/g) / 2$
3	9	16/3	4.17
4.17	17.36	3.837	4.0035
4.0035	16.0277	3.997	4.000002

En el interior del computador la información(digital) se almacena y se transfiere de un sitio (unidad) a otro mediante un código que utiliza sólo dos símbolos (código binario) denotados por 0 y 1.

Por ello, los datos (números, letras, símbolos, ...) deben ser representados mediante combinaciones de ceros y unos según una **codificación**.

Informalmente, codificar la información es reescribirla, incluso en otro **alfabeto** distinto, para adaptarla a las condiciones de la transmisión.

Codificar (binario) es una aplicación inyectiva c de los elementos (mensajes) de un conjunto finito (**alfabeto fuente**) \mathcal{A} al **alfabeto código** $\{0, 1\}^n$.

$$c : \mathcal{A} \rightarrow \{0, 1\}^n$$

Una vez recibido y manipulado el mensaje, es preciso **Descodificarlo**, es decir, traducirlo al alfabeto fuente.

En las unidades de entrada y salida del ordenador se realizan estas transformaciones.

Para **CLAUDE SHANNON** investigador de la *Bell Telephone (EEUU)* e inventor de la teoría de la información en 1947:

la **medida de información** de un mensaje es totalmente independiente de su contenido semántico, y sólo depende de la probabilidad de producción del mismo por parte de una alguna **fente de información**.

Que un perro muerda a un hombre no es noticia; sí lo es que un hombre muerda a un perro.

La cantidad de información que contiene cada uno de los símbolos de una fuente de información con dos únicos símbolos 0, 1 y, que les emite con la misma probabilidad $\frac{1}{2}$ es denominada

bit (de la contracción de Blnary digiT)

Concatenando bits, podemos obtener mensajes con mayor cantidad de información. Así:

- 1 bit = 2 mensajes (0, 1)
- 2 bits = 4 mensajes (00, 01, 10, 11)
- 3 bits = 8 mensajes (000, 001,...111)
- n bits = 2^n mensajes.

Otra unidad significativa de información es:

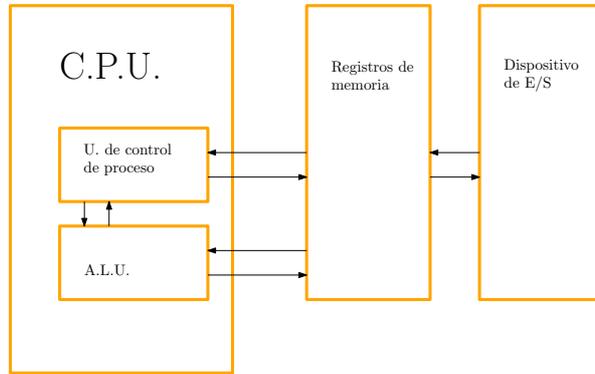
Byte: grupo o mensaje de 8 bits → 2^8 mensajes.

Otras unidades de información o (almacenamiento de información) son:

- Kilobyte (Kb) = 2^{10} bytes = 1024 bytes $\equiv 10^3$ bytes.
- Megabyte (Mb) = 2^{10} Kb = 2^{20} bytes = 1,048,576 bytes $\equiv 10^6$ bytes.
- Gigabyte (Gb) = 2^{10} Mb = 2^{30} bytes = 1,073,741,824 bytes $\equiv 10^9$ bytes.
- Terabyte (Tb) = 2^{10} Gb = 2^{40} bytes $\equiv 10^{12}$ bytes.
- Petabyte (Pb) = 2^{10} Tb = 2^{50} bytes $\equiv 10^{15}$ bytes.

Según el Sistema Internacional de Medidas, para las potencias binarias deberían denominarse Kibibyte(KiB), Mebibyte(MiB), etc. Y reservarse Kilobyte(Kb), Megabyte(Mb) etc, para las potencias decimales.

Probabilidad de ser fulminado por un rayo (por día)	1 entre 9.000.000.000 (2^{33})
Probabilidad de ganar la Lotería Primitiva Española	1 entre 13.983.816 (2^{23})
Probabilidad de ganar la Primitiva y caer fulminado por un rayo el mismo día	1 entre 2^{56}
Tiempo hasta la próxima glaciación	14.000 (2^{14}) años
Tiempo hasta que el Sol se extinga	10^8 (2^{28}) años
Edad del Planeta Tierra	10^8 (2^{28}) años
Edad del Universo	10^{10} (2^{34}) años
Número de átomos en el Planeta Tierra	10^{21} (2^{70})
Número de átomos en el Sol	10^{27} (2^{88})
Número de átomos en la Vía Láctea	10^{67} (2^{222})
Número de átomos en el Universo (excluyendo materia oscura)	10^{77} (2^{255})
Masa de la Tierra	$5,9 \times 10^{24}$ (2^{82}) Kg.
Masa del Sol	2×10^{30} (2^{100}) Kg.
Masa estimada del Universo (excluyendo materia oscura)	10^{50} (2^{166}) Kg.
Volumen de la Tierra	10^{21} (2^{69}) m ³
Volumen del Sol	10^{27} (2^{89}) m ³
Volumen estimado del Universo	10^{82} (2^{272}) m ³



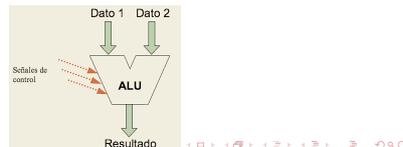
- **UNIDADES DE ENTRADA:** dispositivos que sirven para introducir en el computador los datos e instrucciones. Transforman las informaciones de entrada en señales binarias de naturaleza eléctrica. Ejemplos: un teclado, un digitalizador, INTERNET, una lectora de tarjetas de crédito, etc.
- **UNIDADES DE SALIDA:** son dispositivos a través de los cuales se obtienen los resultados de los programas ejecutados por el computador. Transforman las señales eléctricas binarias en caracteres escritos o gráficos que son visualizados. Ejemplos: un monitor de vídeo, una impresora, INTERNET, etc.
- **MEMORIA:** dispositivos donde se almacenan tanto los datos como las instrucciones.
- **UNIDAD DE CONTROL (UC):** parte central del computador, encargada de interpretar las instrucciones del programa y ordenar su ejecución, generando las señales de control al resto de las unidades.
- **UNIDAD ARITMÉTICA-LÓGICA (ALU) :** contiene los circuitos electrónicos con los que se hacen las operaciones de tipo aritmético (sumas, restas, etc.) y de tipo lógico (comparar dos números, hacer operaciones del Álgebra de Boole binaria, etc.). La ALU y UC constituyen la CPU (Central Processing Unit).

La unidad aritmético-lógica o ALU (Arithmetic Logic Unit) o Camino de datos. Contiene los circuitos electrónicos con los que se hacen las operaciones:

- de tipo aritmético (sumas, restas, etc.) y;
- de tipo lógico (comparar dos números, hacer operaciones del Algebra de Boole binaria, etc.).

Suele operar con datos de 16, 32, 64 o 128 bits. Se denomina **palabra** al conjunto de bits que forma un dato y **la longitud de una palabra** es el número de bits que la forma. Tiene un conjunto pequeño de memorias denominadas registros de la CPU.

- En cada registro cabe un dato.
- Se usan para guardar los datos que se están usando en ese momento.



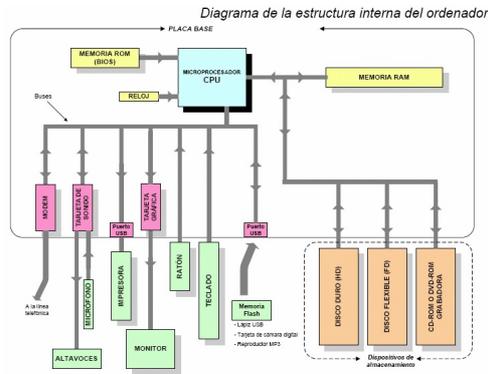
La CPU realiza las operaciones más importantes, contiene un reloj que sincroniza todas las operaciones elementales de la computadora. El periodo del reloj se denomina tiempo de ciclo T y es del orden de nanosegundos. La frecuencia del reloj $F = 1/T$ (medida en Megahercios, MHz, o Gigahercios, GHz) determina, en parte, la velocidad de funcionamiento del computador.

Componentes: Unidad Aritmético-Lógica (ALU) y Unidad de Control (U.C.) Denominación: C.P.U o Procesador, Microprocesador (todo integrado en la misma pastilla, (ALU, UC, memoria caché).

Funciona repitiendo constantemente lo siguiente:

- 1 lee de la memoria una instrucción del programa.
- 2 comprueba cuál es su significado.
- 3 organiza la secuencia de acciones a realizar por cada elemento electrónico del computador para ejecutar la instrucción.

El conjunto de conductores que transmite información entre unidades distintas se denomina **bus**. El **ancho de un bus** es el número de hilos o número de bits que transmite silmultaneamente en paralelo. (Hercios = ciclos/segundo. Megahercios = millones de ciclos/segundo).



La memoria es un elemento fundamental. Su capacidad (medida en Bytes, Megabytes y Gigabytes) y tiempo de acceso fijan, en gran medida, la potencia del ordenador. Se puede clasificar según su velocidad de acceso y capacidad:

Memoria Principal (MP)

- Constituida por circuitos integrados.
- Ligada directamente a la UC y a la ALU, es de acceso rápido.
- Para que un programa se ejecute, sus instrucciones deben estar almacenadas en la memoria principal.
- Esta divida en:
 - Memoria ROM. Sólo de lectura y permanente. Almacena la BIOS (Basic Input-Output System)
 - Memoria RAM. De lectura y escritura, es volátil.
- Estructurada en palabras, para leer o escribir una información es necesario dar la dirección de su palabra.

Memoria Masiva o auxiliar o externa (MM)

- Está constituida por dispositivos tales como discos y cintas magnéticos y ópticos.
- Tiene más capacidad que la memoria principal.
- Es más lenta que la memoria principal.
- Es permanente. Los datos y programas se suelen grabar (introduciéndolos por las unidades de entrada) en la memoria masiva. La información guardada en un disco o cinta permanece indefinidamente hasta que el usuario expresamente la borre.

Registros de la CPU

Internos del sistema, y de acceso muy rápido

Memoria de Caché

- Situada entre la CPU y la memoria principal.
- Alta velocidad.
- Para datos e instrucciones más frecuentemente usados por el computador.

- **Instrucción** conjunto de símbolos que representan una orden de operación o tratamiento para la computadora. Estas operaciones suelen realizarse con o sobre datos.
- **Programa** conjunto ordenado de instrucciones que se dan a la computadora para indicarle las operaciones o tareas que ha de realizar.

Las instrucciones se forman con elementos o símbolos tomados de un determinado repertorio, y se construyen siguiendo unas reglas precisas.

Soporte lógico o Software es el conjunto de programas asociados al computador. Dentro de estos programas se incluyen los suministrados por el constructor, los adquiridos en empresas especializadas, y los redactados por los propios usuarios.

- Sistema
 - Control
 - Sistemas operativos
 - Intérprete de ordenes
 - Diagnostico y mantenimiento
 - Utilidades
- Aplicaciones

Un computador contiene un conjunto de programas, denominado **SISTEMA OPERATIVO (SO)** cuya función es controlar el funcionamiento del computador y poder utilizarlo eficiente y cómodamente.

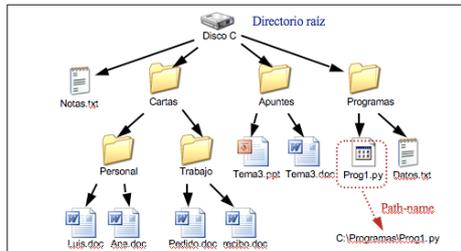
Por ejemplo, un SO de multiprogramación hace que cuando un programa A se esté ejecutando, y tenga una operación de entrada/salida, el tiempo muerto de uso del procesador sea aprovechado para la ejecución de otro programa B. Es una interfaz entre la máquina y los usuarios o programas de aplicación. Su objetivo fundamental es gestionar eficientemente los recursos hardware:

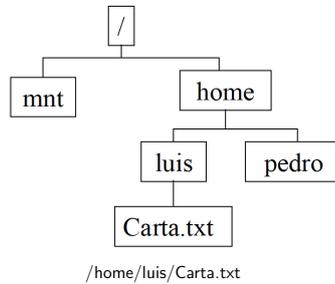
- CPU: Gestión de proceso y gestión de la memoria principal.
- Gestión de periféricos: discos, impresoras, etc.

Los programas del SO se utilizan mediante un lenguaje específico denominado **lenguaje de control**, sus instrucciones se llaman órdenes (commands").
 Ejemplos: MS-DOS, UNIX, Windows, LINUX, MAC OS X.

La mayor parte de los SO incorporan **entornos gráficos** gracias a los cuales, el usuario no precisa utilizar el lenguaje de control bastándole con elegir, gráficamente, alternativas que el sistema operativo le ofrece.

El S.O. posibilita que el usuario no tenga que utilizar direcciones físicas, introduciendo los conceptos de **archivo y directorio (carpeta)**. Cuando se abre un archivo, el S.O. utiliza la **ruta (path-name)**, que contiene una lista de todas las carpetas atravesadas desde la carpeta raíz al directorio en cuestión y extrae, a partir de la información del elemento la tabla de direcciones en disco y la ubica en memoria principal.





El **lenguaje máquina** es el único que entienden los circuitos del computador (CPU). El código de máquina codifica las secuencias de instrucciones como sucesiones de unos y ceros que siguen ciertas reglas. Cada familia de ordenadores dispone de su propio repertorio de instrucciones, es decir, de su propio **código de máquina**.

Sus instrucciones se forman por bits agrupados en campos:

- **Campo de código de operación:** indica la operación correspondiente a la instrucción.
- **Campos de dirección:** especifican los lugares (o posiciones) donde se encuentran los datos con los que se opera o donde hay que ubicar los resultados de la operación.

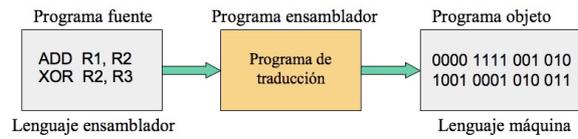
INCONVENIENTES:

- Depende del modelo de computadora
- Contiene sólo operaciones muy elementales
- Repertorio de instrucciones muy reducido
- Inexpresivo para el ser humano

El primer intento de mejora del lenguaje de programación consistió en usar una codificación hexadecimal (en base 16) en lugar del binario. Con ello, las expresiones son más cortas y las posibilidades de error en un dígito se reducen. El **lenguaje ensamblador** fue la siguiente mejora para facilitar la programación en instrucciones de máquina.

Esta simplificación tiene tres aspectos fundamentales:

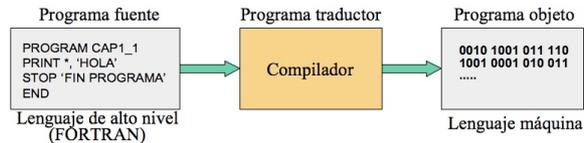
- Empleo de **códigos nemóticos** para representar las instrucciones.
- Empleo de **nombres simbólicos** para representar las direcciones.
- Cada sentencia corresponde a una instrucción de máquina.



Cada CPU tiene su propio instrucciones y, en consecuencia, un lenguaje de máquina y uno o más lenguajes ensambladores propios. Un programa escrito para una CPU de la marca Intel-Core no funcionará en una CPU diseñada por otro fabricante, como Apple.

Los lenguajes de alto nivel están orientados a facilitar la tarea de la programación, disponen de instrucciones más potentes que se expresan de forma cómoda y comprensible. Cada una de ellas equivale a muchas instrucciones en lenguaje máquina.

El computador no entiende estos lenguajes por lo que es necesaria una traducción previa a lenguaje máquina para su ejecución:



- **Compiladores:** traducen todo el programa fuente y generan un programa objeto que se puede ejecutar las veces que se desee.
- **Intérpretes:** ejecutan las instrucciones según las van traduciendo sentencia a sentencia. No generan un programa objeto.

Una Torre de Babel

- **FORTRAN** (Formula Translation): Aplicaciones científico-técnicas (grandes computadores y supercomputadores)
- **COBOL** (COmmon Busines Oriented Language): 1960. Aplicaciones comerciales y de gestión.
- **BASIC** (Beginner's All-purpose Symbolic Instruction Code). Desarrollado a mediados de los 60 como lenguaje interactivo para principiantes de programación. Visual BASIC es la versión de Microsoft.
- **C**: Desarrollado en Bell Labs, a comienzos de los 70. Es más complejo que los anteriores, pero también es más potente, flexible y eficiente.
- **Pascal**. Creado por Wirth en 1971. El mejor lenguaje para aprender a programar y describir algoritmos.
- **Ada**: Es un lenguaje definido por el Ministerio de Defensa de USA a finales de los 70. Esta basado en el Pascal y tiene unas reglas muy estrictas.
- **C++**: Ideado a comienzos de los 80 en los Bell Labs. Es una variante del C que permite utilizar la moderna metodología de la programación ("programación orientada a objetos")
- **Java**: Desarrollado en 1991 por Sun, es similar a C++ pero más sencillo de aprender y usar. Muy usado para programa interactivos y dinámicos ("applets" de web).
- **Perl, PHP** y otros muchos más

- Finales de los 80, habitualmente interpretado, orientado a objetos, con tipos dinámicos; hace énfasis en la legibilidad.
- Guido van Rossum. Centrum Wiskunde & Informatica (Amsterdam)



(BDFL) Benevolente Dictador Vitalicio.

- Python 1 (1994), Python 2 (2000) y Python 3 (2010)
- Libre y se puede instalar en Linux, Windows y Mac.
- Python Software Foundation. <http://www.python.org/>

```
MacBook-Pro-de-jaimE:Desktop jaimE$ python3
Python 3.6.1 (v3.6.1:69e0db56869, Mar 21 2017, 01:21:04)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

- **Numpy**: Generación de tipos de datos científicos. <http://www.numpy.org/>
- **Scipy**: Funciones científicas de uso general. <http://www.scipy.org/>
- **Matplotlib**: Gráficas en 2D y 3D. <http://matplotlib.org/>
- **Visual Python**: Permite realizar representaciones tridimensionales de objetos en movimiento. <http://www.vpython.org/>
- **SymPy**: Cálculo simbólico. <http://sympy.org/en/index.html>
- **Opencv**: Biblioteca libre de visión artificial en C cuyas librerías se han adaptado a Python. <http://opencv.willowgarage.com/documentation/python/>
- **Kivy**: Crear apps para Android y aplicaciones multitouch en Windows, Mac OS X y Linux.
- **Rpy**: Estadística.
- **BioPython**: Bioinformática.
- **SageMath**: Open source mathematics software. <http://www.sagemath.org/>

- MIT: Massachusetts Institute of Technology.
- Stanford: Stanford University.
- UC Berkeley : University of California Berkeley.
- Cornell: Cornell University.
- INRIA: Université de Nice-Sophia Antipolis.
- UZH: Universität Zürich.
- TUM: Technische Universität München.
- UAM: Universidad Autónoma de Madrid.
- UPC: Universitat Politècnica de Catalunya.
- UJI: Universitat Jaume I.

