

# Introducción

Ricardo Pérez López

IES Doñana, curso 2019/2020

## Índice general

<b>1. Conceptos básicos</b>	<b>2</b>
1.1. Informática . . . . .	2
1.2. Ordenador . . . . .	3
1.3. Algoritmo . . . . .	6
1.4. Programa . . . . .	8
1.5. Lenguaje de programación . . . . .	8
<b>2. Evolución histórica</b>	<b>8</b>
2.1. Culturas de la programación . . . . .	8
2.2. Ingeniería del software . . . . .	8
<b>3. Resolución de problemas mediante programación</b>	<b>8</b>
3.1. Análisis del problema . . . . .	8
3.2. Especificación . . . . .	8
3.3. Diseño del algoritmo . . . . .	8
3.4. Codificación del algoritmo en forma de programa . . . . .	8
<b>4. Paradigmas de programación</b>	<b>8</b>
4.1. Imperativo . . . . .	8
4.2. Declarativo . . . . .	8
<b>5. Lenguajes de programación</b>	<b>8</b>
5.1. Definición . . . . .	8
5.2. Evolución histórica . . . . .	9
5.3. Clasificación . . . . .	9
<b>6. Traductores</b>	<b>9</b>
6.1. Compiladores . . . . .	9
6.2. Intérpretes . . . . .	9
<b>7. Entornos integrados de desarrollo</b>	<b>9</b>
7.1. Terminal . . . . .	9
7.2. Editores de texto . . . . .	9

# 1. Conceptos básicos

## 1.1. Informática

- Definición:

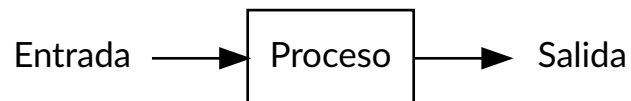
**Informática:**

La ciencia que estudia los sistemas de tratamiento automático de la información, también llamados **sistemas informáticos**.

- Estos sistemas están formados por:
  - elementos físicos (**hardware**)
  - elementos lógicos (**software**) y
  - elementos humanos (profesionales y usuarios).
- El *hardware*, a su vez, está formado por componentes:
  - Ordenadores
  - Soportes de almacenamiento
  - Redes de comunicaciones
  - ...

### 1.1.1. Procesamiento automático

- El procesamiento automático de la información siempre tiene el mismo esquema de funcionamiento:



- El **objetivo** del procesamiento automático de la información es **convertir los datos de entrada en datos de salida** mediante un *hardware* que ejecuta las instrucciones definidas por un *software* (**programas**).
- Los programas gobiernan el funcionamiento del *hardware*, indicándole qué tiene que hacer y cómo.
- La **Programación** es la ciencia y el arte de diseñar dichos programas.

## 1.2. Ordenador

### 1.2.1. Definición

**Ordenador:**

Un ordenador es una máquina que procesa información automáticamente de acuerdo con un programa almacenado.

1. Es una *máquina*.
2. Su función es *procesar información*.
3. El procesamiento se realiza de forma *automática*.
4. El procesamiento se realiza siguiendo un *programa (software)*.
5. Este programa está *almacenado* en una memoria interna del mismo ordenador (arquitectura de **Von Neumann**).

### 1.2.2. Funcionamiento básico

#### 1.2.2.1. Elementos funcionales

- Un ordenador consta de tres componentes principales:

1. **Unidad central de proceso (CPU) o procesador**

- *Unidad aritmético-lógica (ALU)*
- *Unidad de control (UC)*

2. **Memoria**

- *Memoria principal o central*
  - \* *Memoria de acceso aleatorio (RAM)*
  - \* *Memoria de sólo lectura (ROM)*
- *Memoria secundaria o externa*

3. **Dispositivos de E/S**

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

#### 1.2.2.2. Unidad central de proceso (CPU) o procesador

- **Unidad aritmético-lógica (ALU):**

Realiza los cálculos y el procesamiento numérico y lógico.

- **Unidad de control (UC):**

Ejecuta de las instrucciones enviando las señales a las distintas unidades funcionales involucradas.

### 1.2.2.3. Memoria

- **Memoria principal** o central:

Almacena los datos y los programas que los manipulan.

Ambos (datos y programas) deben estar en la memoria principal para que la CPU pueda acceder a ellos.

Dos tipos:

- **Memoria de acceso aleatorio (RAM):**

Su contenido se borra al apagar el ordenador.

- **Memoria de sólo lectura (ROM):**

Información permanente (ni se borra ni se puede cambiar).

Contiene la información esencial (datos y software) para que el ordenador pueda arrancar.

- **Memoria secundaria** o externa:

La información no se pierde al apagar el ordenador.

Más lenta que la memoria principal, pero de mucha más capacidad.

### 1.2.2.4. Dispositivos de E/S

- **Dispositivos de entrada:**

Introducen datos en el ordenador (*ejemplos*: teclado, ratón, escáner...)

- **Dispositivos de salida:**

Vuelcan datos fuera del ordenador (*ejemplos*: pantalla, impresora...)

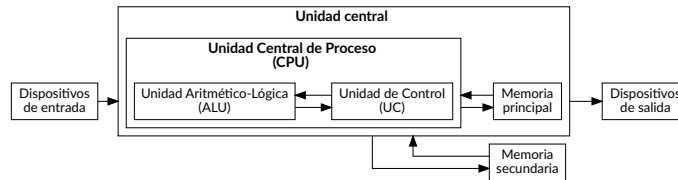
- **Dispositivos de entrada/salida:**

Actúan simultáneamente como dispositivos de entrada y de salida (*ejemplos*: pantalla táctil, adaptador de red...)

- Los dispositivos que acceden a **soportes de almacenamiento masivo** (las **memorias secundarias**) también se pueden considerar dispositivos de E/S:

- Los soportes de **sólo lectura** se leen con dispositivos de entrada (*ejemplo*: discos ópticos).
  - Los soportes de **lectura/escritura** operan como dispositivos de entrada/salida (*ejemplos*: discos duros, pendrives, tarjetas SD...).

- El programa se **carga** de la memoria secundaria a la memoria principal.

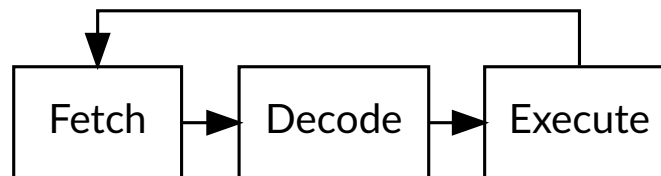


Esquema básico de un ordenador

- Una vez allí, la CPU va **extrayendo** las instrucciones que forman el programa y las va **ejecutando** paso a paso, en un bucle continuo que se denomina **ciclo de instrucción**.
- Durante la ejecución del programa, la CPU recogerá los datos de entrada desde los dispositivos de entrada y los almacenará en la memoria principal, para que las instrucciones puedan operar con ellos.
- Al finalizar el programa, los datos de salida se volcarán hacia los dispositivos de salida.

#### 1.2.2.5. Ciclo de instrucción

- En la **arquitectura Von Neumann**, los programas se almacenan en la memoria principal junto con los datos (por eso también se denomina «arquitectura de **programa almacenado**»).
- Una vez que el programa está cargado en memoria, la CPU repite siempre los mismos pasos:
  1. **(Fetch)** Busca la siguiente instrucción en la memoria principal.
  2. **(Decode)** Decodifica la instrucción (identifica qué instrucción es y se prepara para su ejecución).
  3. **(Execute)** Ejecuta la instrucción (envía las señales de control necesarias a las distintas unidades funcionales).



Ciclo de instrucción

#### 1.2.2.6. Representación de información

##### 1.2.2.6.1. Codificación interna Sistema binario

#### 1.2.2.6.2. Codificación externa ASCII

Unicode

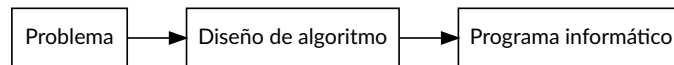
### 1.3. Algoritmo

#### 1.3.1. Definición

**Algoritmo:**

Un algoritmo es un método para resolver un problema.

- Está formado por una secuencia de pasos o **instrucciones** que se deben seguir (o **ejecutar**) para resolver el problema.
- La palabra «algoritmo» proviene de **Mohammed Al-Khowârizmi**, matemático persa que vivió durante el siglo IX y reconocido por definir una serie de reglas paso a paso para sumar, restar, multiplicar y dividir números decimales.
- **Euclides**, el gran matemático griego (del siglo IV a. C.) que inventó un método para encontrar el máximo común divisor de dos números, se considera con Al-Khowârizmi el otro gran padre de la Algorítmica (la ciencia que estudia los algoritmos).
- El estudio de los algoritmos es importante porque la resolución de un problema exige el diseño de un algoritmo que lo resuelva.



Resolución de un problema

#### 1.3.2. Características

- Un algoritmo debe ser:
  - **Preciso**: debe indicar el orden de ejecución de cada paso.
  - **Definido**: si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
  - **Finito**: debe terminar en algún momento, es decir, debe tener un número finito de pasos.

#### 1.3.3. Representación

##### 1.3.3.1. Ordinograma

#### **1.3.3.2. Pseudocódigo**

**1.3.4. Cualidades deseables**

**1.3.5. Computabilidad**

**1.3.6. Corrección**

**1.3.7. Complejidad**

**1.4. Programa**

**1.5. Lenguaje de programación**

## **2. Evolución histórica**

**2.1. Culturas de la programación**

**2.2. Ingeniería del software**

## **3. Resolución de problemas mediante programación**

**3.1. Análisis del problema**

**3.2. Especificación**

**3.3. Diseño del algoritmo**

**3.4. Codificación del algoritmo en forma de programa**

## **4. Paradigmas de programación**

**4.1. Imperativo**

**4.1.1. Estructurado**

**4.1.2. Orientado a objetos**

**4.2. Declarativo**

**4.2.1. Funcional**

**4.2.2. Lógico**

## **5. Lenguajes de programación** 8

**5.1. Definición**

**5.1.1. Sintaxis**

**5.1.1.1. Notación EBNF**



5.1.2. Semántica

5.2. Evolución histórica

5.3. Clasificación

5.3.1. Por nivel

5.3.2. Por generación

5.3.3. Por paradigma

6. Traductores

6.1. Compiladores

6.2. Intérpretes

6.2.1. Interactivos (*REPL*)

6.2.2. Por lotes

7. Entornos integrados de desarrollo

7.1. Terminal

7.1.1. [Zsh](#)

7.1.2. Oh My Zsh

7.1.3. [less](#)

7.2. Editores de texto

7.2.1. Editores vs. IDE

7.2.2. Vim y less

7.2.3. Visual Studio Code

7.2.3.1. Instalación

#### **7.2.3.2. Configuración**

#### **7.2.3.3. Extensiones**