

Tema 1. Introducción a la Programación

Objetivos

- Introducir los principales conceptos relacionados con la informática
- Definir los elementos de un sistema informático
- Conocer los componentes básicos del *hardware* de un ordenador
- Definir el *software* y clasificarlo
- Conocer cómo se representa la información para que sea manipulable por el ordenador
- Introducir la metodología para la elaboración del *software*

Índice

1. Conceptos Básicos
2. Elementos de un sistema informático
3. Soporte físico o *hardware*
4. Soporte lógico o *software*
5. Representación de la información
6. Metodología de la programación
7. Resumen

Conceptos básicos

- **Informática:** **INFORM**ación auto**MÁTICA**.

Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático y racional de la **información** por medio de **ordenadores**. Ciencia que abarca todos los aspectos del diseño y uso de ordenadores

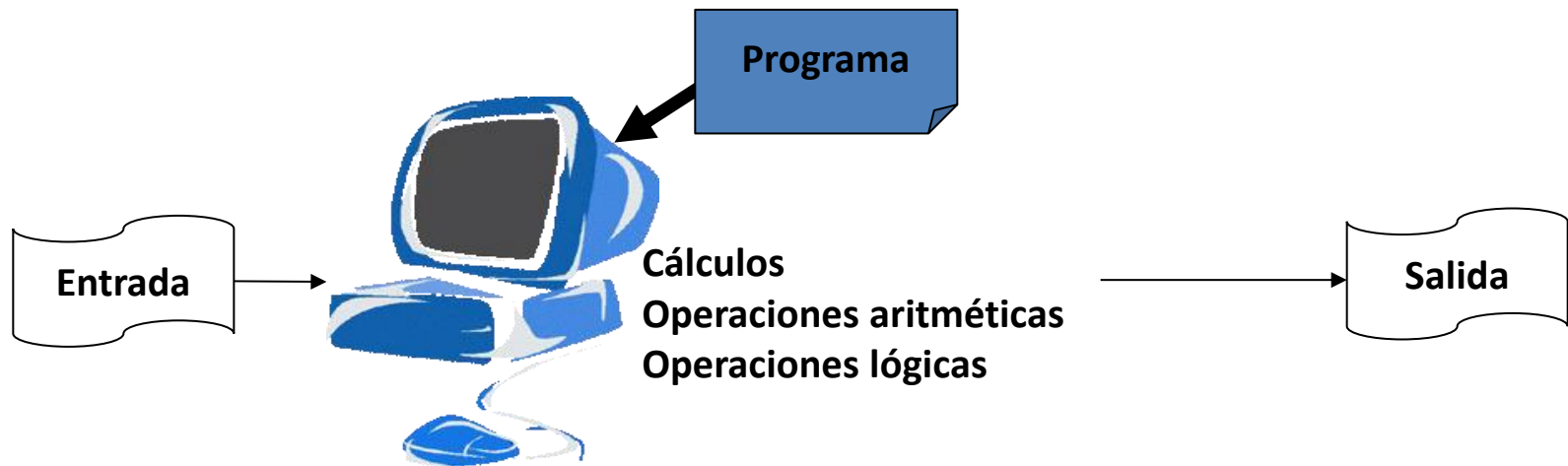
- **Información:** conjunto de símbolos que representan hechos, objetos o ideas

- **Tipos de información:**

- Datos: conjuntos de símbolos que representan una información en forma adecuada para ser procesada
- Instrucciones: indican al ordenador qué debe hacer y con qué datos (procesando los de entrada o produciendo los de salida)

Conceptos básicos

- **Ordenador:** máquina que acepta unos **datos de entrada**, realiza **operaciones aritméticas y lógicas** y obtiene **datos de salida**, bajo el control de un **programa** previamente almacenado



- Necesidad *Hardware + Software*

Conceptos básicos

- **Programa:** conjunto ordenado de **sentencias** que se dan a un ordenador indicando las operaciones que se desea que realice
- **Tipos de sentencias:**
 - Imperativas o **instrucciones**: representan una orden a ejecutar (transferencias de datos, tratamiento de datos, bifurcación y saltos)
 - **Declarativas**: proporcionan información sobre los datos

Conceptos básicos

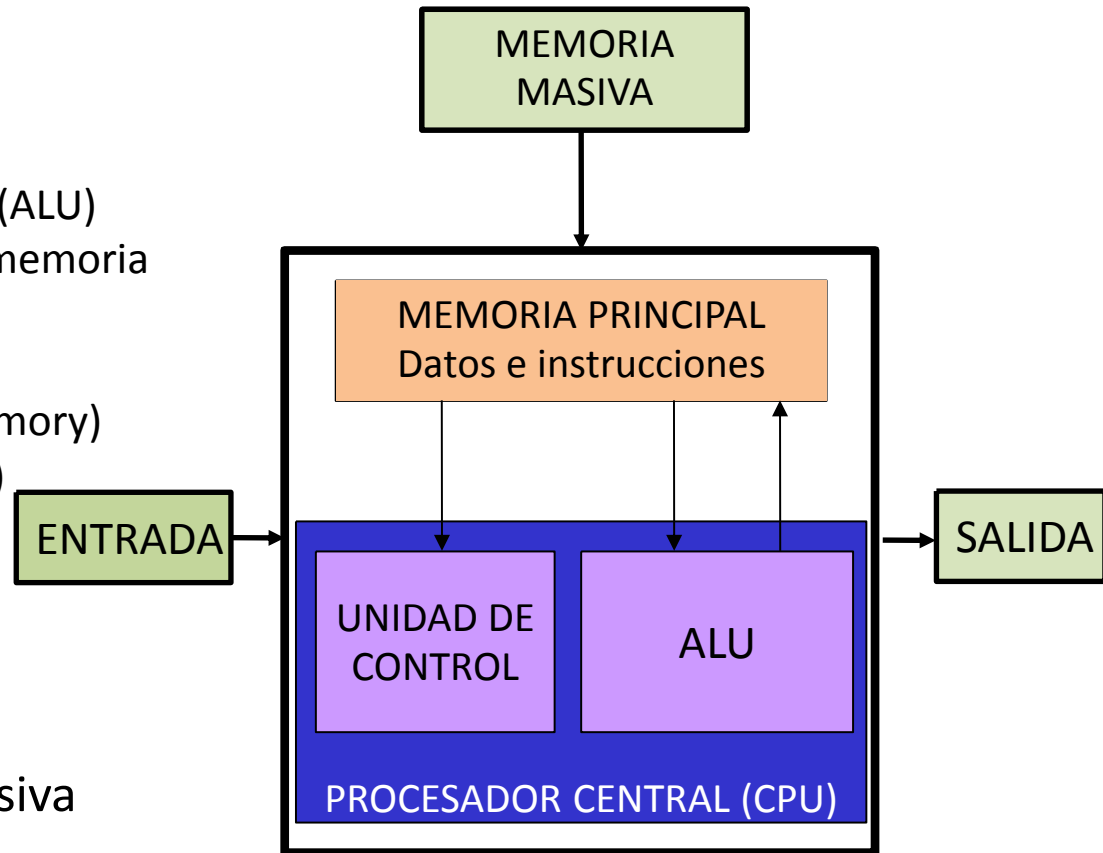
- **Aplicación informática:** Conjunto de programas y su documentación, para la realización de un determinado tipo de trabajo
- **Sistema informático:** Conjunto de elementos necesarios para la realización y explotación de aplicaciones informáticas

Elementos de un sistema informático

- **Hardware o soporte físico:** conjunto de componentes físicos
- **Software o soporte lógico:** conjunto de programas ejecutables sobre el hardware junto con los documentos y datos asociados a los mismos
- **Personal informático:** Conjunto de personas relacionadas con la utilización y explotación de los ordenadores

Soporte físico o *hardware*

- Esquema Von Neumann (1945)
- Computadora Central
 - CPU (Procesador)
 - Unidad de Control (UC)
 - Unidad Aritmético-Lógica (ALU)
 - En la actualidad también memoria caché, coprocesador
 - Memoria principal:
 - RAM (Random Access Memory)
 - ROM (Read Only Memory)
 - Caché (en la actualidad)
- Periféricos
 - Unidades de entrada
 - Unidades de salida
 - Unidades de memoria masiva

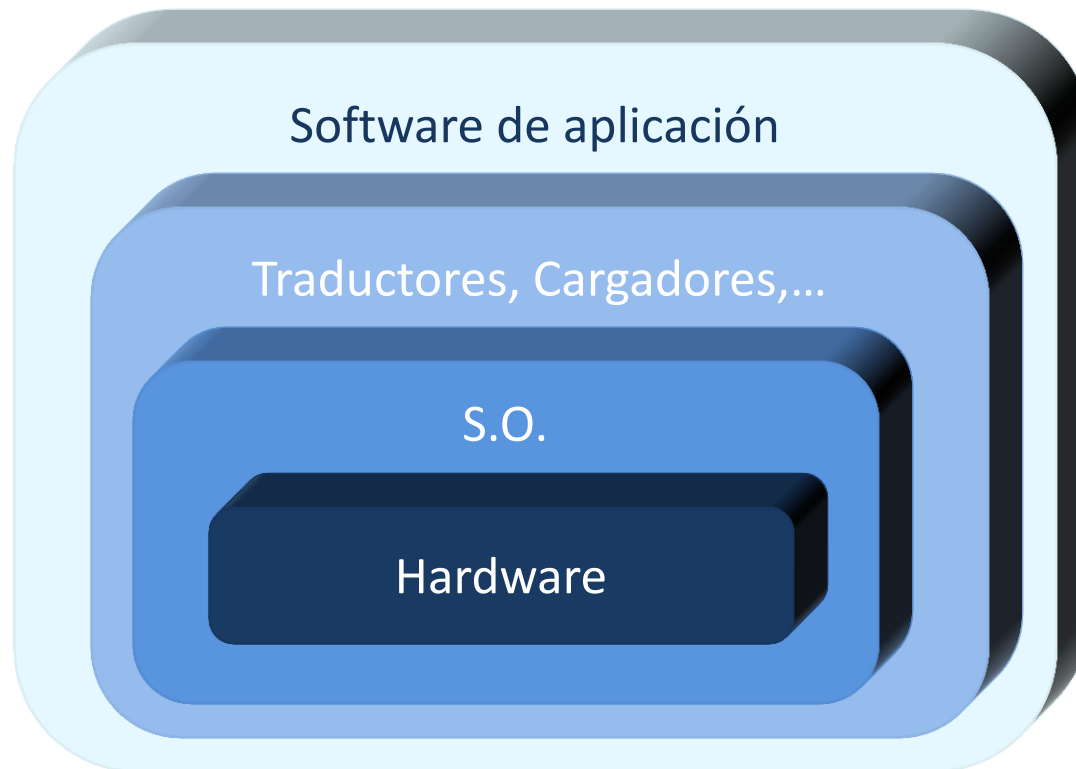


Soporte lógico o *software*

- Conjunto de programas que se ejecutan en un ordenador y su documentación asociada
- **Sistema operativo:** conjunto de programas que **controla** directamente los **recursos** de un ordenador proporcionando una **máquina virtual** más fácil de utilizar que el hardware subyacente

Soporte lógico o *software*

- Capas de *software*



Soporte lógico o *software*

- **Lenguaje de programación:** lenguaje utilizado para la escritura de los programas
- Elementos del lenguaje:
 - Léxico o vocabulario: conjunto de símbolos constitutivos (palabras)
 - Sintaxis o gramática: conjunto de reglas para la construcción de las sentencias correctas
 - Semántica: conjunto de reglas que asocian significado a las construcciones correctas del lenguaje

Soporte lógico o *software*

- Lenguaje o código máquina:
 - Directamente interpretable por la UC
 - Símbolos {0, 1}, complejo
 - No permite uso comentarios } Programas poco legibles
- Repertorio de instrucciones muy limitado y formato muy rígido
- Muy dependiente del *hardware* (específico para cada procesador) (*programas poco portables*)
- Necesidad por parte del programador de conocer direcciones de memoria y resto de unidades físicas

Soporte lógico o *software*

- Lenguajes ensambladores y macroensambladores:
 - Notación simbólica para representar la operación
 - Uso de direcciones simbólicas de memoria
 - Se pueden incluir comentarios
 - En macroensambladores existen macros o módulos que permiten ampliar el repertorio de instrucciones
 - Los programas resultantes necesitan traducción

Soporte lógico o *software*

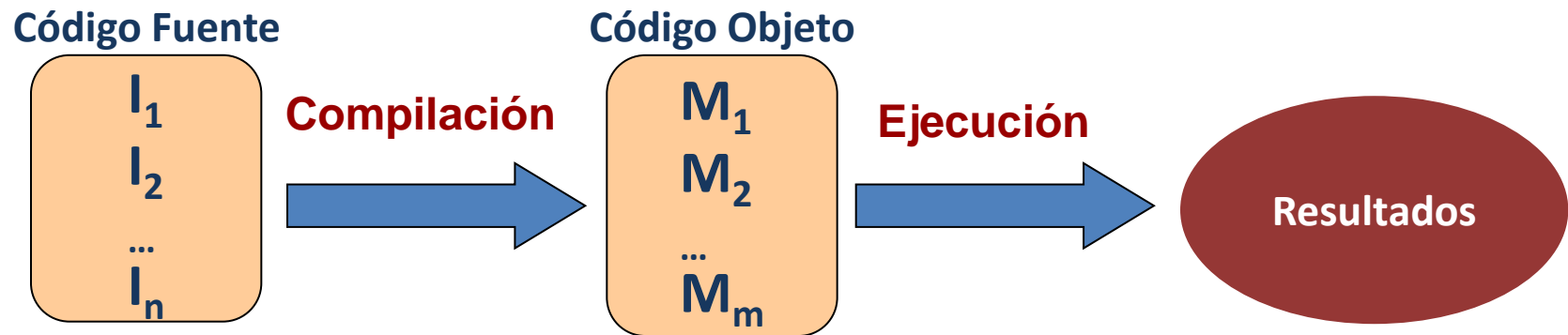
- Lenguajes de alto nivel
 - Vocabulario amplio (caracteres alfabéticos, numéricos y especiales)
 - Permiten uso de variables
 - Repertorio de instrucciones amplio
 - Casi independientes del *hardware*
 - Permite uso de comentarios (se aumenta la legibilidad)
 - Los programas resultantes necesitan traducción

Soporte lógico o *software*

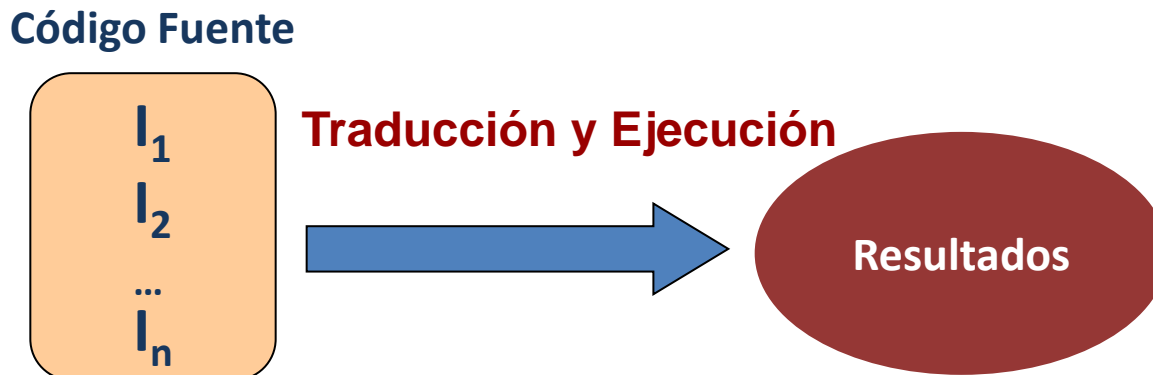
- **Traductor:** Software que traduce programas escritos en un lenguaje de programación a otro
- Tipos:
 - **Compiladores:** traducen globalmente un programa escrito en un determinado lenguaje (**fuelle**) y producen como salida el programa escrito en otro lenguaje (**objeto**)
 - **Intérpretes:** traducen y ejecutan una a una las sentencias de un programa pero sin generar un nuevo programa traducido (no generan código o programa objeto)

Soporte lógico o *software*

- Compilador



- Intérprete



Representación de la información

- Codificación en sistema binario $\{0,1\}$
- **Bit**: unidad mínima de información. Representa la información correspondiente a la ocurrencia de un suceso de entre dos posibilidades distintas
- **Byte**: cantidad mínima de información para representar un carácter. Se suele tomar **8 bits**
- Todos los datos se convierten a su representación binaria

Representación de la información

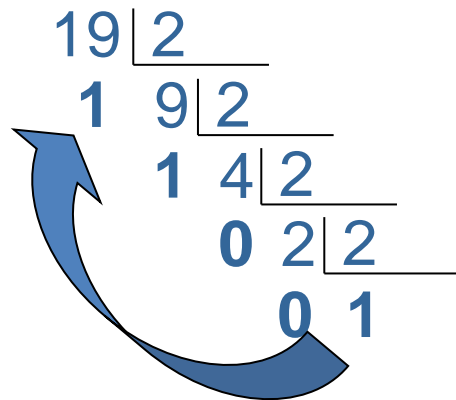
- Múltiplos

	Bytes o bits	
Kilo	10^3	
Mega	10^6	10^3 Kilo
Giga	10^9	10^3 Mega
Tera	10^{12}	10^3 Giga
Peta	10^{15}	10^3 Tera
Exa	10^{18}	10^3 Peta

Representación de la información

- Código binario:

$$19 = 10011$$



$$11010 \overset{\text{dígito}_0}{\leftarrow} = 26$$

$$0*2^0 + 1*2^1 + 0*2^2 + 1*2^3 + 1*2^4$$

$$\sum_{i=0}^{n^{\circ} \text{digitos}-1} \text{dígito}_i * 2^i$$

- n bits permiten representar 2^n datos diferentes
- Todos los datos se convierten a información numérica, luego a binario y se almacenan

Metodología de la programación

- El proceso de la programación es un proceso de resolución de problemas
- Definición (¿Qué?)
 - Análisis del problema
- Desarrollo (¿Cómo?)
 - Diseño del algoritmo
 - Codificación a un lenguaje de programación
 - Pruebas: depuración, verificación y validación
- Mantenimiento (Mejoras y correcciones)
 - Mantenimiento

Ingeniería del
Software

Definición del problema

- Tener claro las Entradas, Salidas e intentar buscar Relación entre ellas
- Una buena definición es esencial para obtener una solución correcta
- Ejemplo Problema: Diseñar un programa que calcule el área de un rectángulo

// Programa AreaRectangulo

// Entradas: la base y la altura del rectángulo

// Salidas: el área del rectángulo

Relación: el área es el producto de la base por la altura

Desarrollo

- Diseño: desarrollo de un **algoritmo** o conjunto finito de acciones para resolver un problema (fundamental en problemas complejos)
- Codificación: expresar las acciones del algoritmo con un lenguaje de programación y así obtenemos un programa (vamos a utilizar C++)
- Pruebas: ejecutar el programa dándole valor a los datos de entrada para comprobar si devuelve bien la salida (ver si resuelve el problema)

Mantenimiento

- Necesidad de realizar correcciones a errores aparecidos cuando el software se está utilizando
- Necesidad de ampliar las funcionalidades del software
- Necesidad de adaptación del software a nuevos entornos

Características de los programas

- **Legibilidad:** programa fácil de leer y entender
- **Portabilidad:**
 - Algoritmo: fácil codificación en cualquier lenguaje
 - Programa: ejecución en cualquier sistema
- **Modificabilidad:** facilidad para hacer correcciones y mejoras.
- **Eficiencia:** aprovechamiento de los recursos del ordenador
- **Modularidad:** división en módulos con funciones bien definidas
- **Estructuración:** se ha de cumplir con las reglas de la Programación Estructurada

Estilo de programación

- Objetivo: hacer el algoritmo/programa más legible
- Elementos de estilo:
 - Uso de comentarios
 - Código autodocumentado
 - Código con formato
 - Uso de constantes

Resumen

- Conceptos básicos
- *Hardware + Software*
- Traductores: Compiladores e intérpretes



- Proceso de programación:
 - Definición o Análisis del problema
 - Diseño (Algoritmo)
 - Codificación (Programa)
 - Pruebas