

TEMA 1. MÁQUINAS QUE EJECUTAN ALGORITMOS

1.1 El ordenador: una máquina que procesa algoritmos.

1.1.1 ¿Qué tareas puede realizar un ordenador?

1.2 Noción de proceso computacional. Estado de un proceso

1.3 Estructura de un ordenador: naturaleza digital

1.4 Definición de información. Codificación

1.5 Un ordenador como una jerarquía de máquinas virtuales

1.6 Lenguajes de programación. Traductores

1.7 La creación de software

1.8 La informática y el papel de la programación de ordenadores

El ordenador, una máquina que procesa algoritmos.

- Era digital.
- Sociedad de la información
 - Internet (servicio Web)
 - Nuevas formas de comunicación
 - Nuevos modelos de negocios
- Un ordenador es un ejemplo de mecanismo

MECANISMO	INDICADORES
Cisterna	Nivel del agua y abertura
Reproductor de CD	Pista que se estás reproduciendo y posición de ella
Reloj digital	Hora, minutos y segundos
Ordenador digital	Variables o elementos de memoria

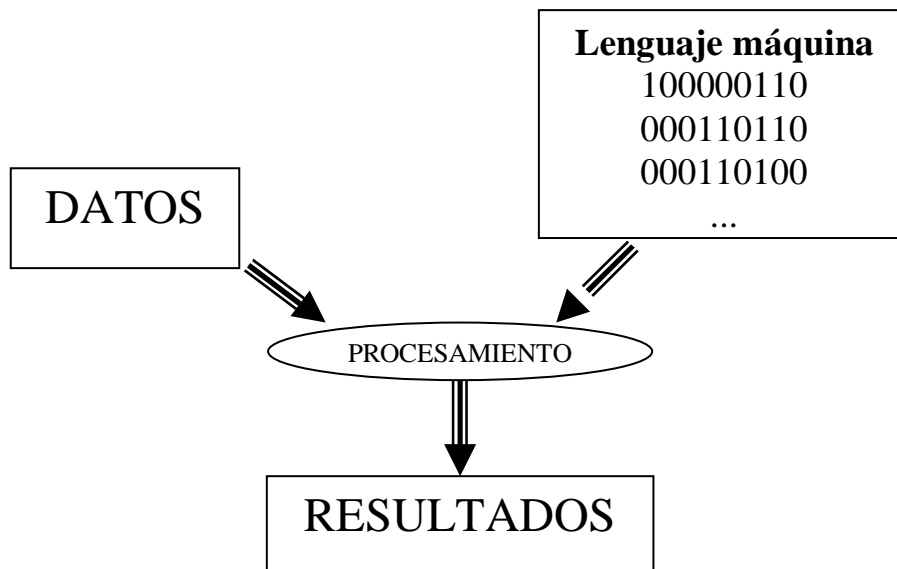
- Definición de ordenador
 - Mecanismo digital de propósito general que se convierte en un mecanismo para uso específico cuando procesa un determinado algoritmo.
 - Nóminas de una empresa
 - Control de temperatura de una habitación
- **Software frente hardware**

El ordenador, una máquina que procesa algoritmos.

Máquina de propósito general + **operaciones** =
máquina de propósito particular

Algoritmo = Secuencia de instrucciones
(operaciones) que especifican las acciones que
deber ejecutar un computador para realizar una
tarea

¿Cómo se conoce el grado de cumplimiento de
la tarea? A través de su estado (**variables**)



El ordenador, una máquina que procesa algoritmos.

El computador ejecuta operaciones muy simples (**instrucciones máquina**)

Lenguajes de programación. Notaciones que expresan las operaciones que pretendemos que ejecute el ordenador, escritas con un nivel de abstracción superior al del lenguaje máquina

Necesidad de un **traductor**

Es habitual utilizar en el análisis del problema una **notación algorítmica**, con el fin de evitar los detalles propios de un lenguaje de programación, y centrarse más en los aspectos que describen el proceso que debe llevar a cabo el computador

Programa versus algoritmo

PROPIEDADES DE UN ALGORITMO

Finitud: La longitud y duración de un algoritmo son finitos.

Definitud: Un algoritmo determina sin ningún tipo de ambigüedad las operaciones que debe ejecutar un ordenador. Estas operaciones se deben de interpretar de una única forma.

Efectividad. Las reglas pueden ser ejecutadas por un ser humano con papel y lápiz.

Generalidad. Normalmente, un algoritmo resuelve una clase de problemas y no un problema en particular.

PROPIEDADES DE UN ALGORITMO

Entradas y salidas. Un algoritmo puede tener varias entradas o ninguna, sin embargo, al menos debe tener una salida, el resultado que se pretende obtener.

Eficiencia. La ejecución de un programa consume una serie de recursos: tiempo y memoria. Estos recursos son limitados y por tanto ante dos algoritmos que solucionen el mismo problema, será más eficaz el que utilice el menor número de recursos.

Determinismo. Un algoritmo se comporta de la misma forma para los mismos datos de entrada.

LA NOCIÓN DE ALGORITMO ES MÁS GENERAL

Algoritmo	Problema
<ul style="list-style-type: none"> • Conjunto de instrucciones para tejer un jersey • Partitura • Un plan de trabajo para cursar una licenciatura • Receta de cocina • Método de Newton 	<ul style="list-style-type: none"> • Tejer un jersey • Tocar una sonata • Cursar una licenciatura • Cocinar un bizcocho • Calcular las raíces de una función

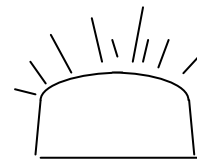
Ingredientes:

- Huevos
- Harina
- leche

...

Receta para bizcocho

Mezclar la levadura con la harina, batir los huevos, cocinar durante media hora.....



Datos de
entrada

Algoritmo

Ordenador

Datos de
salida

¿DETERMINISTA?

¿DEFINIDO SIN AMBIGÜEDADES?

¿QUÉ TAREAS PUEDE REALIZAR UN ORDENADOR?

Problemas del mundo real:

Computables

Tratables

Intratables

No computables

Un problema es **computable** cuando existe un algoritmo que lo resuelve

Un problema computable es **tratable** si existen un algoritmo eficiente (orden polinomial) que lo resuelve

Problemas **NP-completos** un caso especial de algoritmos intratables

Existe un algoritmo eficiente para un problema NP-completo si y sólo existen algoritmos eficientes para todos los problemas NP-completos

Noción de proceso computacional.

Estado de un proceso

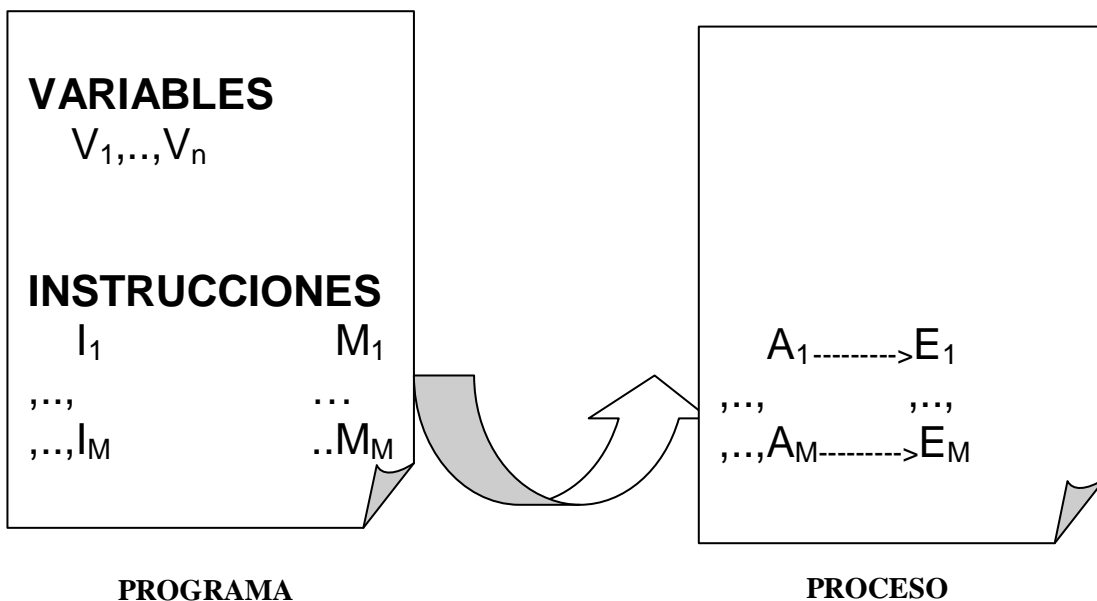


LOCAL 3

VISITANTE 1

PROCESO

VARIABLES



**Cambio estado
en las variables**

Noción de proceso computacional.

Estado de un proceso

Definición de Proceso

Un proceso es un algoritmo en ejecución y es caracterizado por una sucesión de estados.

Definición de *Estado de un proceso*

Determina el grado de progreso de un proceso, viene dado por los valores de las variables y un indicador de la siguiente instrucción a ejecutar.

Definición de *Variable*

Magnitud o información manejada por un algoritmo a la que se da un nombre y registra un valor.

Representa una entidad del mundo real cuyo valor es relevante en el problema a resolver.

Noción de proceso computacional.

Estado de un proceso

Problema: Dado un entero $n \geq 0$ calcular el factorial de n ($n!$) definido como

$$0! = 1$$

$$n! = n \times (n - 1) \times (n - 2) \times \dots \times 1$$

LÉXICO

n es un entero ≥ 0

i es un entero ≥ 1

f es un entero ≥ 1

ALGORITMO

1. **Introducir** n

2. **Asignar** el valor 1 a f

3. **Asignar** el valor 1 a i

4. **Si** $i > n$ **entonces** mostrar f en pantalla y fin de la ejecución

5. **Asignar** a f el producto de i por el valor anterior de f

6. **Sumar** 1 a i

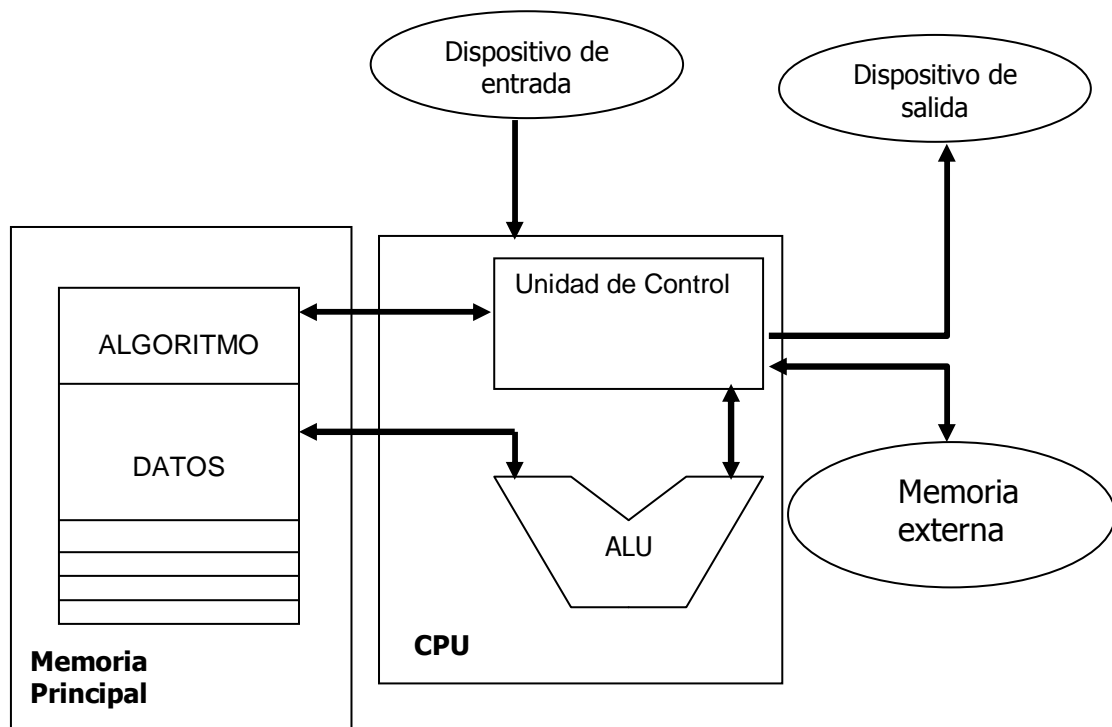
7. **Ir al paso** 4

Noción de proceso computacional.

Estado de un proceso

Paso	n	f	i	Siguiente Instrucción
1	3	?	?	2
2	3	1	?	3
3	3	1	1	4
4	3	1	1	5
5	3	1	1	6
6	3	1	2	7
7	3	1	2	4
4	3	1	2	5
5	3	2	2	6
6	3	2	3	7
7	3	2	3	4
4	3	2	3	5
5	3	6	3	6
6	3	6	4	7
7	3	6	4	4
4	3	6	4	Fin ejecución

Estructura de un ordenador: naturaleza digital



UN COMPUTADOR ESTÁ CARACTERIZADO POR EL CONJUNTO DE INSTRUCCIONES QUE PUEDE EJECUTAR: **INSTRUCCIONES MÁQUINA**

FORMATO: CÓDIGO + [OPERANDOS] (entre 0 y 3 operandos)

CÓDIGO: INSTRUCCIÓN DE SUMA, RESTA, COMPARACIÓN, SALTO ...

OPERANDO: DIRECCIÓN DE MEMORIA, VALOR NUMÉRICO, ...

Definición de información. Codificación

UNIDAD DE MEDIDA DE LA INFORMACION: BIT

CANTIDAD DE INFORMACIÓN OBTENIDA AL CONOCER UNA DE DOS ALTERNATIVAS EQUIPROBABLES.

BYTES (8 BITS), KILOBYTES (2^{10} BYTES = 1024 BYTES),

MEGABYTES (2^{20} BYTES = 1048576 BYTES)

GIGABYTES(2^{30} BYTES), TERABYTES (2^{40} BYTES)

INFORMACIÓN NUMÉRICA

Decimal	Binario
0	0
1	1
5	101
7	111
11	1011
15	1111

INFORMACIÓN ALFANUMÉRICA

Carácter	ASCII Binario	EBCDIC Binario
A	01000001	11000001
B	01000010	11000010
C	01000011	11000011
D	01000100	11000100
E	01000101	11000101

INSTRUCCIONES

Instrucción	Codificación
MOVER	01000101
SUMAR	01010001
SALTAR	01011001

Un ordenador como una jerarquía de máquinas virtuales

USUARIO

Lenguaje de la aplicación

NIVEL APLICACIÓN

Microsoft Word

ENTORNO GPS GNAT
ADA, EMACS, ...

LENGUAJE MÁQUINA

Instrucciones con mayor expresividad e independientes del LM, portabilidad

NIVEL LENGUAJE DE PROGRAMACIÓN

Pascal
C
Java

- Es necesario conocer la **estructura y funcionamiento** del ordenador en particular.
- Naturaleza **binaria** de las instrucciones
- **Ausencia de generalidad.** Un programa se escribe para un procesador con un determinado juego de instrucciones.
- Los programas son muy **largos** debido a la simplicidad de las instrucciones.
- El programador debe idear el programa en términos del **flujo de ejecución** que sigue el procesador.

Nemónicos, formato decimal y hexadecimal para valores y direcciones
Macros, servicios del SO

NIVEL LENGUAJE ENSAMBLADOR

Llamadas al sistema, lenguaje de órdenes

NIVEL SISTEMA OPERATIVO

Linux
MS DOS
Windows
NT
...

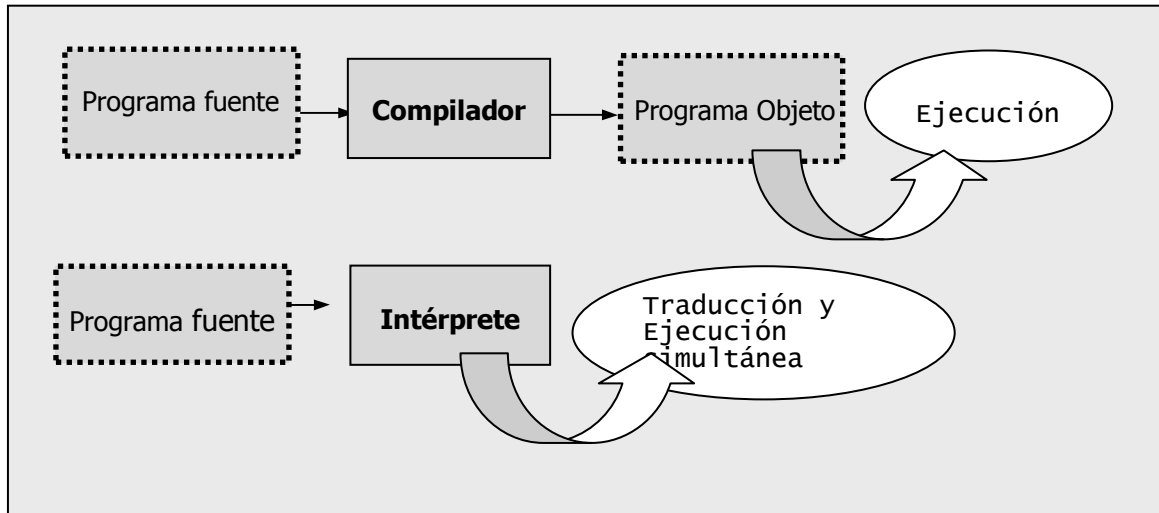
NIVEL MÁQUINA

HARDWARE

CADA NIVEL OFRECE UNOS SERVICIOS QUE EL USUARIO DE AQUEL NIVEL PUEDE UTILIZAR SIN NECESIDAD DE CONOCER EL RESTO DE NIVELES

Lenguajes de programación.

Traductores



Etapas en la compilación:

- Análisis léxico
- Análisis sintáctico
- Análisis semántico
- Generación de código intermedio
- Generación de código máquina
- Optimización del código.

Lenguajes de programación.

Traductores

Paradigma de programación

Colección de conceptos que guían el proceso de construcción de un programa, determinando su estructura. Dichos conceptos controlan la forma en que se piensan y formulan los programas.

Tres tipos de paradigmas:

Imperativo o procedural

Estructurados: Algol, Pascal, Ada

No Estructurados: Fortran, Basic, Cobol, C

Declarativo

Relacional: Prolog, Parlog

Funcional: Lisp, Hope, Miranda

Orientado a objetos:

Smalltalk, Eiffel, C++, Java

La creación de software

Tamaño de los problemas

En los inicios de la programación programar era dominar un lenguaje de programación

Los fracasos en diferentes proyectos dio lugar a la “crisis del software”. 1968 (Conferencia de la OTAN en Garmisch). Nació la **Ingeniería del Software**.

“Disciplina que se ocupa de la aplicación del conocimiento científico al diseño y construcción de programas de ordenador, y a las actividades asociadas de documentación, operación y mantenimiento, lo que proporciona un enfoque sistemático”

Aunque se ha mejorado, continúa la crisis.

La creación de software

Frederick P. Brooks en su artículo “*No Silver Bullet -- Essence and Accident in Software Engineering*”, 1987.

Crear software siempre sería una actividad difícil, ya que existirían problemas esenciales que ninguna técnica podría hacerlos desaparecer: los sistemas software son sistemas muy complejos y sujetos a continuos cambios, y los requisitos son difíciles de capturar

La reutilización, las técnicas para mejorar la captura de los requisitos y disponer de buenos programadores son las mejores armas para atacar la dificultad inherente a la construcción de software.

- ANÁLISIS DEL PROBLEMA
- DISEÑO DE LA SOLUCIÓN
- CONSTRUCCIÓN DEL PROGRAMA
- VALIDACIÓN
- DOCUMENTACIÓN
- MANTENIMIENTO

La informática y el papel de la programación de ordenadores

Informática versos TIC (Tecnologías de la Información y las Comunicaciones)

Sinergia entre informática, comunicaciones y electrónica

Definición de Informática (1989. ACM/IEEE)

“disciplina del estudio sistematizado de los procesos algorítmicos que describen y transforman información, su teoría, análisis, diseño, eficiencia, implementación, y aplicación. La cuestión fundamental es ¿qué puede ser automatizado de un modo eficiente?”

La informática y el papel de la programación de ordenadores

Informática es simultáneamente una disciplina matemática, científica y una ingeniería.

Tres formas de pensar propias de la:

Teoría: usando, desarrollando y entendiendo los principios matemáticos que se aplican

Matemática discreta

Computabilidad

Teoría de la complejidad

Gramáticas formales y autómatas

Con el pensamiento teórico se describen y prueban relaciones entre objetos.

Abstracción: esta forma de pensar tiene su raíz en las ciencias experimentales y el método científico.

Recolección de datos y formación de hipótesis,

Definición de un modelo

Predicción

Diseño

Experimentación

Análisis de resultados

La informática y el papel de la programación de ordenadores

Diseño: tiene que ver con el aspecto de ingeniería de la informática. Se construyen artefactos (aplicaciones, ordenadores, sistemas operativos, traductores, etc.) de un modo sistemático, aplicando técnicas bien definidas.

Requisitos

Especificaciones

Se diseñan y analizan sistemas o mecanismos para resolver problemas y se prueban.

Marco curricular ACM e IEEE (*Computing Curricula* 2001). Se identifican catorce áreas dentro de la informática

Estructuras Discretas

Fundamentos de Programación

Algoritmos y Complejidad

Lenguajes de Programación

Arquitectura y Organización

Sistemas Operativos

Computación Centrada en la Red

Ingeniería del Software

Gestión de la Información

Sistemas Inteligentes

Interacción Hombre-Máquina

Informática Gráfica

Computación Científica

Cuestiones Sociales y Profesionales.