



## UNIDAD TEMÁTICA I

# Conceptos Básicos



# Dato



- ✦ La palabra proviene del latín datum, forma del verbo dare “dar”, que significa “Lo que es dado”
- ✦ Hecho numéricos reunidos para referencia o información
- ✦ Representación de una información de manera adecuada para su tratamiento por un ordenador



# Dato



✦ Son representaciones simbólicas de objetos, hechos, instituciones, conocimientos.

## Ejemplos

- ✦ **15**
- ✦ **25° C**
- ✦ **25 m**
- ✦ **31-10-2005**
- ✦ **Febrero**
- ✦ **María Pérez**



# Información



- ✦ Los datos organizados o procesados adquieren significado y proporcionan conocimiento o desencadenan un comportamiento sobre los hechos u objetos que lo originaron
- ✦ La información se resume en :  
Datos + significado



# Información



## Un grupo de datos es información

- **Datos: 19- Marzo – Maria – 20 ptos.**
- **Información: El 19 de marzo se publicó en acta la calificación de 20 ptos para María**
- **El principal objetivo de la información consiste en aumentar el conocimiento o reducir la incertidumbre.**



# Información



## Ejemplos

- ✦ Juan tiene 15 años
- ✦ 25° C fue la temperatura de esta mañana.
- ✦ El edificio "Robles" mide 25 m de altura
- ✦ La fecha de hoy es 31-10-2005
- ✦ Febrero tiene 28 días y 29 si el año es bisiesto.
- ✦ María Pérez es un nombre muy común



# Problema

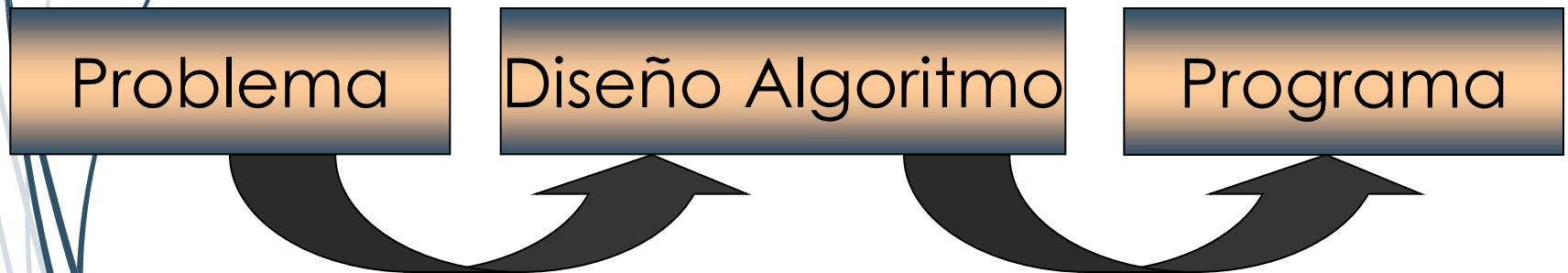


- ✦ Es un asunto o un conjunto de cuestiones que se plantean para ser resueltas.
- ✦ La naturaleza de los problemas varía con el ámbito o el contexto: problemas matemáticos, químicos, filosóficos, etc.
- ✦ Es importante que al abordar un problema se tenga una descripción simple y precisa del mismo, de lo contrario resultaría complejo modular, simular, o programar su solución en un computador.

# Problema



✦ Un programador de computadoras es ante que nada una **persona** que resuelve problemas, y para llegar a ser un programador eficaz se necesita aprender a resolver problemas de un modo riguroso y sistemático





# Problema



➤ Algunos de los pasos para solucionar un problema son:

- **Diseño del algoritmo, que describe la secuencia ordenada de pasos que conduce a la solución de un problema dado. (Análisis del problema y desarrollo del algoritmo)**
- **Expresar el algoritmo como un programa en un lenguaje de programación adecuado (Fase de Codificación)**
- **Ejecución y validación del programa por el computador**



# Algoritmo



- ✦ **Etimología:** Proviene del nombre del matemático persa Mohammed Al-khowanzmi, cuyo apellido traducido al latín es Algorismus
- ✦ **Descripción de un esquema de comportamiento con la ayuda de un repertorio finito, limitado y bien comprendido de acciones elementales**



# Algoritmo



- ✦ **Conjunto de acciones que especifican la secuencia de operaciones realizar, en orden, para resolver un problema**
- ✦ **Los algoritmos son independientes tanto del lenguaje de programación como del computador que los ejecuta.**



# Algoritmo



- **Finito:** un algoritmo siempre debe terminar después de un número finito de pasos
- **Definido:** si se sigue un algoritmo dos veces se debe obtener el mismo resultado
- **Entrada:** un algoritmo tiene cero o más entradas.
- **Salida:** un algoritmo tiene una o más salidas.
- **Efectivo:** resolver el problema para el cual fue diseñado

# Algoritmo



✦ **Ejemplo tradicional de un algoritmo: Cambiar el caucho de un Carro**





# Algoritmo



➤ **Ejemplo tradicional de un algoritmo: Cambiar el caucho de un Carro**

- 1. Buscar herramientas, caucho de repuesto y triángulo**
- 2. Ubicar el triángulo en el lugar adecuado**
- 3. Ir al lugar del caucho averiado**
- 4. Sacar las tuercas**
- 5. Colocar el gato**
- 6. Levantar el carro**



# Algoritmo



✦ **Ejemplo tradicional de un algoritmo: Cambiar el caucho de un Carro**

**7. Sacar el caucho**

**8. Colocar el caucho de Repuesto**

**9. Colocar las tuercas**

**10. Apretar las tuercas**

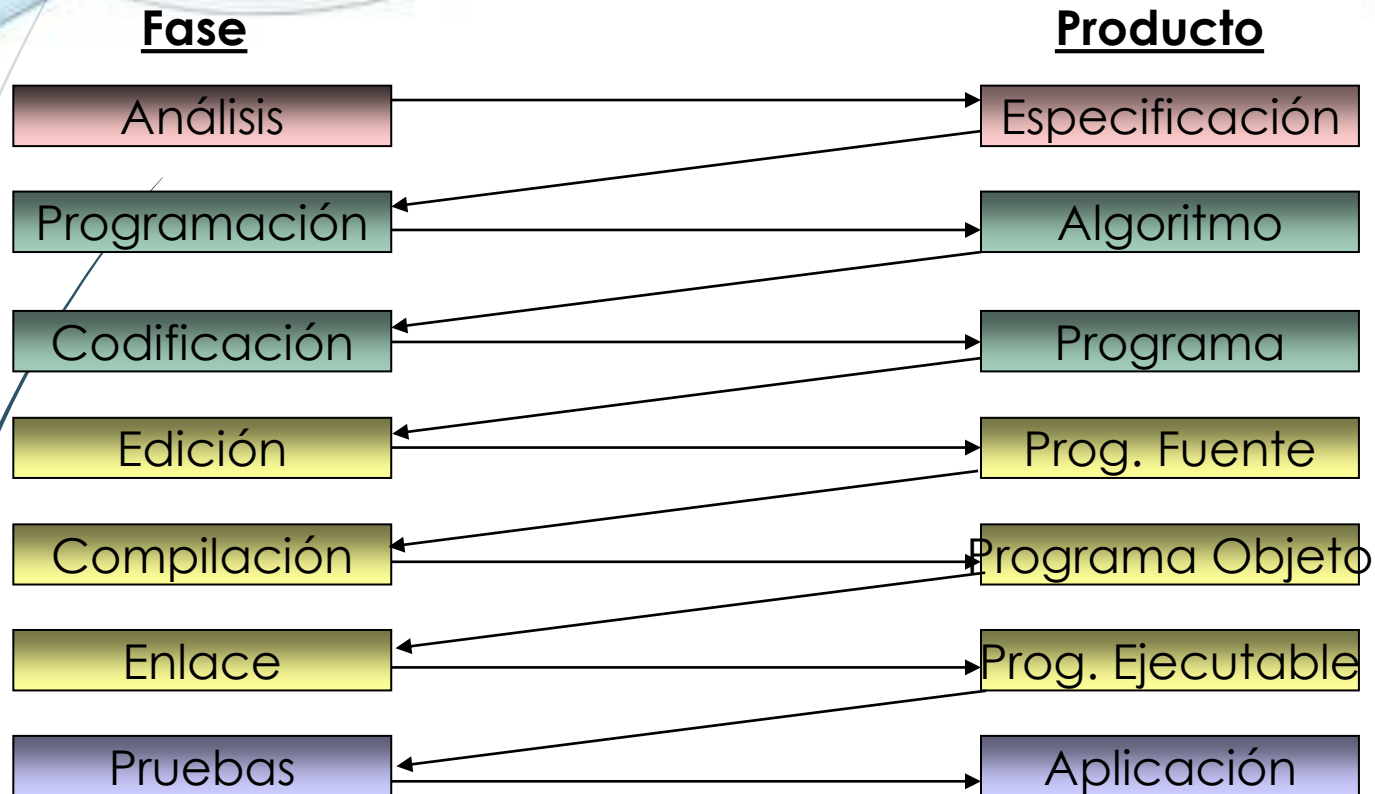
**11. Guardar las herramientas**

**FIN**



# Algoritmo

## Fases de Desarrollo de un Algoritmo

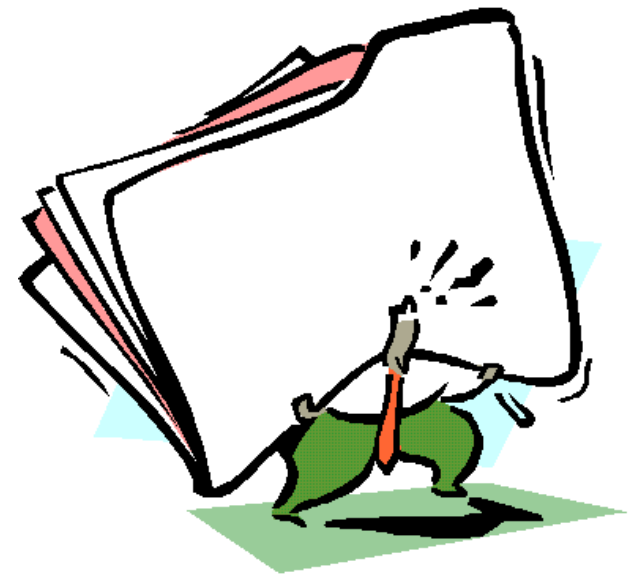


# Algoritmo

➤ **Fase de Análisis:** consiste en el estudio detallado del problema con el fin de obtener una serie de documentos (especificación) en los cuales quedan totalmente definido el proceso a seguir en la automatización



Estudio detallado

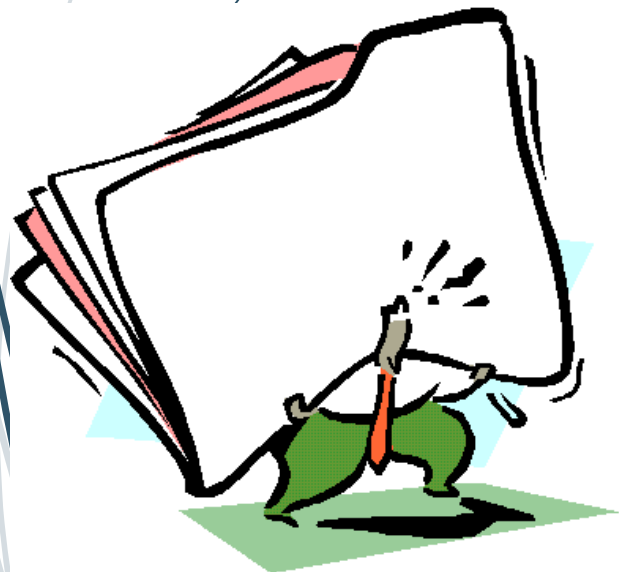


Documentos de Especificación



# Algoritmo

➤ **Programación:** consiste en la realización del algoritmo que resuelve el problema de acuerdo a la especificación dada en la fase anterior. El algoritmo se representa en una notación pseudoformal que también se conoce como pseudocódigo.



Documentos de  
Especificación



**Algoritmo PRUEBA**

**Inicio**

**Variables**

A,B,C: entero

Leer(A,B)

$C \leftarrow A+B$

Escribir (C)

**Fin**



# Algoritmo

➤ **Codificación:** consiste en la traducción del algoritmo a un programa escrito en un lenguaje de programación

**Algoritmo** Sumar

**Inicio**

**Variables**

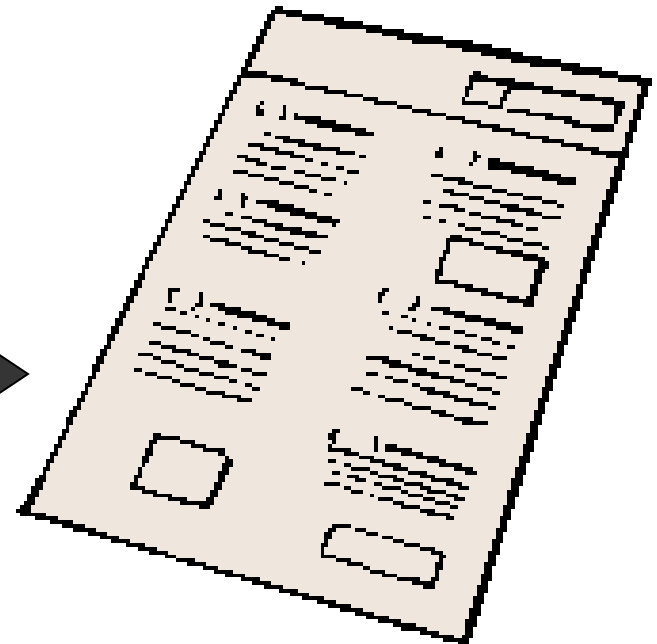
A,B,C: entero

Leer(A,B)

$C \leftarrow A+B$

Escribir (C)

**Fin**

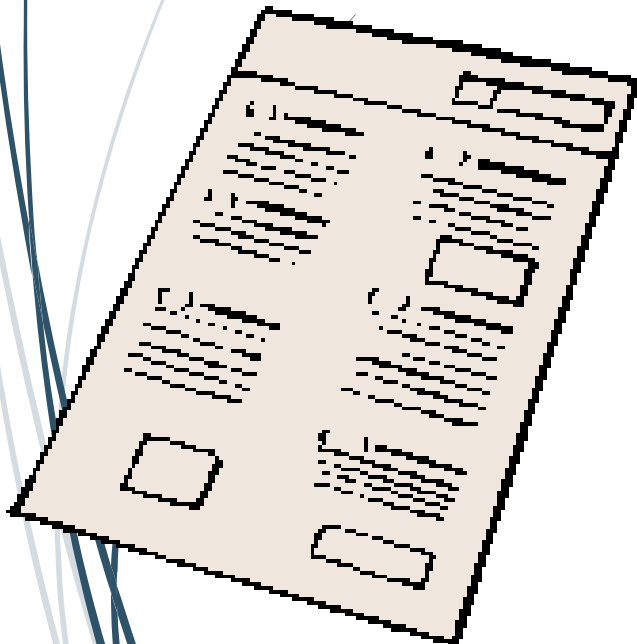


**Programa escrito  
en un Lenguaje de Programacion**

# Algoritmo



★ **Edición:** consiste en la transcripción del programa al computador, generalmente por medio de un editor de programas o procesador de textos, obteniendo un programa fuente



Programa escrito

en un Lenguaje de Programacion



```
4
5
6  uses
7    Windows, Messages, SysUtils, Variants, Classes, Graphics,
8    Dialogs, System.ComponentModel, Borland.Vcl.StdCtrls;
9
10 type
11   TForm1 = class(TForm)
12     Button1: TButton;
13     procedure Button1Click(Sender: TObject);
14   private
15     { Private declarations }
16   public
17     { Public declarations }
18   end;
19
20 var
21   Form1: TForm1;
22
23 implementation
24   {$R *.res}
25
26 procedure TForm1.Button1Click(Sender: TObject);
27 begin
28   ShowMessage('Merhaba delphiturkiye.com Üyeleri');
29 end;
30
31 end.
```

Programa fuente

# Algoritmo



➤ **Compilación:** consiste en obtener el programa objeto a partir del programa fuente, mediante el traductor de lenguaje, el cual además de efectuar la traducción, incluye un análisis sintáctico.

```
4
5  uses
6    Windows, Messages, SysUtils, Variants, Classes, Graphics,
7    Dialogs, System.ComponentModel, Borland.Vcl.StdCtrls;
8
9  type
10   TForm1 = class(TForm)
11     Button1: TButton;
12     procedure Button1Click(Sender: TObject);
13   private
14     { Private declarations }
15   public
16     { Public declarations }
17   end;
18
19  var
20     Form1: TForm1;
21
22  implementation
23
24     {$R *.nfm}
25
26   procedure TForm1.Button1Click(Sender: TObject);
27   begin
28     ShowMessage('Merhaba delphiturkiye.com Üyeleri');
29   end;
30
31  end.
```

Programa fuente



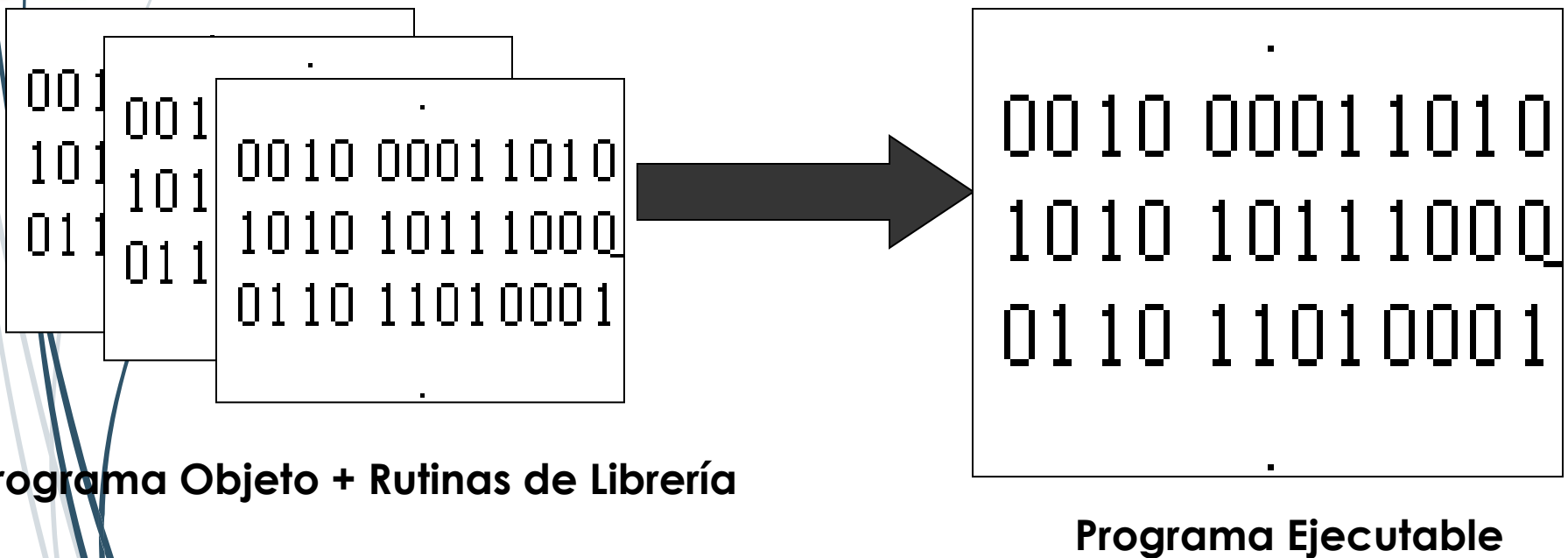
```
.
0010 00011010
1010 10111000
0110 11010001
.
```

Programa Objeto

# Algoritmo



★ **Enlace:** Consiste en la inclusión de determinadas rutinas internas del lenguaje y en el caso de programación modular, se enlazan los distintos módulos.





# Algoritmo

➤ **Prueba:** consiste en determinar si el programa funciona correctamente, los datos de entrada que se utilizan para la prueba, deben ser incorrectos

```
0010 0001 1010
1010 1011 1000
0110 1101 0001
```



Programa Ejecutable

Aplicación



# Otros conceptos



- ✦ **Método de Cálculo:** procedimiento que tiene todas las características de un algoritmo excepto que posiblemente carezca de ser finito
- ✦ **Programa:** algoritmo expresado en un lenguaje de programación. Es un conjunto de instrucciones que se dan a una computadora para realizar un proceso determinado.
- ✦ **Instrucción:** es la especificación de una acción determinada, expresada en un lenguaje de programación.



# Otros conceptos



✦ **Acción:** es un suceso que dura un tiempo finito y produce un resultado o efecto perfectamente definido y previsto. Opera sobre un objeto y es posible reconocer el efecto de tal acción por los cambios de estado presentados por el objeto.

✦ **Programación:** disciplina dentro de la computación dedicada a la resolución de problemas mediante el computador, cuyo producto final son los programas. Es la actividad de expresar un algoritmo en forma de programa

# Lenguajes



- ✦ **Lenguaje:** conjunto de reglas y convenciones que se utilizan para comunicar información.
- ✦ **Lenguaje de Programación:** es un lenguaje que permite la traducción de un algoritmo para que sea entendible por el computador. Son los lenguajes utilizados para escribir programas de computadoras.
- ✦ **Tipos de Lenguaje de Programación:**
  - Lenguaje de Máquina
  - Lenguaje de Bajo Nivel (Ensamblador)
  - Lenguaje de Alto Nivel

# Lenguaje de máquina



- ✦ Son aquellos que están escritos en lenguaje directamente inteligibles por la máquina, ya que sus instrucciones son cadena binarias (0's ,1's) que especifican una operación.
- ✦ Las instrucciones en lenguaje máquina dependen del Hardware de la computadora, difiere de una PC a otra.
- ✦ La ventaja de los lenguajes máquina es que ofrecen mayor velocidad de ejecución.
- ✦ Existen algunas desventajas: dificultad y lentitud en la codificación, poca fiabilidad, los programas no son portables.

# Lenguaje de bajo nivel

- ✦ Son lenguajes que permiten escribir programas con instrucciones similares al lenguaje humano.
- ✦ Son más fáciles de utilizar que los lenguajes de máquina, pero éstos también dependen de la máquina en particular.
- ✦ El mejor ejemplo es el lenguaje ensamblador
- ✦ Un programa escrito en lenguaje ensamblador requiere una fase de traducción.
- ✦ Entre las desventajas tenemos: que este lenguaje depende de la máquina y demanda una mayor exigencia para los programadores, ya que deben conocer tanto las técnicas de programación así como el interior de la máquina



# Lenguaje de alto nivel



- ✦ Son los más utilizados por los programadores
- ✦ Son independientes de la máquina
- ✦ Los programas escritos en estos lenguajes son portables
- ✦ Aumento de la ocupación de la memoria
- ✦ Las estructuras de los programas se basan en reglas sintácticas
- ✦ Ejemplos: Borland C, C++, Pascal, Ada, Prolog, Smalltalk, Visual Basic

# Otros conceptos

- ✦ **Notación Pseudoformal:** es un lenguaje de especificación de algoritmo. El uso de tal lenguaje hace el paso de codificación final relativamente fácil. La ventaja de este lenguaje es que el programador puede concentrarse en la lógica y en la estructura de control y no preocuparse de las reglas de un lenguaje
- ✦ **Programa Fuente:** es el código del programa original escrito en un lenguaje de programación. También conocido como código fuente
- ✦ **Programa Objeto:** es un programa en lenguaje de máquina que resulta de la compilación de un programa fuente.

# Otros conceptos



- ✦ **Traductores de Lenguajes:** son programas que traducen a su vez los programas fuentes escritos en lenguajes de alto nivel a código máquina. Estos se dividen en intérpretes y compiladores.
- ✦ **Compilador:** es el proceso de traducción de programas fuentes a programas objetos. En este proceso se debe utilizar como paso intermedio un programa llamado enlazador (linker) el cual da como resultado un programa en lenguaje máquina directamente ejecutable.

# Otros conceptos



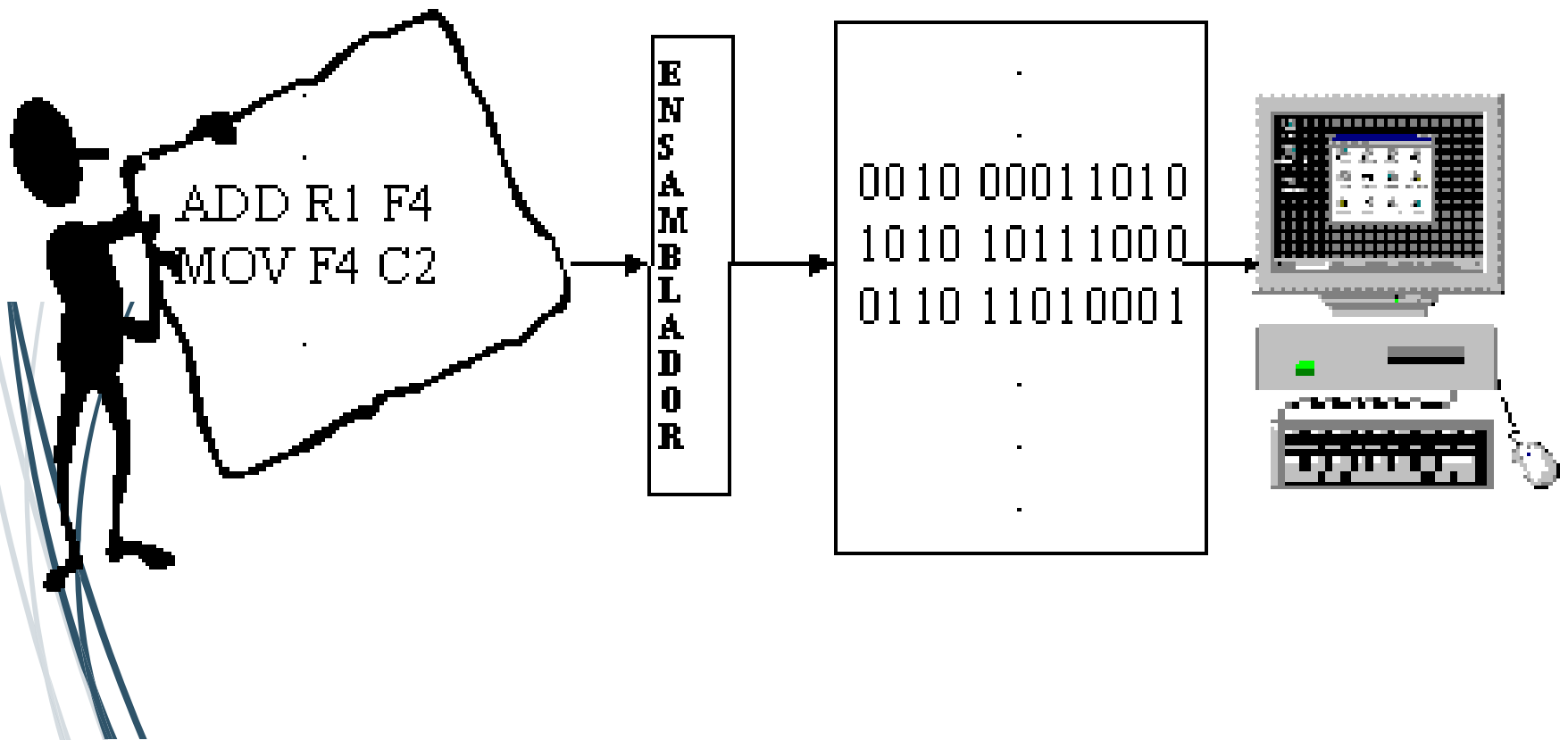
★ **Intérprete:** es un traductor que toma un programa fuente, lo traduce y a continuación lo ejecuta. (Basic, Smalltalk)



# Otros conceptos



- **Ensamblador:** programas que transforman programas escritos en lenguaje ensamblador a lenguaje máquina y ejecutable





# Sistemas Operativos



## Definición:

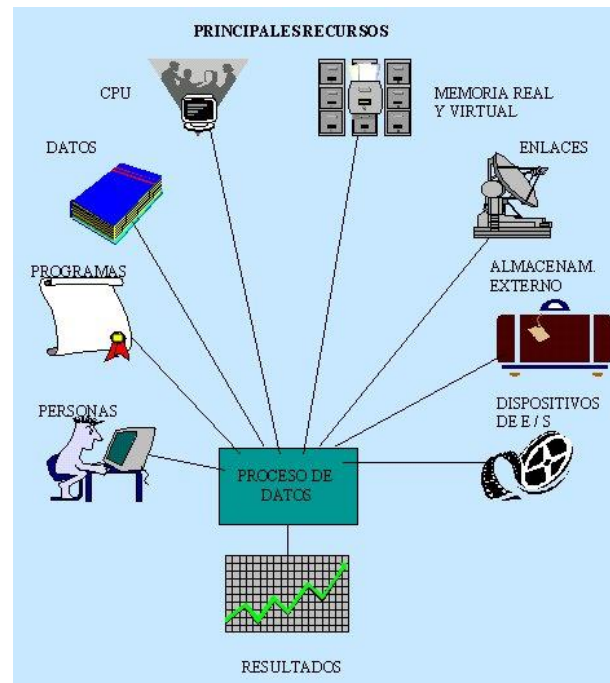
- ✦ Son programas hechos en lenguaje de bajo nivel, compuestos por un conjunto de normas y procedimientos para operar una computadora.
- ✦ Es un conjunto de programas de control que tienen por objeto facilitar el uso de la computadora y conseguir que ésta se ejecute eficientemente
- ✦ Es un programa que se encarga de gestionar y asignar los recursos Hw al usuario, entendiendo por recursos Hw: la CPU, la memoria principal, Discos Duros y otros periféricos.

# Sistemas Operativos



## Objetivos:

- ★ Su principal labor es control sistemáticamente los recursos de la computadora al momento de ejecutar un proceso, es decir, es el encargado de supervisar, administrar y gerenciar los elementos con que cuenta el computador.



# Sistemas Operativos

## Objetivos:

✦ Gestionar y administrar eficientemente los recursos Hw de la computadora, permitiendo **ejecutar concurrentemente varios programas** sin que haya conflicto en el acceso de cada uno de los recursos que se requiere y sin que ningún programa monopolice un recurso determinado.

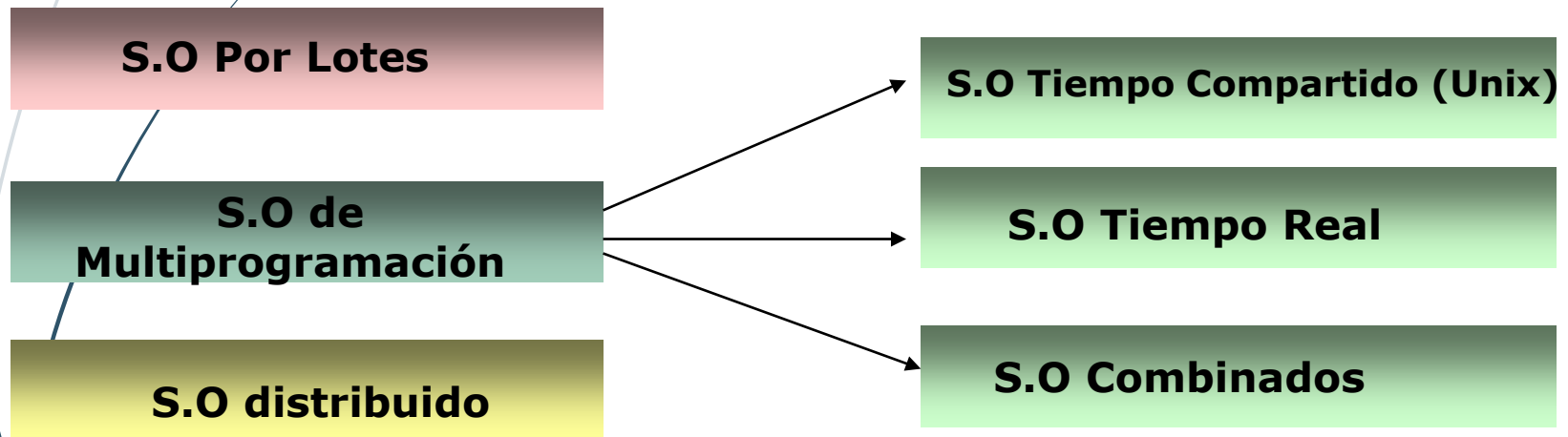


# Sistemas Operativos



## Tipos:

- Se clasifican de acuerdo a la forma de ofrecer los servicios:





# Sistemas Operativos



## □ Sistemas por lotes (*batch*)

- Se podrían considerar como los primeros SOs reales
- El SO estaba almacenado en memoria
- Cargaba un único trabajo en memoria (desde el lector de tarjetas)
- Ejecutaba el trabajo (generaba su salida)
- Cargaba el siguiente trabajo
- Las tarjetas de control indicaban qué hacer al SO



# Sistemas Operativos



## ❑ Problemas

- Largas esperas entre lotes de trabajo
- La dificultad de manejar la concurrencia entre E/S y ejecución en CPU ayudó a estimular el desarrollo de los SSOO multiprogramados
- Carecen de interactividad entre el usuario y los trabajos que se ejecutan.

# Sistemas Operativos



## ■ S.O de Multiprogramación

- Objetivo: Aprovechar los tiempos de espera de un trabajo en la CPU para ejecutar instrucciones de otro trabajo
- Método:
  - - Mantener los trabajos simultáneamente en memoria
  - - Elegir el trabajo a conmutar
- Se optimiza la productividad (*throughput*) del sistema

# Sistemas Operativos



## ■ Tiempo Compartido

- Objetivo: Permitir la interacción entre el usuario y el trabajo que se está ejecutando.
- Método:
  - Utilizar las técnicas de multiprogramación y planificación de la CPU para proporcionar a cada usuario una pequeña proporción del tiempo de CPU.
- Características:
  - Cada usuario tiene la impresión de disponer de la máquina completa.
  - Se intenta optimizar el tiempo de respuesta.
  - Basados en la asignación de fracciones de tiempo (se divide el tiempo de CPU de forma equitativa entre los procesos).

# Sistemas Operativos



## ■ Sistemas de tiempo real

- El factor tiempo se convierte en una variable fundamental.
- El sistema operativo debe responder siempre dentro de los márgenes requeridos por el sistema controlado, si se sale de esos márgenes, se tendrán que buscar otras soluciones.
- Existen actualmente lenguajes de programación (Ada) que nos dan grandes posibilidades para controlar procesos en tiempo real.

# Sistemas Operativos



## Dónde se utilizan....(SO tiempo real)

- Sistemas de control de procesos
- Sistemas de control de vuelo
- Sistemas de control de automóviles
- Sistemas de defensa
- Sistemas de vigilancia intensiva
- Sistemas de telecomunicación, etc.



# Sistemas Operativos



## ■ Sistemas Operativos Distribuidos

- Los usuarios pueden acceder a recursos remotos de la misma manera en que lo hacen para los recursos locales.
- Permiten distribuir trabajos, tareas o procesos, entre un conjunto de procesadores. Puede ser que este conjunto de procesadores esté en un equipo o en diferentes, en este caso es transparente para el usuario.

• Los sistemas distribuidos deben de ser muy confiables, ya que si un componente del sistema se descompone otro componente debe de ser capaz de reemplazarlo.

# Sistemas Operativos



- ✦ **Proceso:** es básicamente un programa en ejecución. Consta del programa ejecutable, sus datos y pila, contador y otros registros, además de toda la información necesaria para ejecutar el programa.
- ✦ **Multiprogramación:** modo de operación en cual se incrementa la utilización del CPU, puesto que éste se conmuta de una tarea a otra para lograr que las mismas avancen, mientras se mantienen en uso los dispositivos periféricos. Estas tareas se encuentran en memoria principal
- ✦ **Grado de Multiprogramación:** mide el uso de la CPU desde un punto de vista probabilístico. Normalmente es el rendimiento que se obtiene del CPU en función del número de procesos activos que mantiene un sistema.



# Paradigmas de Programación



Es una colección de patrones conceptuales que modelan la forma de razonar sobre problemas, de formular algoritmos, y a la larga, de estructurar programas.

Tipos de Paradigmas:

- Paradigma Funcional
  - Paradigma Imperativo
- Paradigma Orientado a objetos
  - Paradigma Ensamblador
- Paradigma Lógico

# Paradigma Funcional



- Tiene su base en el concepto de función matemática:  
 $f: \text{dominio} \rightarrow \text{rango}$
- Para programar:
  - Se construyen funciones sencillas
  - Se construyen funciones más complejas a partir de las sencillas
  - Se evalúan las funciones sobre los datos de entrada
- Ejemplos: APL, ML, Lisp.

# Paradigma Funcional



*Ejemplo:* Mayor de tres números en funcional

Definición de funciones max y max1:

**$\text{max}(x,y) := \text{if } x > y \text{ then } x \text{ else } y$**

**$\text{max1}(x,y,z) := \text{max}(\text{max}(x,y),z)$**

NOTA: ":@" en este caso significa "se define como"



# Paradigma Imperativo



- ✦ Este paradigma viene bien representado por la arquitectura Von Neuman (1903-1957), ya que utiliza este modelo de máquina para conceptualizar las soluciones: "Existe un programa en memoria que se va ejecutando secuencialmente, y que toma unos datos de la memoria, efectúa unos cálculos y el resultado se almacena en alguna celda de memoria".
- ✦ La memoria del computador permite tener almacenado, en celdas numeradas consecutivamente, tanto las operaciones como los datos a manipular.
- ✦ Los datos se almacenan en variables.

# Paradigma Imperativo



- ✦ Las operaciones a ejecutar se llaman instrucciones, cuya realización se llama ejecución.

- ✦ La instrucción principal es la asignación, la cual tiene el siguiente formato:

`< variable > ← < expresión >`

- ✦ La forma básica de expresar un algoritmo consiste en declarar variables, diseñar una secuencia de asignaciones que transformen los valores almacenados.

# Paradigma Orientado a Objetos



- ✦ Es muy abstracto
- ✦ Ya no se ve las variables como celdas de memorias sino como objetos
- ✦ Un objeto pertenece a cierta clase, que define el conjunto de operaciones utilizables
- ✦ Ejemplos: Smalltalk, C++, Object Pascal

# Paradigma Ensamblador



- ✦ Los valores manipulados no son abstractos, sino que se maneja su representación binaria en memoria
- ✦ No se usan expresiones funcionales sino zonas de almacenamiento intermedio, llamados registros
- ✦ Se hace operación tras operación

# Paradigma Lógico

- ✦ Tiene su base en el concepto de predicado o relación
- ✦ Un programa lógico está formado por hechos y reglas
- ✦ Para programar:
  - Se definen hechos (o predicados básicos)
  - Se diseñan implicaciones para definir predicados complejos
  - Se determina la veracidad de los predicados para individuos concretos
- ✦ Ejemplo: Prolog, único en su clase



**Gracias por su  
atención...!!**