

ALGORITMOS



**PRESENTADO POR:
JAIME CORONEL SANTILL'AN**

PRIMERA UNIDAD



INFORMATICA

El tratamiento de la información es tan antiguo como el hombre y se ha ido potenciando y haciendo más sofisticado con el transcurso del tiempo hasta llegar a la era de la electrónica.

El hombre no ha parado a lo largo de la historia de crear máquinas y métodos para procesar la información.



DEFINICIONES BASICAS

Informática: Proviene de la unión de las palabras información y automática. Es la ciencia que permite el tratamiento automático de la información por medio de computadores.

Información: Es cualquier conjunto de símbolos que representen hechos, objetos o ideas.



- **Computadora:** Es una máquina compuesta de una serie de circuitos electrónicos, capaz de recoger unos datos de entrada, efectuar con ellos ciertos cálculos y devolver los datos por medio de algún medio de salida.



ASPECTOS QUE DIFERENCIAN EL COMPUTADOR DEL RESTO DE LA MÁQUINA CON CAPACIDAD DE TRATAR INFORMACIÓN

- Gran velocidad de tratamiento de la información.
- Gran potencia de cálculo aritmético y lógico.
- Capacidad para memorizar los programas y datos necesarios para resolver cualquier problema técnico o de gestión.



- Capacidad de comunicación con las personas y con otras máquinas y dispositivos para recibir o transmitir datos.
- Posibilidad de tratamiento de datos en tiempo real.
- Actúa sin intervención de un operador humano y bajo el control de un programa previamente almacenado en la propia computadora.



TIPOS DE INFORMACIÓN

Datos: Son conjuntos de símbolos que utilizamos para expresar o representar un valor numérico, un hecho, un objeto o una idea, en la forma adecuada para su tratamiento.

Instrucciones: Indican a la computadora qué es lo que debe realizar

FUNCIONAMIENTO BASICO DE UN COMPUTADOR



REPRESENTACION DE LA INFORMACION

La información se almacena dentro del ordenador de forma codificada en código binario.

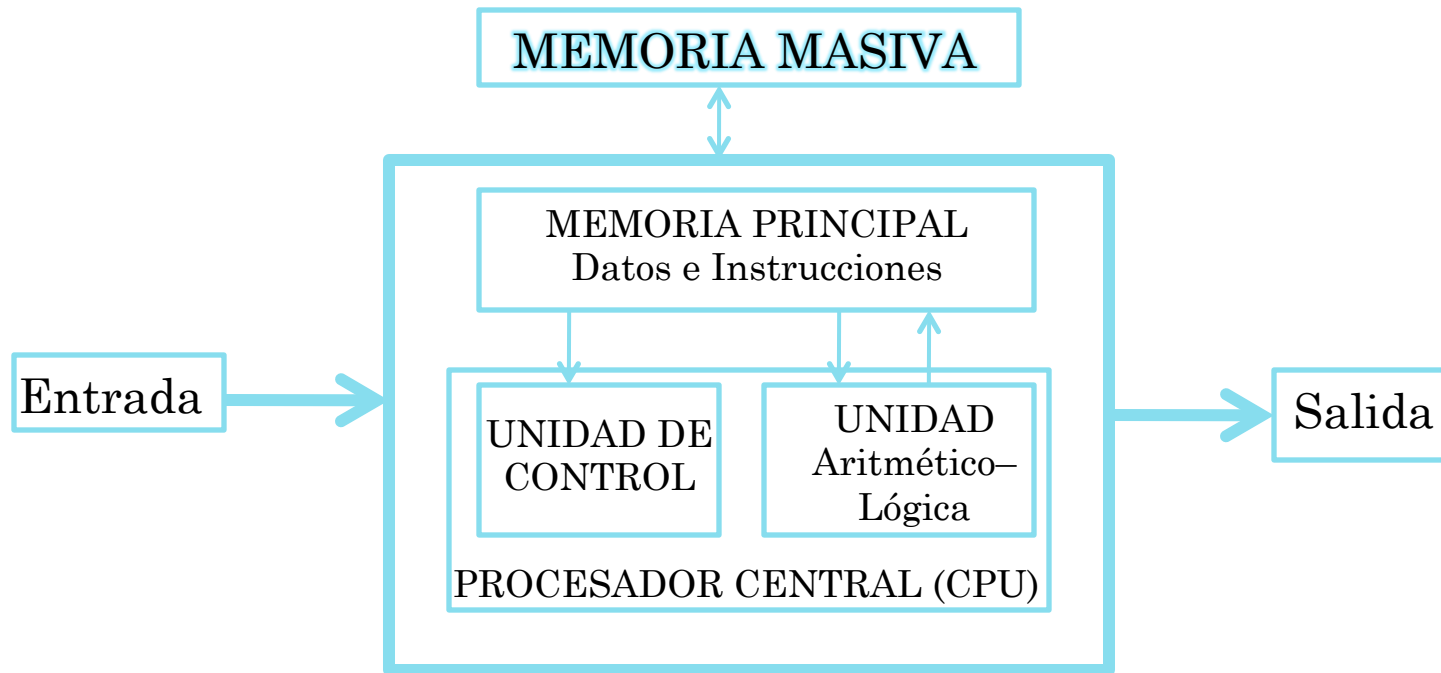
El **BIT** es la unidad elemental de información que equivale a un valor binario y constituye, dentro de una computadora la capacidad mínima de información.



- Un byte es el número de bits necesarios para almacenar un carácter. Este número va a depender del código concreto usado por la computadora, aunque generalmente se usan 8, esto es, podemos asumir que un byte equivale a 8 bits.



ESTRUCTURA DE UN COMPUTADOR



CLASIFICACION DE LAS COMPUTADORAS

En función del Uso

- Computador de uso general
- Computador de uso específico

En función de la potencia, capacidad o el tamaño del computador

- Supercomputadores
- Macrocomputadores (mainframes)
 - Minicomputadores
- Estaciones de trabajo (workstations)
- Computadores profesiones / personales (Pc's)



PROGRAMAS E INSTRUCCIONES

Un **programa** es un conjunto de sentencias que se dan a una computadora indicándole las operaciones que se desea que realice. Las sentencias son conjuntos de símbolos, que se construyen siguiendo un lenguaje de programación y se clasifican en:

- Sentencias imperativas o instrucciones
 - Sentencias declarativas

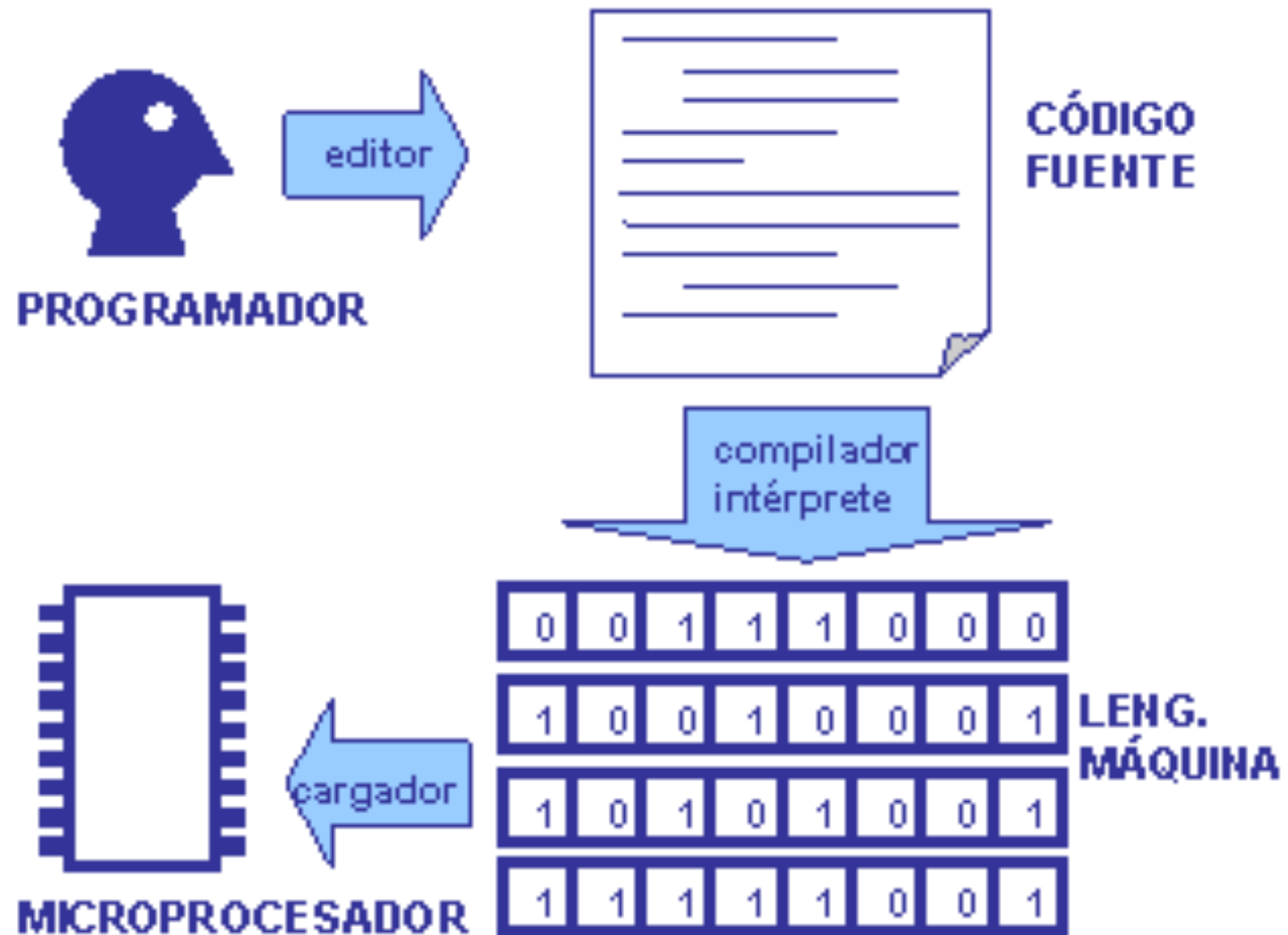


Las **instrucciones** se pueden clasificar en:

- De transferencia de datos
- De tratamiento
- De bifurcación y saltos

Los **lenguajes de alto nivel** son independientes de la computadora y facilitan la programación.





COMPILADORES E INTERPRETES

Los **Traductores** son programas que toman como datos de entrada programas escritos en un lenguaje de alto nivel y devuelven como datos de salida el programa en lenguaje máquina de esa computadora equivalente, de tal forma que ya si puede ser ejecutado por la CPU de la computadora.

Los **Compiladores** son traductores que cogen en conjunto el programa escrito en lenguaje de alto nivel, programa fuente, lo traducen y generan un programa en código máquina, programa objeto.



Los **Intérpretes**, van analizando, interpretando y ejecutando instrucción a instrucción del programa fuente y por tanto no generan programa objeto.

El **sistema operativo** es una serie de programas que permiten controlar y utilizar de forma eficiente y cómoda la computadora



PROGRAMACION DE COMPUTADORAS

El **Hardware** es el soporte físico de la computadora y el **Software** es el soporte lógico.

LENGUAJES DE PROGRAMACION

- FORTRAN
- COBOL
- LISP
- PASCAL
- PROLOG
- SMALLTALK
- OBJECT PASCAL
- DELPHI
- JAVA
- “C
- C++



LA LÓGICA DE PROGRAMACIÓN

Uno de los aspectos importantes en la programación de computadoras, es la lógica de la programación, es por eso que esta actividad pretende, a parte de relajar, activar la capacidad de analizar y encontrar los métodos adecuados de solucionar diferentes problemas



TIPOS DE DATOS

- Datos Numéricos:

Enteros(Representan los números que no posee componente fraccionaria y pueden ser tanto positivos como negativos) y

Reales (Representan todos los números que poseen componente fraccionaria y también pueden positivo o negativo)

- **Datos Lógicos (booleano)**, este tipo de dato solo puede tomar uno de dos valores (verdadero o falso)



◉ **Datos tipo Carácter:** Representan datos alfanuméricos que pueden ser:

Cadena de caracteres (string): es una sucesión de caracteres numéricos, letras, símbolos, etc.

Carácter: (char): contiene solo un carácter y también se incluye las comillas para su asignación “I”



VARIABLES Y CONSTANTES

Una **variable** es un espacio reservado en el computador para contener valores que pueden cambiar durante el desarrollo del algoritmo.

Variables locales: es aquella que afecta únicamente el subprograma.

Variable Global: variable que afecta a un programa en todo su contexto, programa principal y modulo.

Una **Constante** es un espacio reservado para contener valores que no cambian a lo largo de la ejecución de un algoritmo



CARACTERÍSTICAS DE LOS NOMBRES DE LAS VARIABLES Y CONSTANTES

- Normalmente deben iniciar con una letra
- No deben contener símbolos ni signos de puntuación (#, (?...)
- No deben contener espacios en blanco



OPERADORES

Las operaciones que se realicen sobre estas variables y/o constantes, están definidas por una serie de operadores, entre los cuales se encuentran:

Aritméticos

- Potencia $\wedge$ **
- Producto *
- División / Div Mod
- Suma +
- Resta -

Relacionales

- Igual a =
- Menor que <
- Menor o igual que <=
- Mayor que >
- Mayor o igual que >=
- Distinto a <>



Alfanuméricos

○ Concatenación +

Ej.

'UN' + 'AD' = 'UNAD'

Lógicos

• Negación	Not	no
• Conjunción/producto	And	y
• Disyunción/suma	Or	o

Paréntesis

Permite alterar el orden en que realizan las diferentes operaciones



SEGUNDA UNIDAD



Algoritmos

Es un conjunto de pasos lógicos ordenados, secuencialmente y finita, escritos de tal forma que permiten visualizar la solución de un problema determinado en un momento específico.

ALGORITMO: NOMBRE

El nombre en latín de **algoritmo** proviene de la traducción que realizó **Fibonacci**, de la obra del matemático árabe **Al'Khwarizmi** llamada , *Algoritmi de Numero Indorum*.



CARACTERÍSTICAS

Un algoritmo
Debe ser...



FINITO

Debe tener
terminar en algún
momento



CONCRETO

Debe indicar un
orden de
realización de
cada paso.



LEGIBLE

Debe estar bien
estructurado
para su fácil
entendimiento.



EFICIENTE

Debe realizar las operaciones con un mínimo de utilización de recursos.



**NO
AMBIGUO**

Debe generar el
mismo resultado
siempre que se
siga.



DEFINIDO

Debe realizar las
funciones u
operaciones para
las que fue
creado.



PRECISO

Debe estar libre
de errores.
(Validado)



ALGORITMO: ESTRUCTURA

Datos

Corresponden a los datos requeridos para realizar el algoritmo (datos de entrada) y los datos que son generados (datos de salida)

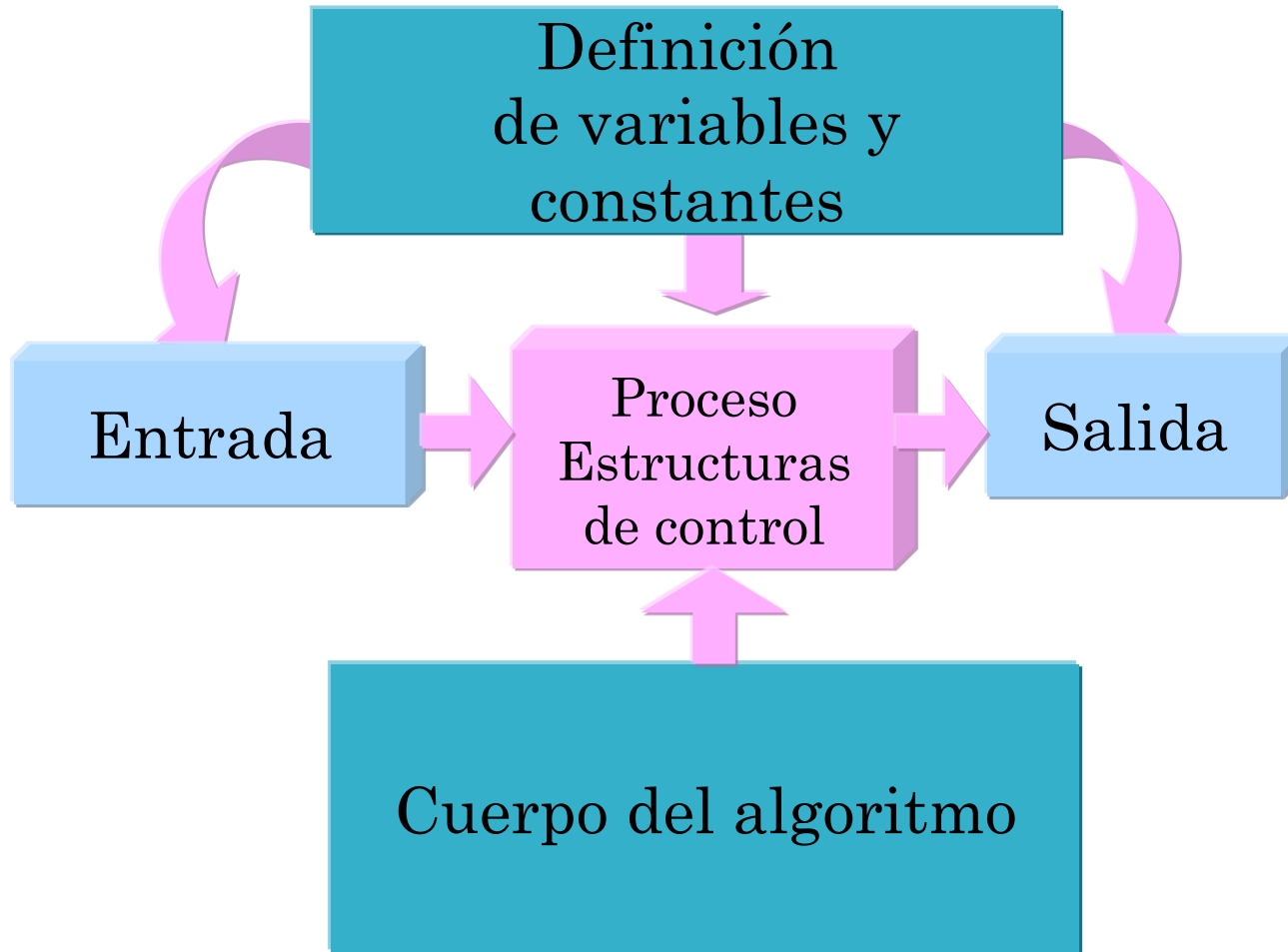
Procesos

Conforma el grupo de instrucciones que realizan las operaciones con los datos.

Estructuras de Control

Determinan la organización de las instrucciones que deben ser realizadas.

ELEMENTOS



QUIENES PUEDEN HACER UN ALGORITMO

Toda persona, implícitamente y diariamente diseña y realiza algoritmos, para dar solución a situaciones cotidianas de forma natural.

Sin embargo el programador, diseña el algoritmo consciente de que al realizar cada paso obtendrá la solución de un problema específico.



REQUISITOS

Los algoritmos se crean para resolver problemas.
Es importante que junto al algoritmo, describamos claramente el problema que éste nos permite resolver.

Debe Definirse el problema



No debemos omitir el contexto de nuestros algoritmos.

Es necesario establecer lo que se necesita y dónde se debe comenzar.

Debe estar dentro de contexto



Seguir los pasos del algoritmo debe llevarnos a la resolución del problema.

Siempre que sea posible seguiremos personalmente los pasos de nuestro algoritmo para comprobar que son efectivamente correctos y conducen efectivamente a la solución esperada.

Debe resolver el problema



Por ejemplo:

Si se requiere hallar la velocidad de un automóvil, es necesario, definir si la distancia debe ser en metros, kilómetros, etc. y el tiempo estará dado en segundos u horas, ya que la velocidad puede representarse en Km/h ó mts/seg.

Debe evitar la
ambigüedad



ALGORITMO: TÉCNICAS DE DISEÑO

Esta técnica permite dividir el problema en pequeñas partes, a las cuales se les da solución por separado, luego se integran las soluciones para resolver el problema principal.

Top Down

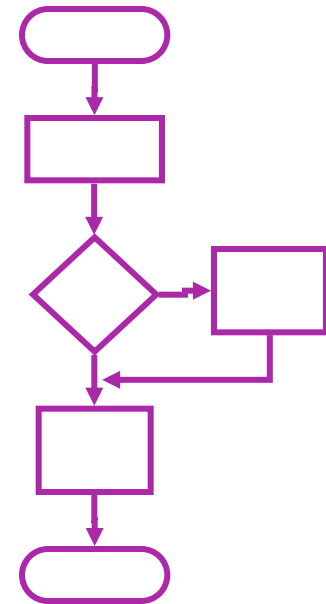
Divide y vencerás



ALGORITMO: TÉCNICAS DE REPRESENTACIÓN

Es una técnica que permite representar gráficamente las operaciones y estructuras que se van a realizar, mediante una simbología estándar, con un único punto de inicio y uno de finalización.

Diagrama de Flujo



PSEUDOCÓDIGO

Esta técnica permite representar el algoritmo mediante un lenguaje más estructurado, facilitando su posterior codificación.

Inicio
Instrucción 1
Instrucción 2
Si condición entonces
 Instrucción 3
 :
Instrucción n
Fin

PSEUDOCÓDIGO: CÓMO SE HACE?

- 1 Cada instrucción que se va a realizar debe comenzar por un **verbo**, ejemplo: Muestre, Haga, Lea, etc.
- 2 Se debe mantener una **identación** o sangría sobre el margen izquierdo para identificar fácilmente el comienzo y final de las estructuras
- 3 La representación de las **estructuras** son similares u homónimas de los lenguajes de programación, ejemplo: inicio, fin, mientras que, repita hasta, si_entonces_sino, etc.



PSEUDOCÓDIGO: CÓMO SE HACE?

Inicio : Denota el punto de inicio del algoritmo.

Leer : Denota la acción de introducir datos o variables desde un dispositivo estándar de entrada.

Imprimir : Representa la acción de enviar datos desde variables a un dispositivo estándar de salida.

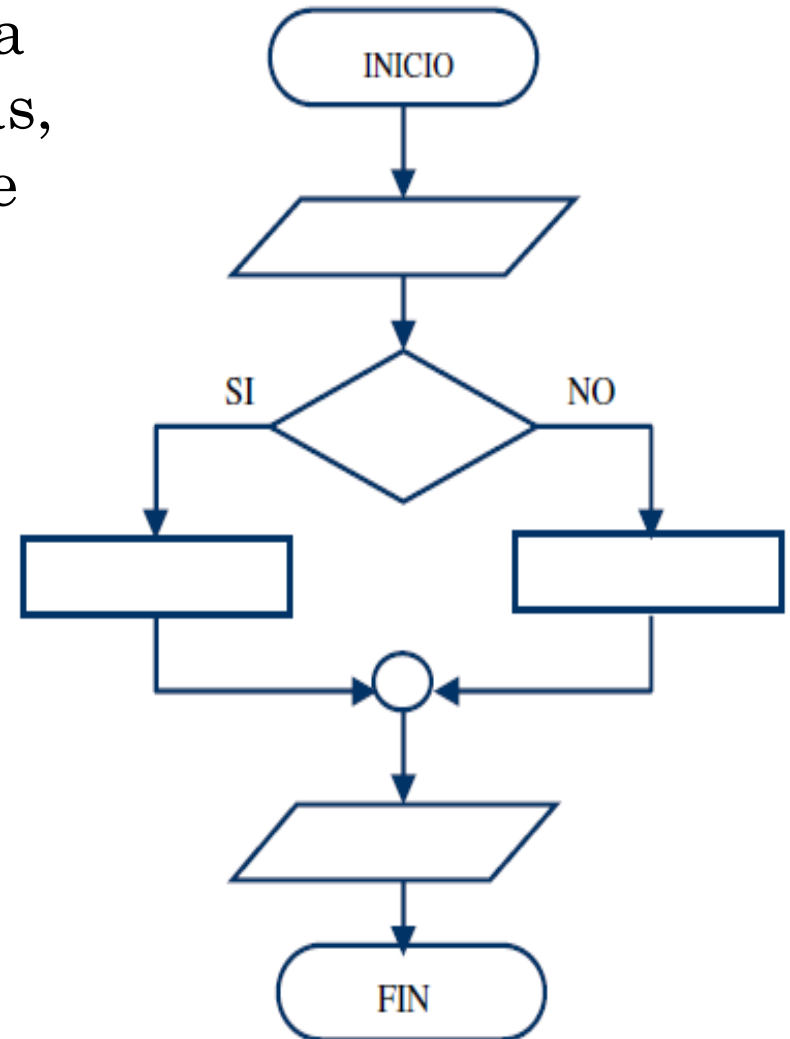
Calcular : Denota la realización de cualquier operación aritmética que genere valores para ser almacenados en una variable.

Fin: Denota el punto de finalización del algoritmo.



DIAGRAMAS DE FLUJO

Son una herramienta útil en la programación de computadoras, cuyas características, hace que se aplique no solo en la informática si no en todos los procesos que llevan una secuencia lógica



Aspectos fundamentales

- Sencillez: Construcción fácil.
- Claridad: Fácil reconocimiento de sus elementos.
- Utilización de normas en la construcción de algoritmos.
- Flexibilidad: Facilidad en las modificaciones.



SÍMBOLOS



Inicio y Final



Entrada y Salida (Lea y Escriba)



Operaciones Generales



Decisión



Símbolos de Conexión



Líneas de Flujo



HERRAMIENTAS QUE MEJORAN LA PRESENTACION

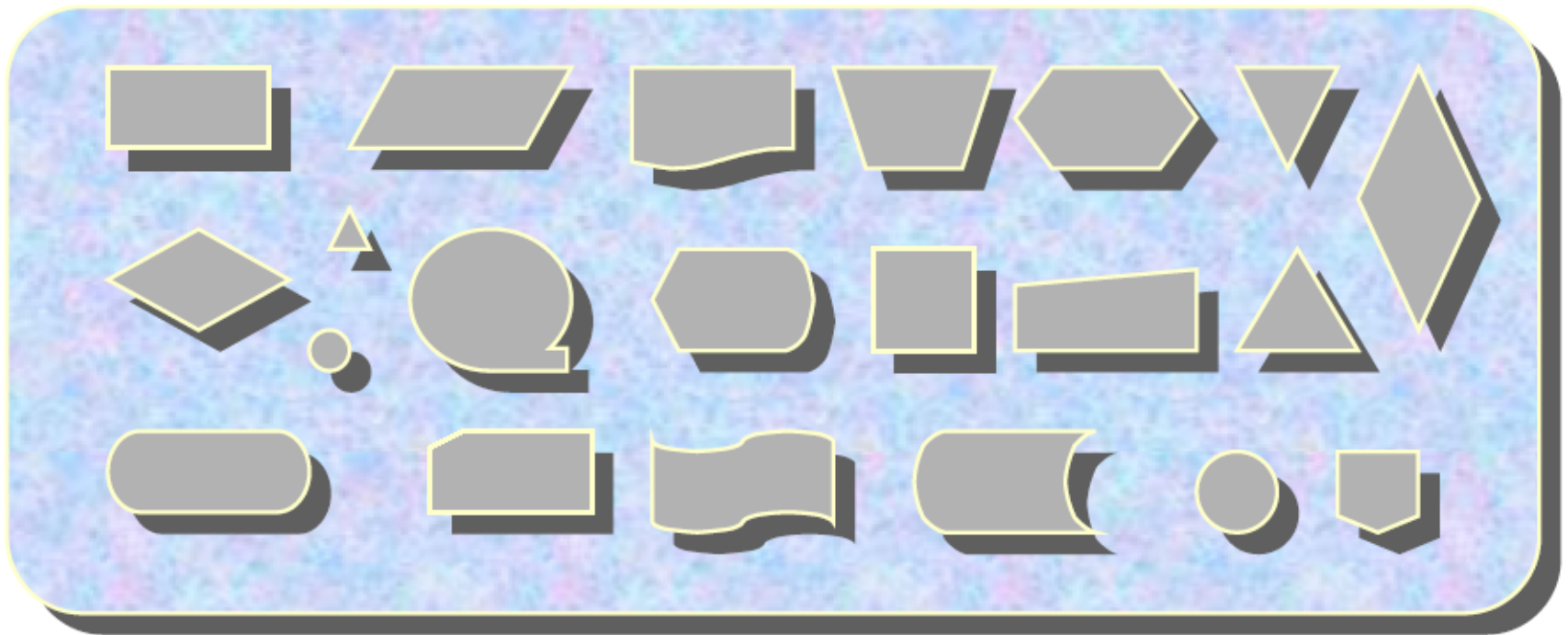
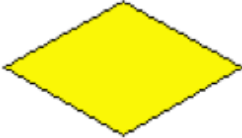


DIAGRAMA DE FLUJO: Simbología

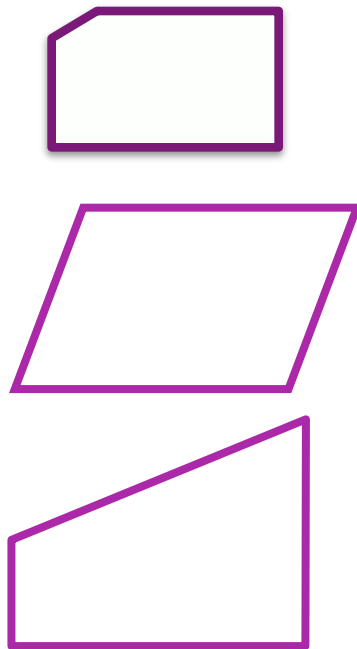
	Inicio/Final Se utiliza para indicar el inicio y el final de un diagrama; de Inicio sólo puede salir una línea de flujo y al final sólo debe llegar una línea		Decisión Indica la comparación de dos datos y dependiendo del resultado lógico (falso o verdadero) se toma la decisión de seguir un camino del diagrama u otro
	Entrada/Salida Entrada/Salida de datos por cualquier dispositivo (scanner, lector de código de barras, micrófono, parlantes, etc.)		Impresora/Documento. Indica la presentación de uno o varios resultados en forma impresa
	Entrada por teclado. Entrada de datos por teclado. Indica que el computador debe esperar a que el usuario teclee un dato que se guardará en una variable o constante		Pantalla Instrucción de presentación de mensajes o resultados en pantalla
	Acción/Proceso Indica una acción o instrucción general que debe realizarse (operaciones aritméticas, asignaciones, etc.)		Conector Interno Indica el enlace de dos partes de un diagrama dentro de la misma página
	Flujo/Flechas de Dirección Indica el seguimiento lógico del diagrama. También indica el sentido de ejecución de las operaciones		Conector Externo Indica el enlace de dos partes de un diagrama en páginas diferentes

Inicio
Fin

Se utiliza para
indicar el punto de
inicio y finalización
del diagrama



Lectura Captura



Permite indicar la
Entrada de datos
desde
un dispositivo
estándar



Proceso

Permite indicar la
realización
de un proceso
matemático,
o una operación de
asignación



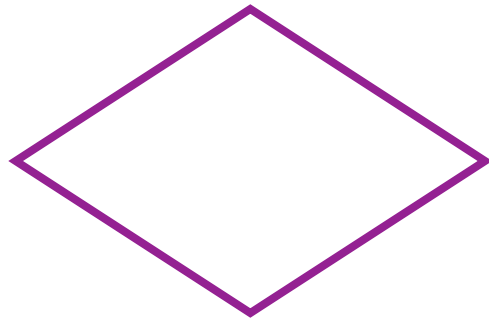
Impresión



Indica la realización
de
operaciones de salida
a un
dispositivo estándar
(el monitor o
impresora.)



Decisión



Permite establecer una condición relacional ó lógica que puede tomar un valor de verdadero o falso, de este símbolo se deducen 2 flujos alternativos de ejecución.



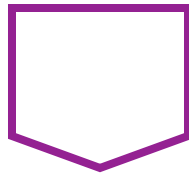
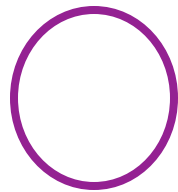
Flujo



Permiten enlazar los símbolos de un sentido único pueden ser horizontales o verticales. Estas no pueden entrecruzarse y cada una de ellas debe tener un único símbolo de partida y un único símbolo de destino.



Conectores



Permiten dar continuidad al diagrama si la página o área de trabajo esta llena, el círculo se utiliza como un conector dentro de la misma página, el otro símbolo se define como un conector a otra página.



ALGORITMO: FASES DE DISEÑO

Algoritmo

Definición del problema

Análisis del problema

Selección de la mejor alternativa

Diagramación

Prueba de escritorio



ALGORITMO: DEFINICIÓN DEL PROBLEMA

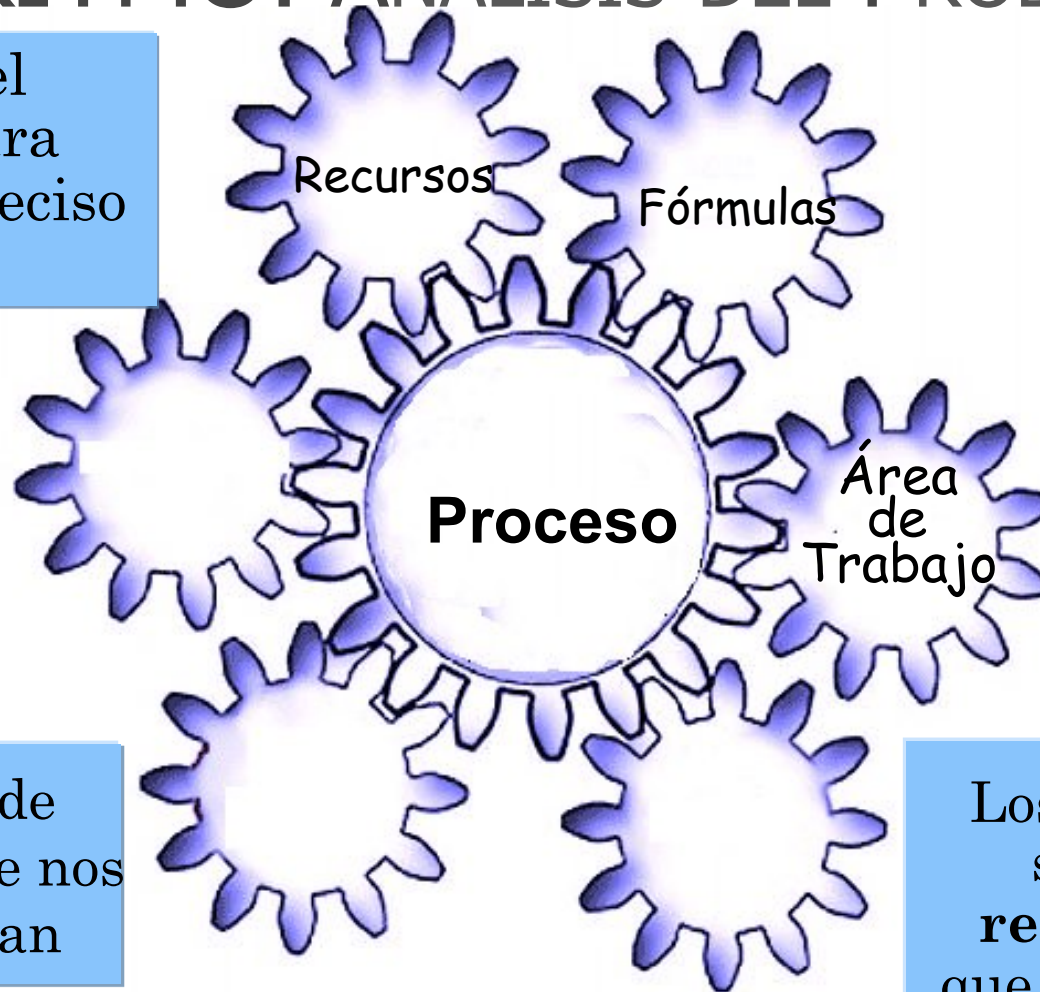
Está dada por el enunciado del problema, el cuál debe ser claro y completo

Es importante que conozcamos exactamente que se desea.

Mientras qué esto no se comprenda, no tiene caso pasar a la siguiente etapa.

ALGORITMO: ANÁLISIS DEL PROBLEMA

Entendido el problema para resolverlo es preciso analizar



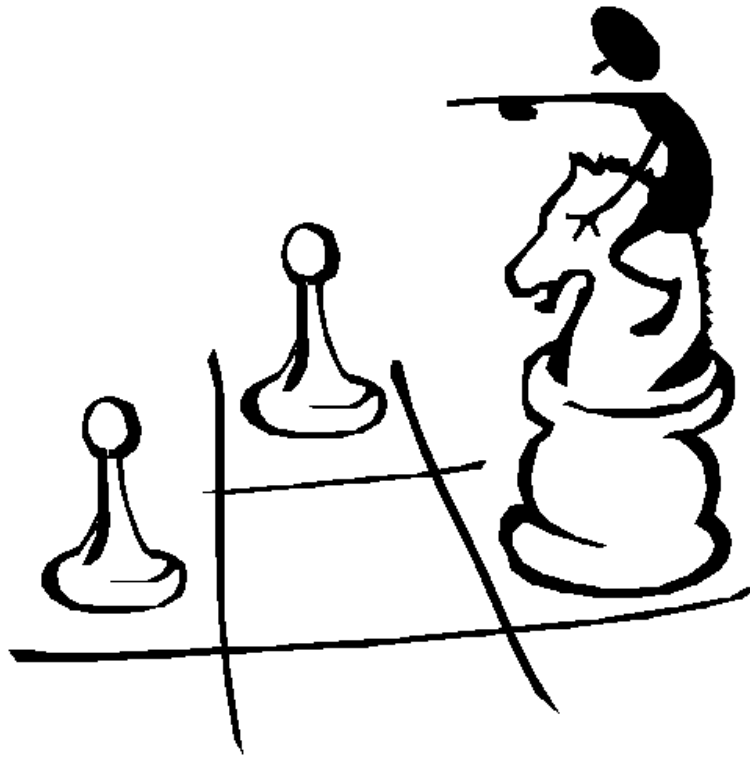
Los datos de **entrada** que nos suministran

Los datos de salida o **resultados** que se esperan



Analizado el problema
Posiblemente tengamos
varias formas de resolverlo





Lo importante es
determinar cuál
es la mejor alternativa



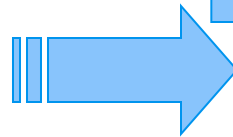


La que produce los
resultados Esperados en el
menor tiempo y al menor
costo



ALGORITMO: DIAGRAMACIÓN

Una vez que
sabemos
cómo resolver el
problema



Dibujar
gráficamente
la lógica de la
alternativa
seleccionada

Plasmar la
solucion
mediante el
Pseudocódigo

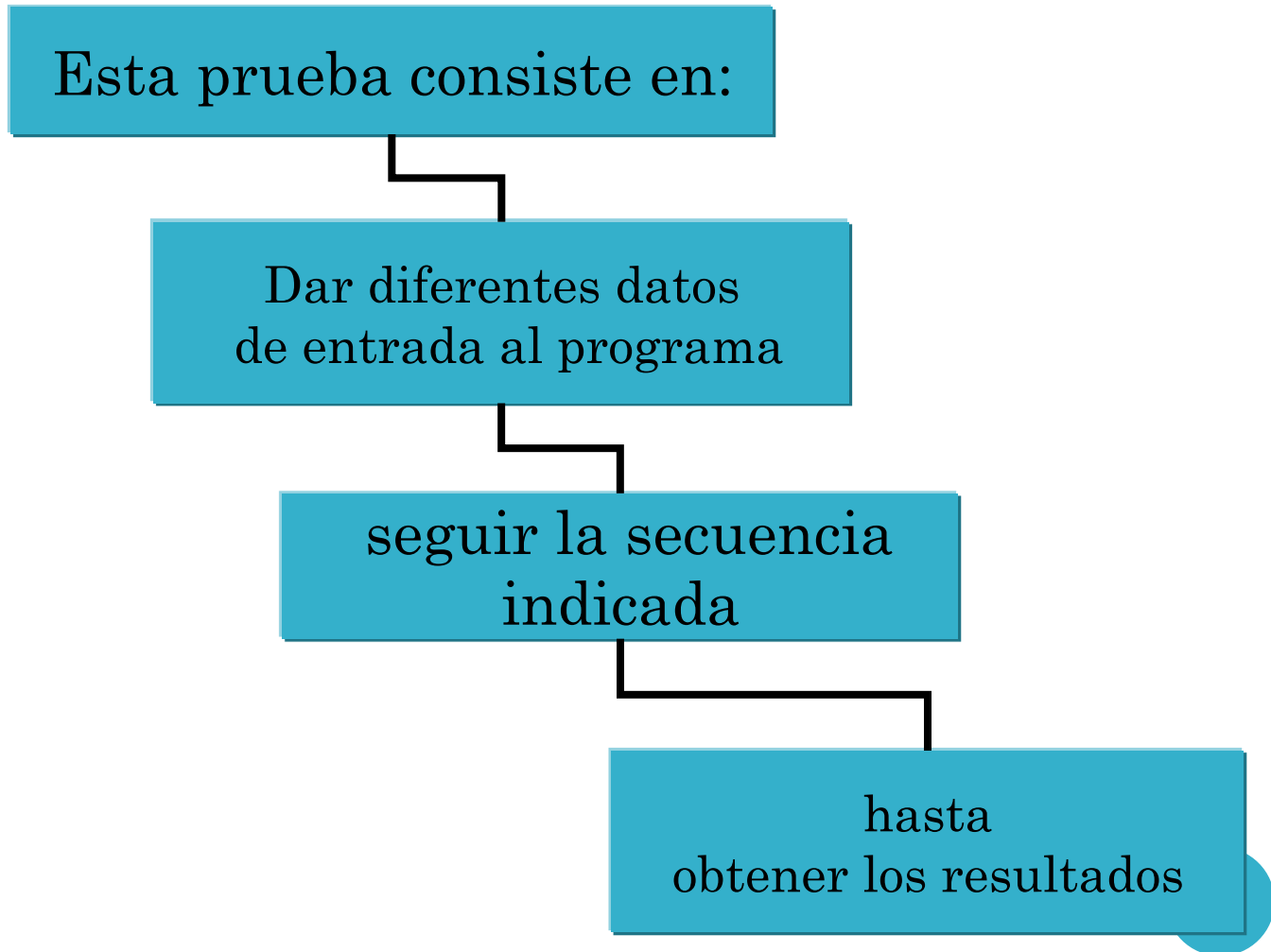
PRUEBA DE ESCRITORIO

Esta prueba consiste en:

Dar diferentes datos
de entrada al programa

seguir la secuencia
indicada

hasta
obtener los resultados



GRACIAS

