FIEE - 2014

FASES EN EL DESARROLLO DE UN PROGRAMA

Ing. José Antonio Poma G.

Fases: Resolución de un Problema

- ✓ El proceso de resolucion de un problema con una computadora conduce a la escritura (codificacion) de un programa y a su ejecucion.

1. ***Análisis del Problema.***
2. ***Diseño de Algoritmo.***
3. ***Transformación del Algoritmo en programa.***
   - **a) Codificación.**
   - **b) Compilación, enlace y ejecución.**
   - **c) Verificación.**
   - **d) Depuración.**
   - **e) Documentación.**

Análisis del Problema

- ✓ En esta fase se debe contemplar exactamente lo que debe hacer el programa y el resultado o solucion deseado.
- ✓ Presisando especificaciones detalladas de entrada y salida respondiendo a las siguientes Preguntas.

1. ***Que entrada se requiere?***
2. ***Cuál es la salida deseada?***
3. ***Que metodo produce la salida deseada?***

Diseño de un Algoritmo

DIAGRAMA DE FLUJO.

- ✓ Es una re presentacion grafica de un algoritmo utilizando simbolos normalizados por el ANSI (American National Standards Institute).

PSEUDOCODIGO

- ✓ Es una Herramienta de programacion en la que instrucciones se escriben en palabras similares al ingles o espanol, que facilitan tanto la escritura como la codificacion de programas.

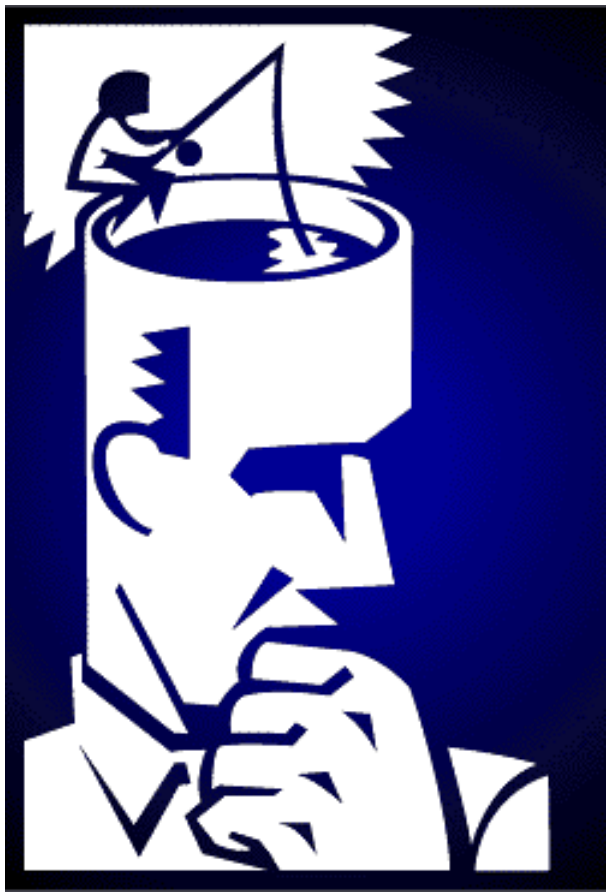
CARTA N-S (Nassi – Schneider)

- ✓ Es una Herramienta de programacion que permite la representacion grafica de un algoritmo. Este es el prototipo de los programas estructurados (Programacion anti-goto).

FIEE - 2014

ALGORITMO: Definición

- 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.



FIEE - 2014

ALGORITMO: Caracteríticas

CONCRETO

LEGIBLE

FINITO

Debe generar el mismo resultado siempre que se siga.

EFICIENTE

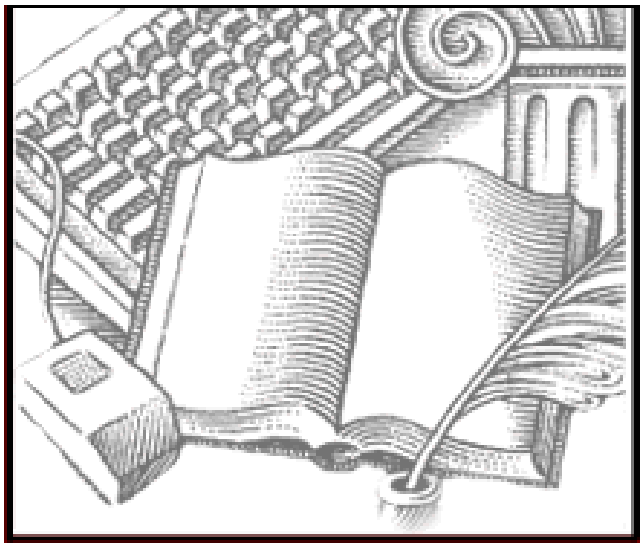
DEFINIDO

NO AMBIGUO

PRECISO

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.***



FIEE - 2014

FIEE - 2014

ALGORITMO: Estructura

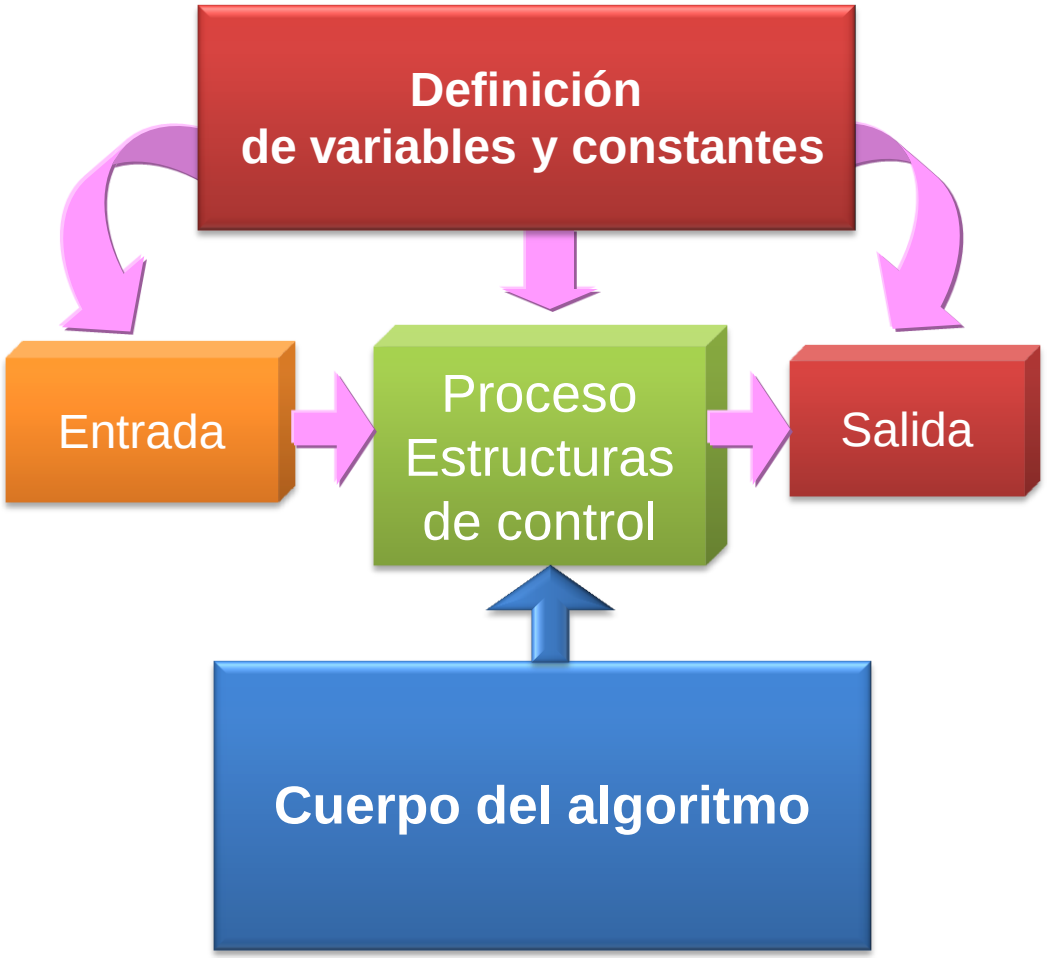
Determinan la organización de las instrucciones que deben ser realizadas.

- Datos
- Procesos
- Estructuras de Control

ALGORITMO: Elementos

FIEE - 2014

Las instrucciones que se van a realizar deben estar bien estructuradas y tener un orden lógico, con el fin de evitar inconsistencias en el resultado.



FIEE - 2014

ALGORITMO: Quienes pueden hacer un Algoritmo Como se hace?

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 conciente de que al realizar cada paso obtendrá la solución de un problema específico.

FIEE - 2014

ALGORITMO: Como se hace?

El no programador haría lo siguiente:

Buscar la página de cines en el diario local y mirar si ve la película anunciada. Si la ve anunciada mira en qué cine la hacen y se va a verla. Si no la ve anunciada, espera a los estrenos de la semana que viene.

ALGORITMO: Como se hace?

El programador sin embargo, lo haría de este otro modo:

1. Buscar la página de cines en el diario local, con fecha de hoy
2. Revisar la cartelera de arriba abajo y de izquierda a derecha, buscando entre los títulos existentes.
3. Si se encuentra el título La Guerra de las Galaxias, no seguir buscando. Apuntar el nombre del cine, su dirección y los horarios
4. Si no se encuentra el título en la cartelera, esperar una semana y volver a empezar el proceso a partir del punto 1 de esta lista.

ALGORITMO: Ejemplo

Encender un automóvil

1. Poner la llave.
2. Asegurarse que el cambio esté en neutro.
3. Pisar el el acelerador.
4. Girar la llave hasta la posición “arranque”.
5. Si el motor arranca antes de 6 seg, dejar la llave en la posición “encendido”.
6. Si el motor no arranca antes de 6 seg, volver al paso 3 (como máximo 5 veces).
7. Si el auto no arranca, llamar a la grúa.

ALGORITMO: Requisitos

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 Definirse del problema
- Debe estar dentro de contexto
- Debe resolver el problema
- Debe evitar la ambigüedad

FIEE - 2014

FIEE - 2014

ALGORITMO: Técnicas de Diseño

Está 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

FIEE - 2014

ALGORITMO: Técnicas de Presentación

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.

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

Diagrama de Flujo

Pseudocódigo

```
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.

Diagrama de Flujo: Simbología

Proceso

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

Diagrama de Flujo: Simbología

Flujo

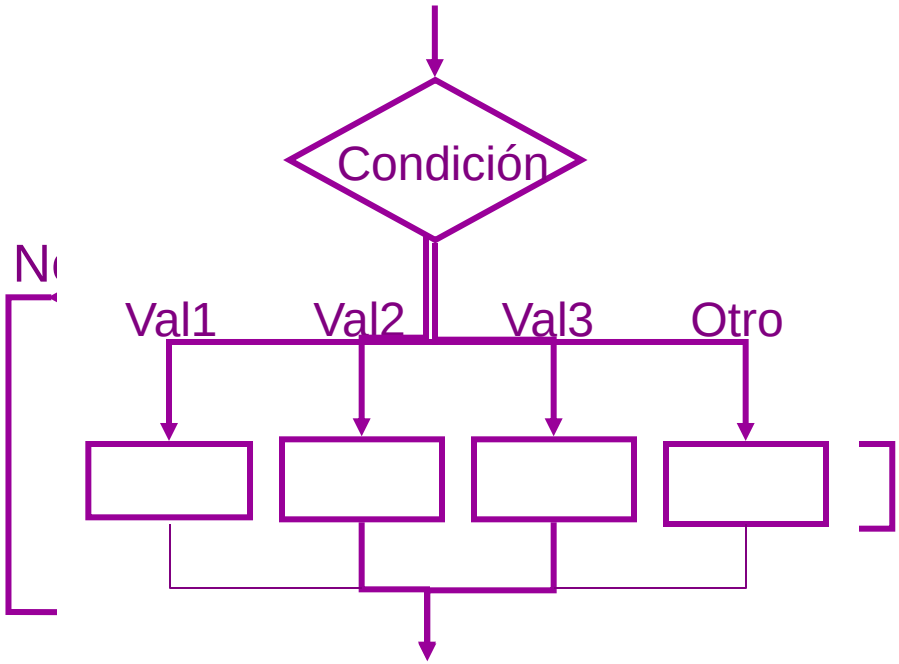
Permiten enlazar los símbolos de un sentido único pueden ser horizontales o verticales. Estas no pueden entrecruzarse y cada una de ellas debe tener un único símbolo de partida y un único símbolo de destino.

FIEE - 2014

Diagrama de Flujo: Simbología

Estructuras de Decisión (Condición)

Decisión Múltiple

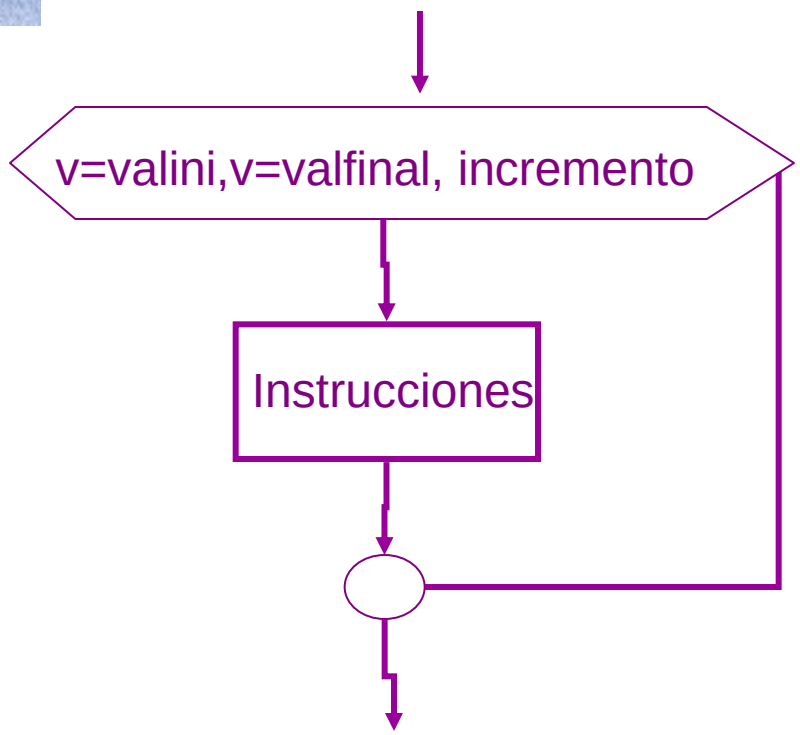


Caso condición
  Val1:
    Instrucciones
  Val2:
    Instrucciones
  Val3:
    Instrucciones
  Otros:
    Instrucciones
Fin Caso

Diagrama de Flujo: Simbología

Estructuras de Ciclo

Ciclo Para




```
:
Para v=valini, v=valfinal, inc
    Instrucciones
 Fin Para
:
```

v: variable
valini: valor inicial
valfinal: valor final
inc: incremento

FIEE - 2014

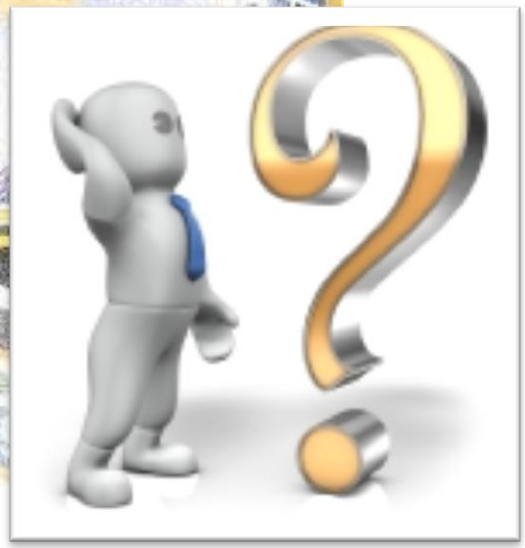
ALGORITMO: Fases del Diseño

Algoritmo

ALGORITMO: Definición del Problema

FIEE - 2014

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

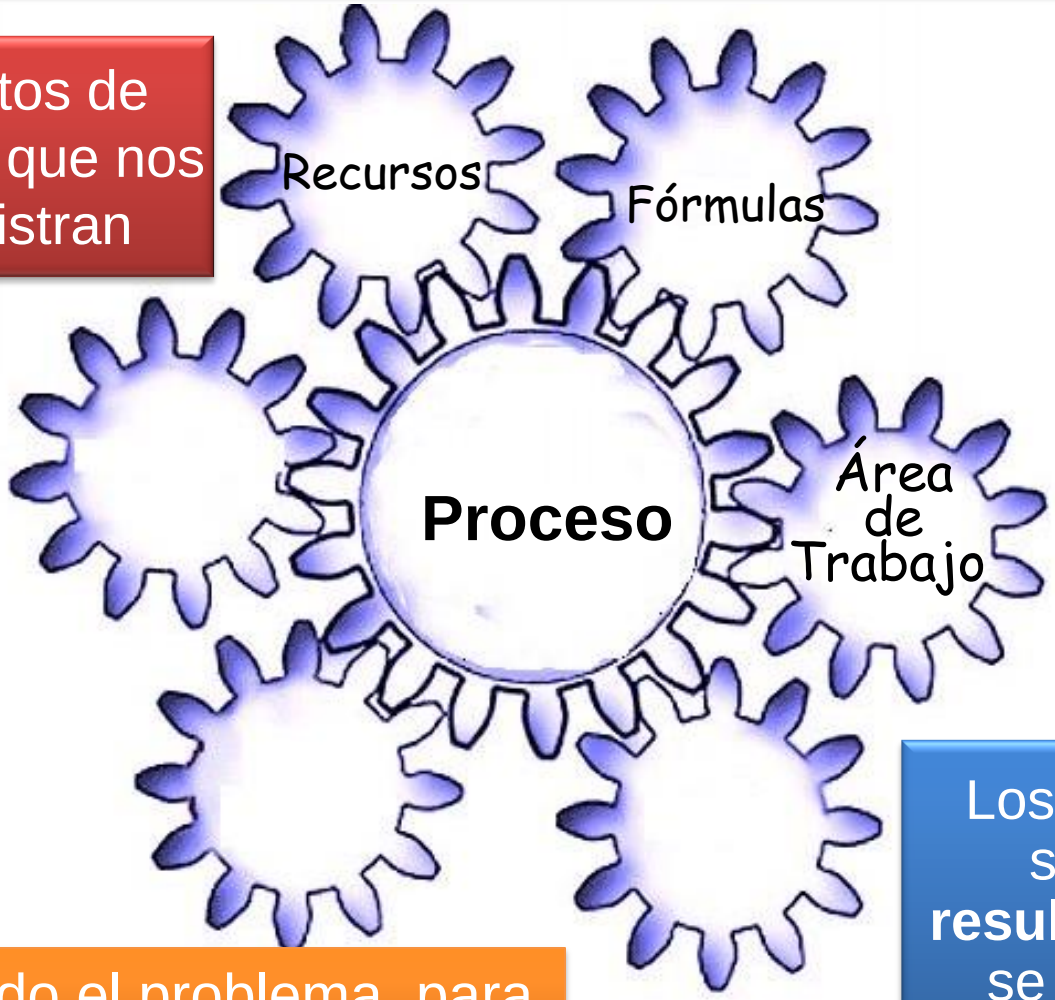


Es importante que conozcamos exactamente que se desea.

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

ALGORITMO: Análisis del Problema

Los datos de **entrada** que nos suministran



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

Entendido el problema para resolverlo es preciso analizar

FIEE - 2014

ALGORITMO: Selección de Alternativa

Se debe tener en cuenta el principio de que las cosas siempre se podrán hacer de una mejor forma.

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

FIEE - 2014

FIEE - 2014

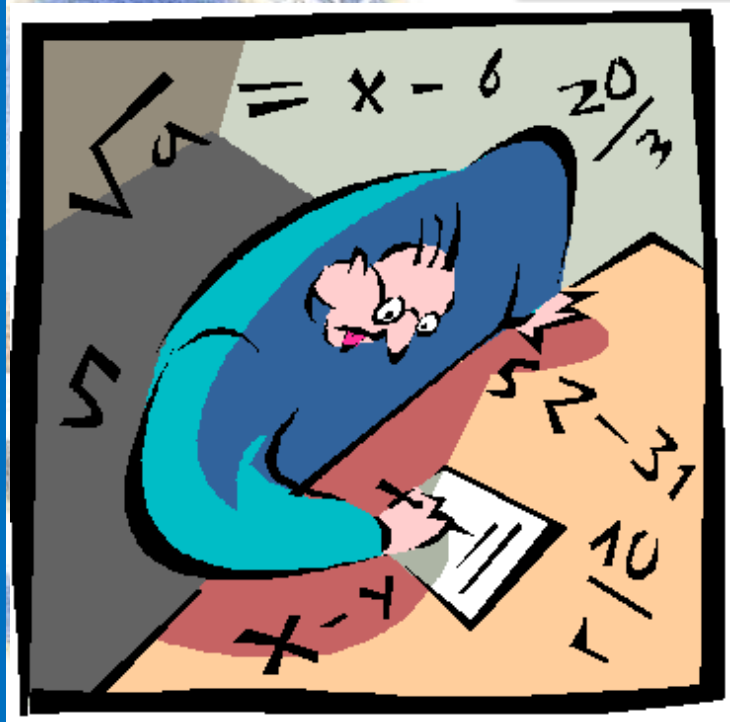
ALGORITMO: Diagramacion

Una vez que sabemos cómo resolver el problema

Dibujar gráficamente la lógica de la alternativa seleccionada

Plasmar la solucion mediante el Pseudocódigo

ALGORITMO: Prueba de Escritorio



Esta prueba consiste en:

Es Recomendable

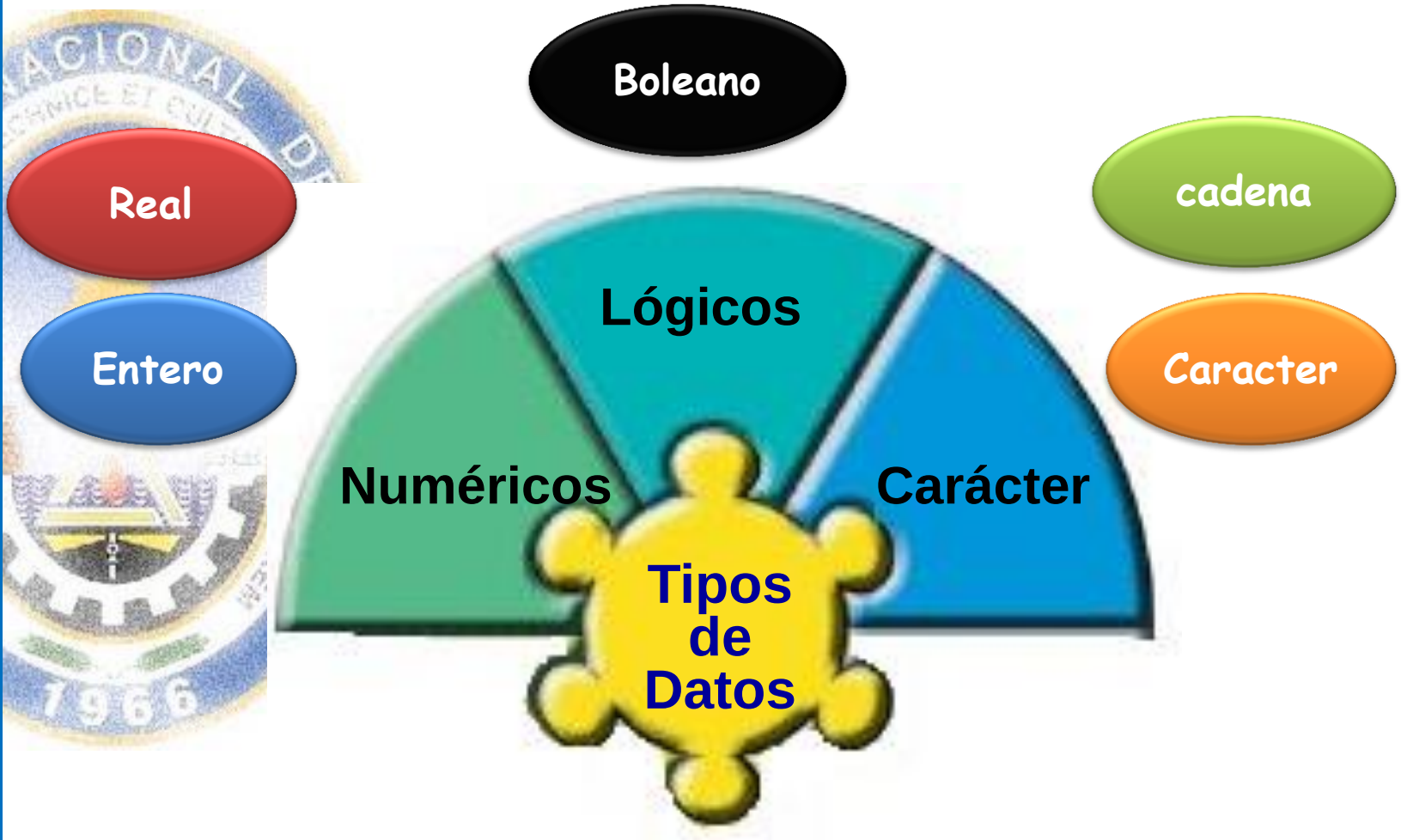
Dar diferentes datos de entrada y considerar todos los posibles casos, aún los de excepción o no esperados, para asegurar que el programa no produzca errores en ejecución cuando se presenten estos casos.

ALGORITMO: Conceptos

Constante

Corresponde a un espacio de memoria que almacena un dato que dentro del programa en ejecución mantiene siempre su contenido (valor).

ALGORITMO: Tipos de Datos



FIEE - 2014

ALGORITMO: Operadores

FIEE - 2014

+
-
/
^
%

Aritméticos

Ejemplo: Calcular el Área de un triangulo dado sus tres lados

- ✓ Para esto usaremos las variables a,b,c que representarán la medida de la longitud de cada lados del triángulo.
- ✓ La variable A para el area y
- ✓ P para su semiperimetro

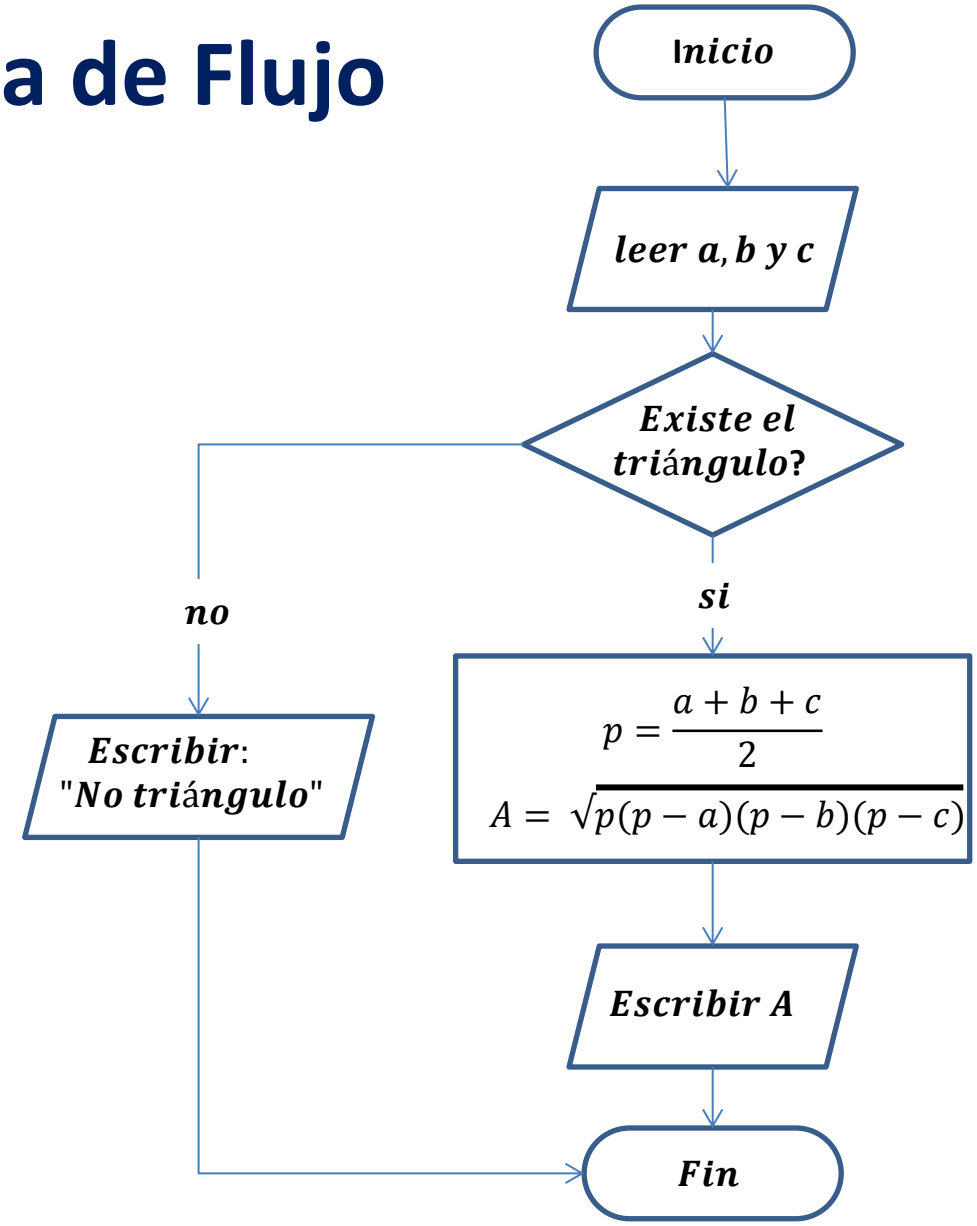
$$A = \sqrt{p(p-a)(p-b)(p-c)}$$

Pseudocodigos

```
leer a
leer b
leer c
Si el triangulo existe
    calcular p
    calcular A
sino
mostrar el mensaje “El triangulo no existe”
Fin de Si
```

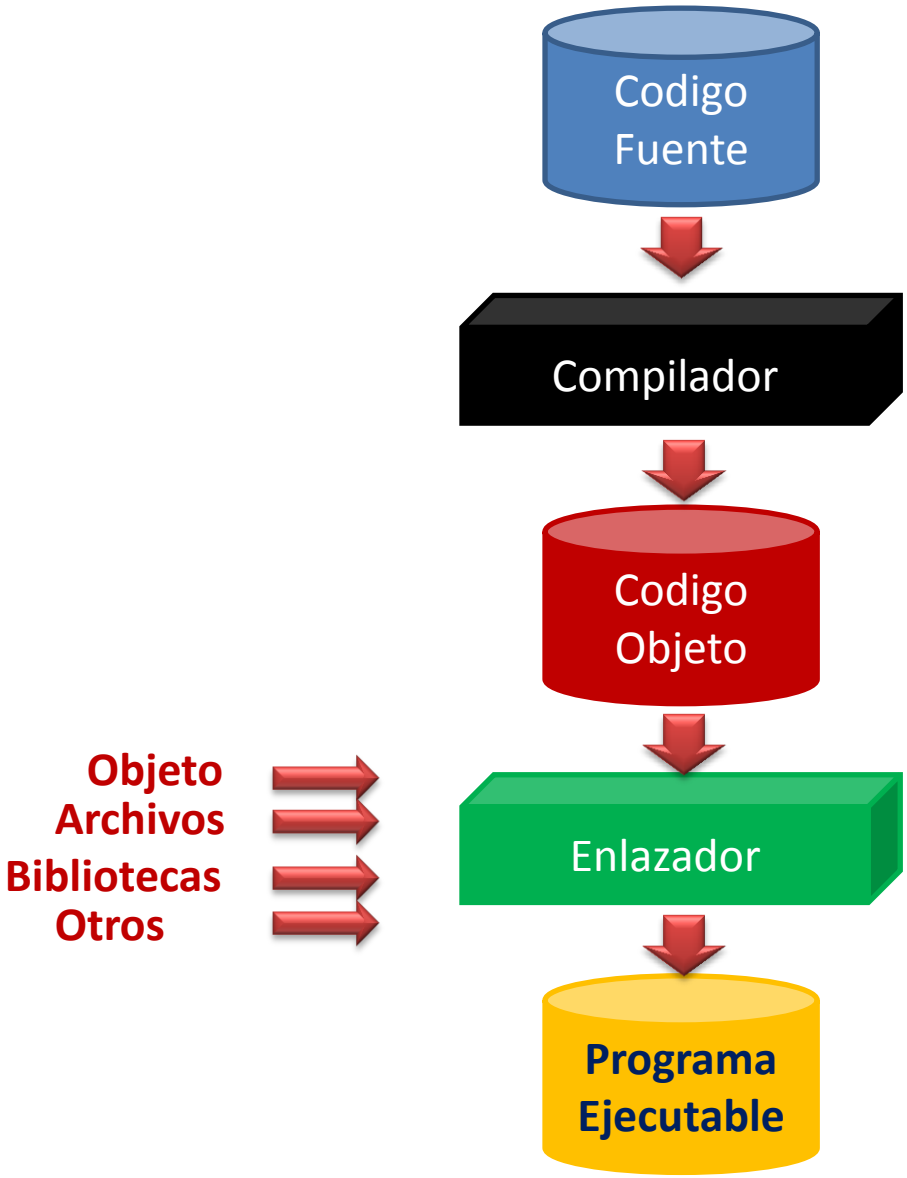
FIEE - 2014

Diagrama de Flujo



FIEE - 2014

Escritura de un Programa en C++



Elementos del lenguaje

- ✓ Un programa en C++ es una secuencia de caracteres que se agrupan en componentes léxicos (elementos del lenguaje) que comprenden el vocabulario básico del lenguaje.
- ✓ A estos componentes léxicos se les denomina “tokens”
  - ❑ los identificadores.
  - ❑ Las palabras reservadas. (p.claves)
  - ❑ Las constantes
  - ❑ Las constantes de cadena
  - ❑ Los operadores y
  - ❑ Los signos de puntuación.

Identificadores

- ✓ Son secuencias de caracteres: letras, digitos y subrayados (_).
- ✓ Que sirven para dar nombres a las variables, constantes, tipos, funciones, clases,y etiquetas de un programa.

1. Un identificador es una secuencia arbitraria de letras y dígitos.

   nombres
   direccion
   Telefono

2. El primer caracter del identificador debe ser una letra o subrayado.

   el_unico
   _fiee2012

3. El caracter subrayado (_) se trata como una letra.

Identificadores

4. Los identificadores que comienzan con dos subrayados estan reservadas para el uso interno del compilador C++.

   __MSDOS__
   __BORLANDC__

5. Las letras mayúsculas y minúsculas se tratan como caracteres diferentes.

   uni
   UNI
   unac2012

6. Un identificador definido por el usuario no puede tener el mismo nombre de una palabra reservada.

FIEE - 2014

Palabras Reservadas

1. Son identificadores predefinidos que tienen un significado especial para el compilador C++.

asn	continue	float	new	signed	try
auto	default	for	operator	sizeof	typedef
break	delete	friend	private	static	union
case	do	goto	protected	struct	unsigned
catch	double	if	public	switch	virtual
char	else	include	register	template	void
class	enum	int	return	this	volatile
const	extern	long	short	throw	while

Constantes

1. Una constante es un valor que no cambia en un programa durante su ejecucion, este es fijado por el compilador.

- ❑ Constante literal
- ❑ Constantes definidas
- ❑ Constantes enumeradas
- ❑ Constantes declaradas.

Constantes literales

Se clasifican el 4 grupos:

- ✓ Constantes enteras.
  - 123// formato decimal
  - 0264//formato octal
  - 0xBE3F//formato hexadecimal
  - 1132U// constante entera unsigned int (U)
  - 1302//constante entera de tipo long (L)
- ✓ Constantes reales
  - 13.23 7773.2233123 3.141592654 .23
  - 1.24E6 notación cientifica del numero 1.24×10^6
- ✓ Constantes de un solo caracter
  - 'M' 'E' '\n' '\r'
- ✓ Constantes de caracteres (cadena).
  - "FIEE 2012" tiene 9 caracteres.
  - "unac 2012"

Constantes Definidas (simbólicas)

Son identificadores asociados a una cadena o valor mediante directivas del procesador #define.
A estos identificadores se les denomina macros.
La forma general de la directiva es:

```
#define nombre_macro valor
```

```
#define CIERTO 1
#define FALSO 0
#define pi 3.141592
```

- ✓ El compilador C++ sustituye los valores 1, 0 y 3.141592 cuando encuentre las macros CIERTO, FALSO y pi respectivamente

```
cout<< "El valor de pi es" <<pi;
```

Mostrara en la pantalla

El valor de pi es 3.141592

Constantes Enumeradas

Son un conjunto de constantes enteras enumeradas en orden ascendente (0,1,…) asociadas cada una a un identificador en el orden que son puestas, con la finalidad de crear listas de elemtos afines.

enum *nombre_enumeracion{lista_de_enumeracion} lista_de_variables*

- ✓ **El siguiente fragmento de codigo define una enumeracion llamada dias y declara ultimodia del tipo dias.**

```
enum dias{lunes,martes,miercoles,jueves,viernes,sabado,domingo};
//definicion del tipo enumerado dias
dias ultimodia;
//declaracion de la variable ultimodia del tipo dias
```

```
ultimodia = domingo;
//ultimodia toma el valor entero 6
```

Constantes Declaradas

Tipo const

Permiten especificar valores que no pueden ser cambiados durante la ejecucion del programa.

const *tipo nombre=valor*

```
const int Semana=7;
const char curso[]="Lenguaje de Programación C++"
const float e=2.718282;
```

Tipo volatile

Permiten especificar un valor que puede cambiar por medios no explicitamente especificados por el programa, es decir por ninguna sentencia. Su uso se da cuando alguna optimizacion puede cambiar el orden de evaluacion de una expresion durante el proceso de compilacion.

Const volatile *tipo nombre=valor*

Ejemplo

```
#include <iostream>
int main(){
    using namespace std;

    cout<< "Hola FIEE 2014B..."
    cout<< endl;
    cout<< "Bienvenidos al curso LP..."<< endl;

    system ("pause")
    return 0;
}
```