Lógica de Programación

Introducción Algoritmos

Prof. Luis Gerardo Peña Camacho

Algoritmos
Introducción

- Actualmente el uso de la computadora se hace más común
- Son utilizados por las personas en sus actividades diarias. ¡Tú, por ejemplo, la utilizas!
- La mayoría de las veces lo hacemos a nivel de aplicación, esto es presionando en los íconos o botones, deslizando algunos objetos por lectores de códigos de barras o presionando pantallas sensibles al tacto.

Algoritmos

Introducción

- Pero ¿no sería agradable que pudieras desarrollar tus propios programas para dar solución a tus necesidades?
- ¿poder construir tus propios videojuegos?
- ¿generar aplicaciones que te ayuden en tus tareas?
- ¿Cómo puede lograrlo? conociendo la forma básica para realizar la planeación de cualquier aplicación, así como el lenguaje universal para su diseño.

Algoritmos
Introducción

- Por ejemplo, para construir una casa, un arquitecto necesita tener los planos de cómo desea que quede construida, de igual forma un programador requiere tener la estructura del programa que busca crear y es a través de la construcción de algoritmos y diagramas como se pueden generar los “planos” que faciliten la solución del problema.

Algoritmos

- ¿Qué debemos saber?
  - ¿Qué es un problema o situación cotidiana?
  - ¿qué es un algoritmo y cuáles son sus características?
  - ¿qué es un diagrama, cómo se construye y qué representa cada una de las figuras?

Algoritmos

- ¿Qué debemos saber?
  - ¿Qué es un problema o situación cotidiana?
  - ¿qué es un algoritmo y cuáles son sus características?
  - ¿qué es un diagrama, cómo se construye y qué representa cada una de las figuras?

Algoritmos
Conceptos Básicos

- Un algoritmo es una serie de instrucciones que colocadas en un orden lógico conducen a la solución de un problema.
- También se puede decir que un algoritmo es la fase preliminar al escribir un programa en cualquier lenguaje de programación.
- El programador de computadoras es ante que nada una persona que resuelve problemas, por lo que para llegar a ser un programador eficaz se necesita aprender a resolver problemas de un modo lógico, riguroso y sistemático.

Algoritmos
Pasos para la solución de Problemas

1. **Diseño de algoritmo**, que describe la secuencia ordenada de pasos que conducen a la solución.
2. **Solución de un problema dado**. (Análisis del problema y desarrollo del algoritmo). Expresar el algoritmo como un programa de lenguaje de programación adecuado. (Fase de codificación.)
3. **Ejecución y validación** del programa por la computadora.

Algoritmos
Técnicas

- Primero analiza el problema, se muestra o presenta una solución lógica o matemática, luego identifica sus procesos ya sea de Entrada, Proceso, Salida y luego plasma esos procesos en un algoritmo.

Algoritmos
Técnicas

Por ejemplo:

- Elaborar un programa que realice la suma de dos valores enteros.

  ¿Que preguntas nos debemos hacer? p.e ¿Qué datos voy a ingresar?, ¿Qué dato será mi resultado? y

  ¿Cómo consigo ese resultado?

  La respuesta sería: que deberás ingresar dos valores numéricos enteros, que el resultado sería la suma de esos dos valores y el proceso para conseguirlo sería la suma del numero 1 más el numero 2. Luego de ello solo tendrías que plasmarlo en un algoritmo.

Algoritmos
Paso para diseñar algoritmos

1. Análisis
2. Escribir el algoritmo. Diseño.
3. Prueba

Algoritmos
Paso para diseñar algoritmos

1. **Análisis**

   Es sumamente importante hacer un buen análisis de cual es específicamente el problema a resolver. Para esto es bueno ayudarse mediante gráficos del problema o en caso de que no sea graficable, también se puede resolver el problema para casos específicos y luego generalizarlo para todos los posibles casos.

   También se deben observar cuales serían los casos especiales, es decir, aquellos casos que no cumplan la norma general, y tratar de evaluarlos de otra forma.

   Este paso es el que más tiempo debe llevarle a un buen programador, ya que de un buen análisis depende los buenos resultados que arroje el algoritmo.

Algoritmos
Paso para diseñar algoritmos

2. Escribir el algoritmo. Diseño.

Después de haber analizado el problema en una forma abstracta, se debe llevar al papel, mediante instrucciones adecuadas al análisis.

Si el problema fue bien analizado, este paso es muy rápido a comparación del anterior.

Algoritmos
Paso para diseñar algoritmos

3. Prueba

Este paso es opcional y se aplica siguiendo paso por paso las instrucciones del algoritmo, notando los diferentes valores que van tomando las variables, de forma que se pueda verificar si hay errores en alguna instrucción (debug).

Obviamente, éste método es muy engorroso para algoritmos muy extensos, por lo que en estos casos no sería aplicable.

Algoritmos

TIPOS DE DIAGRAMACIÓN ALGORÍTMICA

- En general, hay varias formas de escribir un algoritmo, pero explicaremos dos de las técnicas más conocidas: *Diagramas de Flujo (DFD)* y *Pseudocodificación*.

Algoritmos

TIPOS DE DIAGRAMACIÓN ALGORÍTMICA

- **DIAGRAMA DE FLUJO - DFD**

  Los Diagramas de Flujo constan de figuras que encierran las instrucciones y líneas que apuntan hacia la siguiente instrucción de modo que no se pierda el orden.

  Además, las operaciones elementales como la entrada, el proceso y la salida de los datos se reconocen de las demás instrucciones porque no se encierran en rectángulos sino en otra clase de figuras, por lo que según la figura en la que esté encerrada la instrucción se reconocerá su significado.

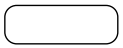
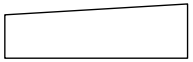
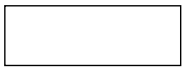
  Los diagramas de flujo son muy didácticos, por lo que son muy fácil de entender.

Algoritmos

TIPOS DE DIAGRAMACIÓN ALGORÍTMICA

- **DIAGRAMA DE FLUJO - DFD**

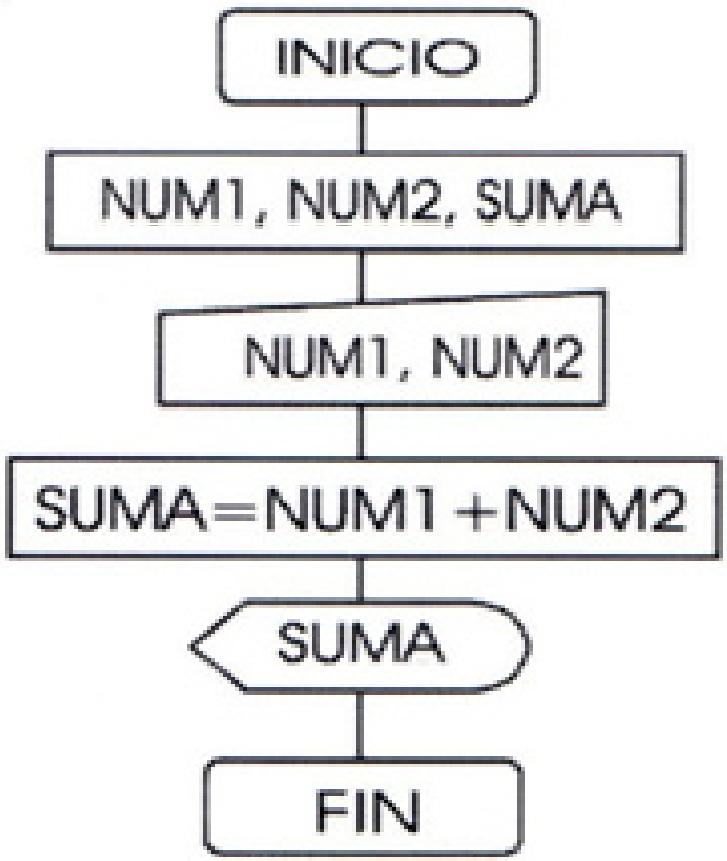
SIMBOLOGIA UTILIZADA EN EL DIAGRAMA EPS	
SIMBOLOGIA	SIGNIFICADO
	Entrada/Salida. Datos de entrada y resultado
	Proceso. Operaciones para obtener el resultado esperado
	Líneas de flujo. Indican la secuencia del flujo de operación

SIMBOLOGIA USADA EN LOS DIAGRAMAS DE FLUJO	
Símbolo	Significado
	Inicio/Fin. Determina el inicio y fin del algoritmo
	Entrada por teclado. Representa el ingreso de los datos al programa
	Proceso. Representa las operaciones que se efectúan para obtener el resultado.
	Decisión. Representa las operaciones de tipo lógico que contenga el algoritmo
	Salida por impresora. Se utiliza cuando se desea obtener el resultado en papel.
	Salida por pantalla. Se utiliza cuando solamente se va a mostrar el resultado en pantalla.
	Conector. Se utiliza para conectar bloques del diagrama cuando el diagrama es grande y es necesario dividirlo.
	Líneas de flujo. Indican la secuencia del flujo de operación del diagrama.

Algoritmos

TIPOS DE DIAGRAMACIÓN ALGORÍTMICA

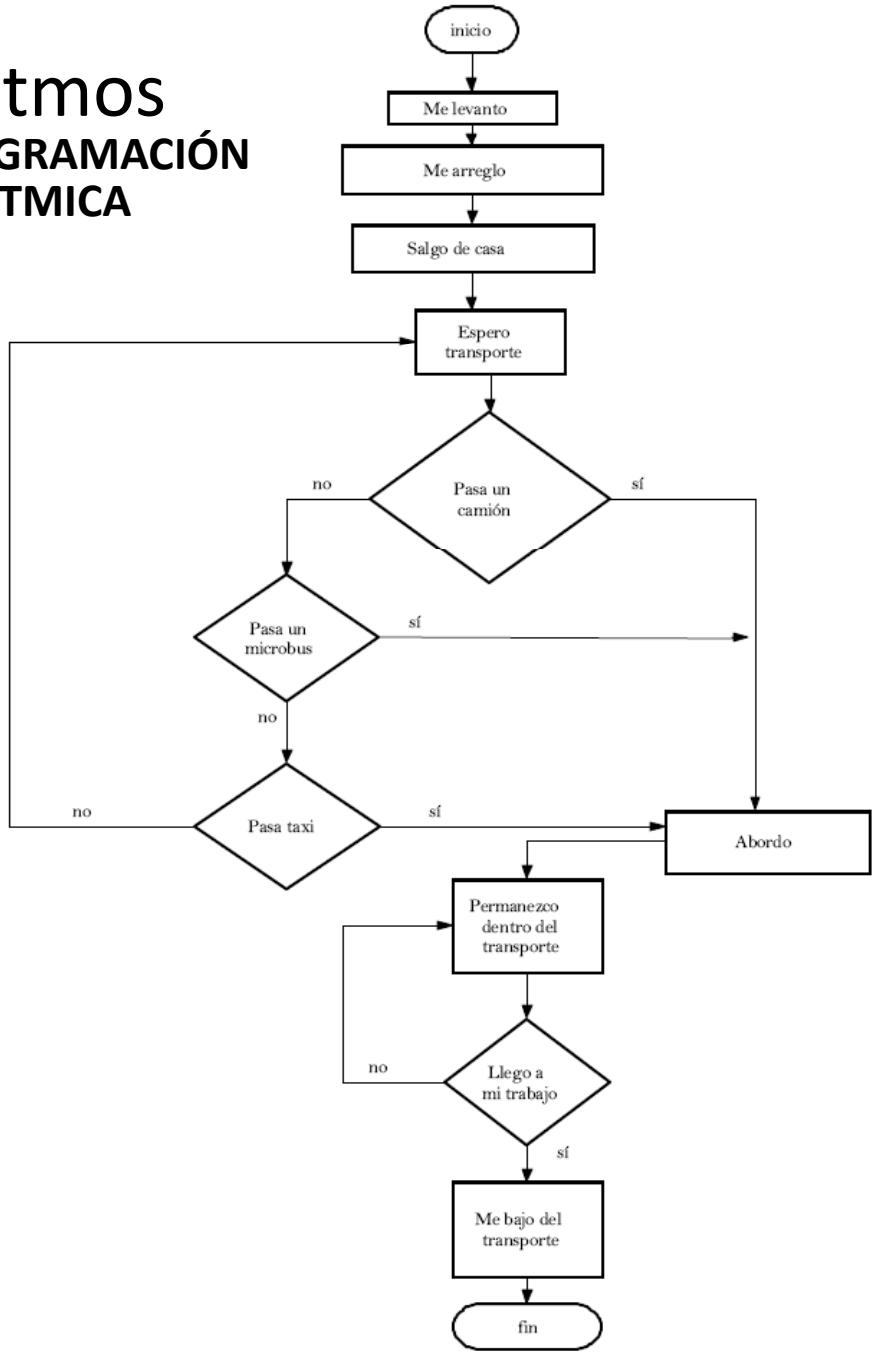
- **DIAGRAMA DE FLUJO**
  EJEMPLO:
  Hacer un programa que me muestre la suma de dos valores**.**



Algoritmos

TIPOS DE DIAGRAMACIÓN ALGORÍTMICA

- **DIAGRAMA DE FLUJO EJEMPLO:**



Algoritmos

TIPOS DE DIAGRAMACIÓN ALGORÍTMICA

- **PSEUDOCODIGO:**

  En el pseudocodigo, cada instrucción es una línea y las operaciones elementales se escriben de una forma tan clara que será imposible que algún programador no las entienda.

  Aquí utilizaremos un pseudocodigo un poco personalizado, para hacer más sencilla la explicación.

  Para realizar dichas instrucciones, se utilizan básicamente Variables y Operadores.

Algoritmos

TIPOS DE DIAGRAMACIÓN ALGORÍTMICA

- **PSEUDOCODIGO**

  EJEMPLO:
  Hacer un programa que me muestre la suma de dos valores.

```
INICIO

NUM1, NUM2, SUMA

Leer <-- NUM1
Leer <-- NUM2
SUMA = NUM1+NUM2
Mostrar --> SUMA

FIN
```

Algoritmos

VARIABLES

- Las variables son espacios que se reservan o separan en la memoria para almacenar cualquier tipo de información (números, letras, frases, etc.), las variables pueden tener cualquier nombre pero preferiblemente deben ser mnemotécnico, es decir, que tenga relación con él o los datos que se almacenarán en la variable.

“Declarar una variable es definir que tipo de datos se van a almacenar en dicha variable.”

Algunas reglas:

1. No pueden comenzar con números.
2. No pueden contener símbolos u operadores en ninguna posición del nombre, excepto el carácter “_”.
3. No pueden contener espacios.

Algoritmos
VARIABLES

- **Ejemplos**

- A : Es un nombre correcto, aunque posiblemente muy poco mnemotécnico.
- A23: correcto.
- OJO5MALO: correcto.
- HOLA#: incorrecto, ya que tiene un símbolo dentro de su nombre.
- A _23 : incorrecto, porque hay un espacio dentro del nombre.
- A-23 : incorrecto lleva un símbolo menos.
- A_23 : correcto, ya que el carácter "_" es el único símbolo permitido para nombrar variables.

Ejemplos Específicos:

- Si necesitamos una variable para guardar el sueldo de una persona, usaremos: sueldo_Per ó sueldoPer
- En este otro caso guardaremos el nombre de un trabajador, usaremos: nom_Trab ó nomTrab ó nomTRAB.

Algoritmos

CONSTANTES

- Las constantes tienen un valor fijo (su contenido nunca cambia), se le da cuando se define la constante y que ya no puede ser modificado durante la ejecución de cualquier programa.

  Si se necesita incluir en el algoritmo alguna constante, se debe seguir la misma metodología de la asignación de variables, pero con la limitación que el identificador debe estar escrito con letras mayúsculas y que la declaración se haga a continuación del inicio del algoritmo.

Algoritmos
CONSTANTES

- Ejemplo:

- Ced <-- ‘18808481'
- Institucion <-- ‘Universidad de Los Andes‘

Recuerda las constantes también son variables (espacios reservados en memoria), la única diferencia que el valor que contienen una constante no cambia nunca.

Algoritmos

Tipos de Operadores

- **Operadores aritméticos**
- **Operadores de comparación**

Algoritmos

Tipos de Operadores

- **Operadores aritméticos**
- Nos permiten realizar operaciones numéricas con nuestras variables.

+	→ Suma
-	→ Resta
*	→ Multiplicaicón
/	→ División
%	→ devuelve el residuo de la división (resto)
^	→ Potenciación

Ejemplo: **RES = NUM1** + **NUM2**

Algoritmos

Tipos de Operadores

- **Operadores aritméticos**

  Las operaciones en un computador tienen un cierto orden de importancia, es decir, algunas se efectúan primero que otras. Este orden en forma descendente es: Potenciación, modulo, división, multiplicación, suma y resta.

  Si no se coloca ningún paréntesis en la operación aritmética, el orden anterior es el que tiene la máquina para realizar las operaciones, pero al colocar algo entre paréntesis, esa parte de operación primará sobre las demás.

Algoritmos

Tipos de Operadores

- **Operadores de comparación**

  Se utilizan principalmente en nuestras condiciones para comparar dos variables y verificar si cumple o no la propiedad del operador, este retornara un valor lógico como Verdadero o Falso.

  = → Igualdad

  <> → Diferente

  < → Menor que

  <= → Menor igual que

  > → Mayor que

  >= → mayor igual que

Algoritmos

Tipos de Operadores

- **Operadores de comparación**

EJEMPLO:

```
Si NOTA >= 10.5 Entonces
  Mostrar "APROBADO"
Sino
  Mostrar "DESAPROBADO"
Fin Si
```

- **OPERADORES LÓGICO**

  Se usan en combinación con los operadores de comparación cuando la expresión de la condición lo requiere.

  AND → Y
  OR → O
  NOT → No

Algoritmos

Tipos de Operadores

- **Operadores de comparación**

EJEMPLO:

```
Si (EDAD > 13) AND (EDAD < 30) Entonces
  Mostrar "JOVEN"
Sino
  Mostrar "NOS ES JOVEN"
Fin Si
```

Algoritmos

Diseño de formulas

- Las operaciones en un computador tienen un cierto orden de importancia, es decir, algunas se efectúan primero que otras.
- **Orden en forma descendiente:**

  División, multiplicación, suma y resta.

  Si no se coloca ningún paréntesis en la operación aritmética, el orden anterior es el que tiene la máquina para realizar las operaciones, pero al colocar algo entre paréntesis, esa parte de operación primará sobre las demás.

Algoritmos

Diseño de formulas

- Ejemplos:
- **1. 5 x 2 + 6 =** Se desglosa de la siguiente manera:((5 * 2) + 6)
-
- **2. 8 + 5 – 3 + 5 * 2 =** Se desglosa de la siguiente manera: (8+5) - (3+(5*2))

Algoritmos

Diseño de formulas

- Ejemplos:

$\frac{5x2}{2}$ = Se desglosa de la siguiente manera:((5 * 2) / 2

$\left(\frac{\left(\frac{6}{2}\right)+((5\times 2)+(3\times 2))}{2}\right)$ = Se desglosa así:

(6/2) + (((5*2) + (3*2)) /2)

Algoritmos

Contadores

- Un contador es aquel que permite incrementar el valor de una variable numérica, de uno en uno, de dos en dos, etc. (esto se aplica también de forma negativa)
-
- EJEMPLO: CONT = CONT + 1
- (Si esto se utiliza dentro de un bucle y/o estructura repetitiva, incrementará la variable CONT de uno en uno).

Algoritmos

Programación Modular

- En algunas ocasiones se debe llamar un bloque de código más de una vez, una forma de hacerlo es colocar las instrucciones en subprogramas que se invocan cada vez que se necesiten.
- Los procedimientos y funciones que se declaran son subprogramas que son llamados de un programa principal, un procedimiento o función puede contener otros procedimientos y funciones estos son escensialmente partes separadas de código que ejecutan tareas pequeñas de un programa GRANDE.

Algoritmos

Programación Modular - Procedimientos

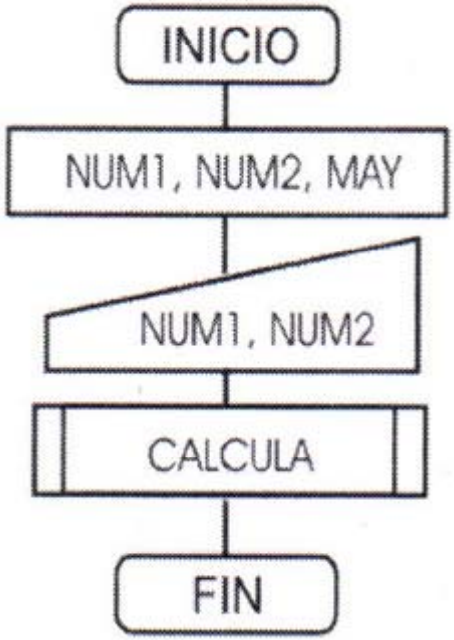
- **PROCEDIMIENTOS**

  Un procedimiento es un sub algoritmo que realiza una tarea concreta. En lugar de reescribir el código completo de esa tarea cada vez que se necesite, únicamente se hace una referencia al procedimiento.

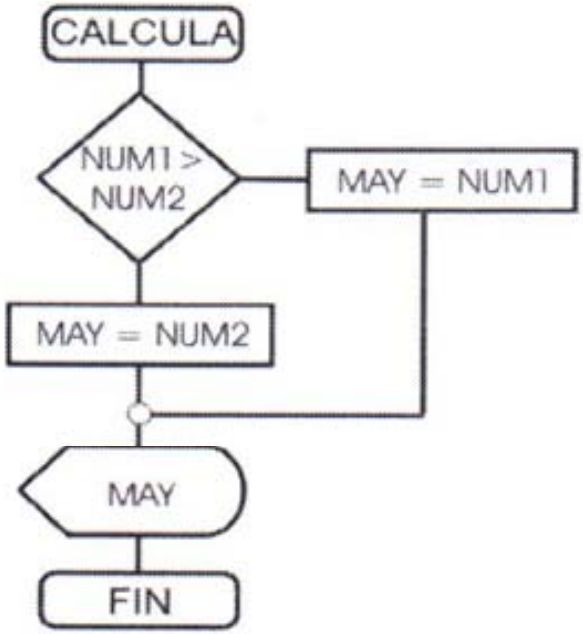
Algoritmos

Programación Modular - Procedimientos

- EJEMPLO: Elaborar un programa que calcule el mayor de 2 números usando procedimientos.



Algoritmo Principal



Procedimiento Calcula

Algoritmos

Programación Modular - Funciones

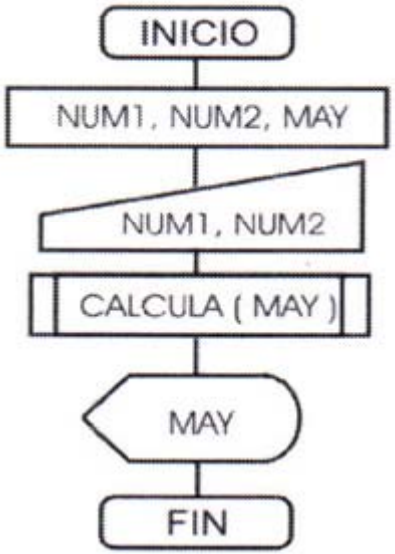
- **Funciones**

  Las funciones son, al igual que los procedimientos, un conjunto de sentencias que se ejecutan constantemente, la diferencia entre éstas y los procedimientos es que las funciones regresan un valor.

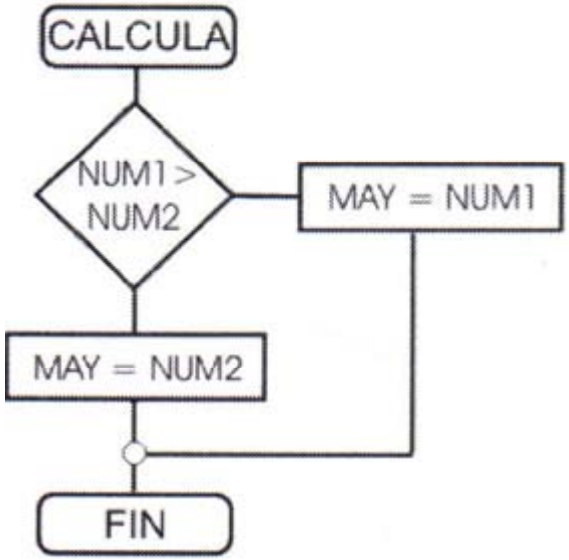
Algoritmos

Programación Modular - Procedimientos

- EJEMPLO: Elaborar un programa que calcule el mayor de 2 números usando funciones.



Algoritmo Principal



Función Calcula

Algoritmos

Referencias

- Joyanes, Luis. *Fundamentos de Programación, Algoritmos y Estructuras de Datos.* Mexico, Mc Graw Hill, 2000.
- MATA-TOLEDO, RAMÓN Y CUSHMAN PAULINE. Introducción a la programación. Serie Schaum. Mc Graw-Hill. 2001.
- Wikipedia Articulo: http://es.wikipedia.org/wiki/Algoritmos
- María Alejandra Quintero.

  Introducción a la Programación con aplicaciones en Visual Basic. http://www.forest.ula.ve/~mariaq/IntroProgVB.pdf
- **Herramientas de Diagrama de Flujo**
  - http://es.scribd.com/doc/4669112/MANUAL-DFD-MEJORADO
  - http://projects.gnome.org/dia/docs.html
  - http://www.abcdatos.com/tutoriales/tutorial/v1538.html