

DIAGRAMAS DE FLUJO

Tabla de contenido

1. INTRODUCCIÓN.....	1
2. ¿Qué es un DIAGRAMA DE FLUJO?.....	2
3. SIMBOLOGÍA.....	2
4. CONSIDERACIONES.....	5
5. SÍMBOLOS COMPUESTOS.....	6
6. REPRESENTACIÓN DE UN DIAGRAMA EN PSEINT.....	7
7. EJEMPLO EN PSEUDOCÓDIGO.....	9
8. REFERENCIAS.....	13

Nota: La información teórica que se presenta entre las secciones 2 y 5 fue obtenida de la referencia [número 1](#).

1. INTRODUCCIÓN

El presente documento tiene pretende servir de guía para la docencia en la asignatura de programación básica. Pese a que su desarrollo se ciñe a los lineamientos establecidos en el syllabus de la asignatura transversal de programación básica de la universidad distrital, es útil en cualquier ámbito en el que se requieran conceptos y fundamentos básicos de programación.

2. ¿Qué es un DIAGRAMA DE FLUJO?

Un diagrama de flujo es la representación gráfica de un proceso. Un diagrama de flujo tiene como objetivo facilitarnos la comprensión de un algoritmo o simplificar el análisis de un proceso. El diagrama de flujo consta de símbolos como cuadros, rombos, óvalos, etc. que representan a cada uno de los pasos a seguir durante un proceso. Estos símbolos están conectados por flechas de un solo sentido y nos indican la secuencia en que se van desarrollando las distintas tareas.

Los diagramas de flujo se desarrollaron con la idea de representar procesos en áreas como la administración de empresas, la planeación de proyectos, la manufactura de productos, la planificación de estrategias de ventas y muchas áreas más, no obstante, han cobrado un interés muy especial en el análisis y planeación de procesos para el desarrollo de sistemas computacionales. Y al mismo tiempo han demostrado tener una gran utilidad en el proceso de aprendizaje y formación de futuros desarrolladores de software.

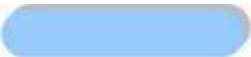




Los diagramas facilitan la comprensión de métodos de solución para problemas matemáticos y lógicos, permitiendo a los estudiantes abstraerse de un lenguaje determinado y enfocarse en el procedimiento de solución. De este modo, la posterior etapa de codificación se simplifica enormemente al tener ya resuelta la metodología a través de un diagrama de flujo.

DIAGRAMAS DE FLUJO



3. SIMBOLOGÍA

Existen diferentes símbolos utilizados en la diagramación, de los cuales podemos destacar cinco símbolos básicos con los cuales se puede representar prácticamente cualquier algoritmo informático por muy complejo que éste sea.



Estos símbolos son los siguientes:

Simbolo	Descripción
	<p>Inicio / Terminación. Este símbolo se utiliza para señalar el comienzo así como el final de un diagrama. Tradicionalmente se colocan las palabras "INICIO" ó "FIN" dentro de la figura para hacerlo más explícito.</p> <p>Es el único símbolo que solamente tiene una conexión (flecha) ya sea de salida, en el de inicio, o de entrada, para el de fin.</p>
 	<p>Entrada de datos. En este símbolo se indican los valores iniciales que deberá recibir el proceso. Esto se hace asignándoles letras o nombres de variables para cada uno de los valores y anotando estas letras en el interior de la figura.</p> <p>Existen otros símbolos que también representan una entrada de datos pero no consideramos que su utilización, o combinación, aporte mayor utilidad al objetivo intrínseco de ejemplificar una entrada de datos.</p> <p>Este símbolo siempre deberá tener al menos una conexión entrante (generalmente del inicio) y una de salida.</p>
	<p>Proceso de datos. Este símbolo lo utilizaremos para señalar operaciones matemáticas, aritméticas o procesos específicos que se realicen con nuestros datos.</p> <p>La manera de anotar dichos procesos, puede ser mediante una descripción breve de la operación o mediante una asignación de dicha operación hacia una variable como por ejemplo: $R \leftarrow A + B$</p> <p>Este símbolo siempre deberá tener al menos una conexión de entrada y una de salida.</p>
	<p>Decisión. Este símbolo nos representa una disyuntiva lógica o decisión. En su interior se anota una instrucción o pregunta que pueda ser evaluada como cierta o falsa y que determine el flujo del programa.</p> <p>Este símbolo es el único que puede contener dos salidas y en cada una de las salidas se suele poner un rótulo de "si/no" o "cierto/falso" indicando con esto cual de ellas se tomará según el resultado de la evaluación de la función.</p>

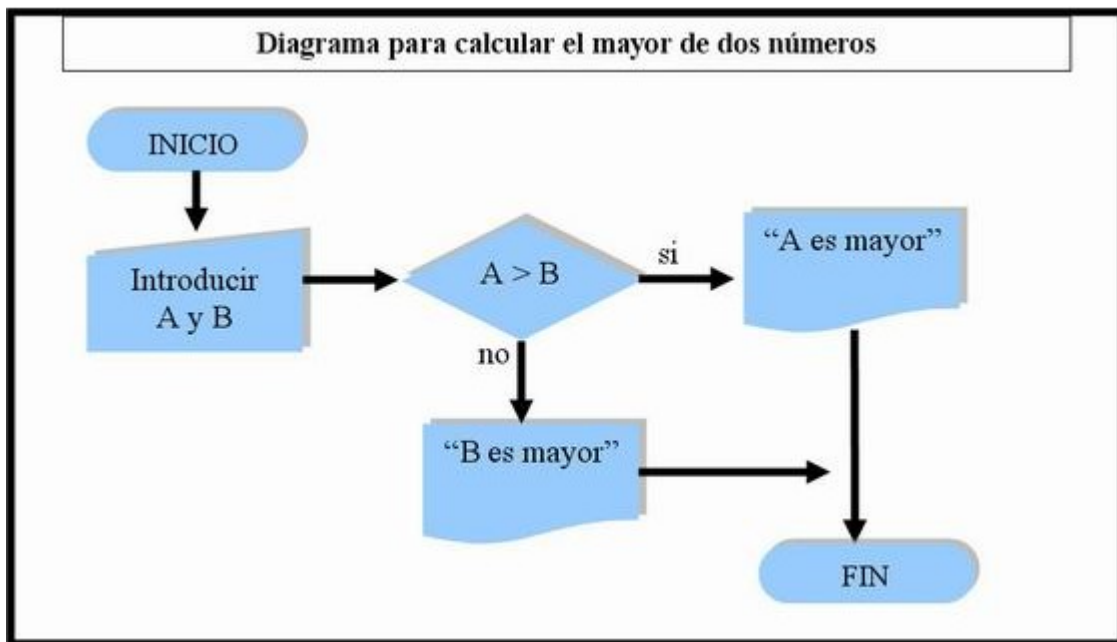
DIAGRAMAS DE FLUJO

	Es una buena práctica de diagramación utilizar siempre el mismo lado para los positivos siempre que esto sea posible.
	<p>Desplegado de información. Este símbolo se utiliza para mostrar un resultado, el cual puede representar la solución al problema que se pretende resolver y que fue conseguida a través del resto del diagrama.</p> <p>Dentro de su interior se anotará la variable con el resultado final o el mensaje que represente el resultado del algoritmo.</p>
	<p>Generalmente veremos este símbolo muy cerca del final del proceso y precedido por el símbolo de terminación.</p> <p>Este símbolo siempre deberá tener al menos una conexión de entrada y una de salida.</p>

En la diagramación, también contamos con una serie de símbolos auxiliares que no intervienen en el proceso del algoritmo, pero que pueden ser útiles para ayudarnos a dar claridad a nuestros diagramas, algunos de ellos son los siguientes:

Simbolo	Descripción
	<p>Conector. Este símbolo se utiliza para indicar un salto dentro del diagrama. Se utiliza con el propósito de facilitar la disposición plana de un diagrama y evitar el cruce excesivo de líneas a través del mismo.</p> <p>Este conector va asociado a un conector “gemelo” y junto con él, representa una puerta de entrada y de salida para el flujo del diagrama, es decir que cuando una flecha termina en un conector marcado con la letra “A”, se continuará el diagrama a partir de otro conector marcado con la misma letra tal como si se tratara de una línea continua in interrumpida.</p>
	<p>Conector de página. Este conector es idéntico en funcionamiento que el anterior, pero su forma pentagonal lo distingue y nos indica que debemos buscar el “gemelo” en una página distinta de la actual. Este conector lleva asociado una especie de salto entre páginas.</p>

4. CONSIDERACIONES

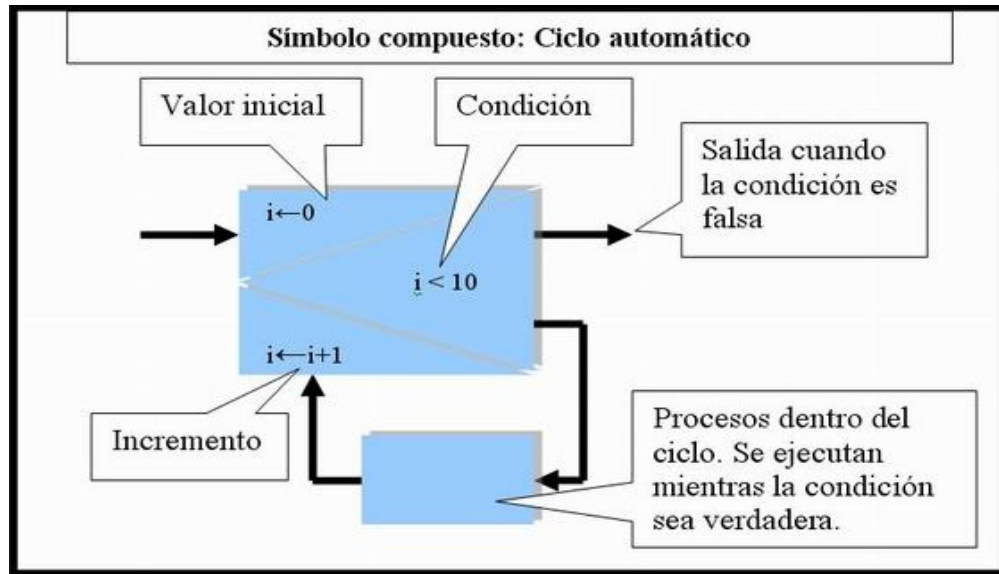


Aprovecharemos el diagrama anterior, para mencionar algunas consideraciones importantes:

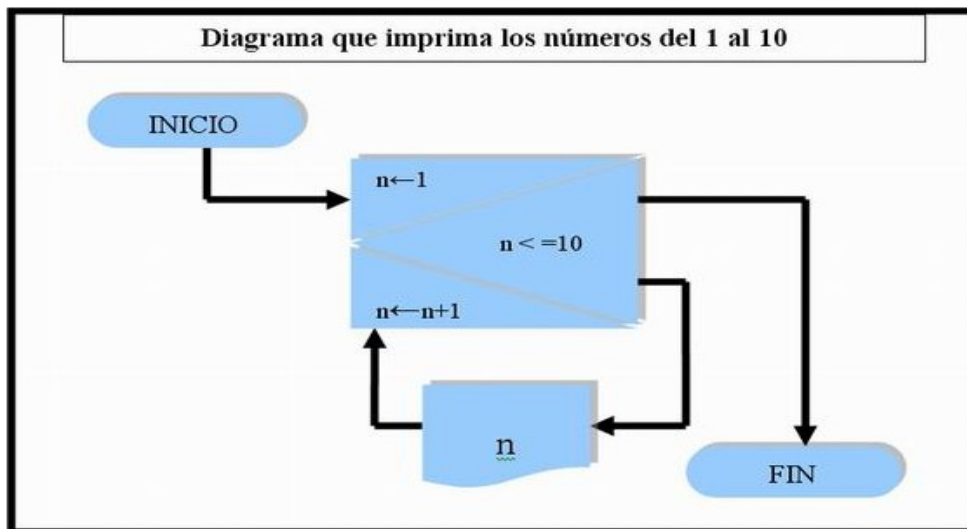
- El diagrama puede desarrollarse en cualquier dirección, sin embargo es aconsejable que el desarrollo se realice en lo posible de arriba hacia abajo y de izquierda a derecha.
- Debe procurarse que el símbolo de inicio se encuentre en la parte superior o superior-izquierda del diagrama.
- El final se debe procurar que quede en la parte inferior o inferior-derecha. Si esto no es posible, debe separarse ligeramente del cuerpo del diagrama a fin de que sea fácilmente identificado.
- Se pueden utilizar palabras para especificar la acción dentro del símbolo como es el caso de "Introducir A y B", aunque esto es innecesario ya que con poner simplemente "A, B" se sobreentiende. Es el mismo caso de poner "Comparar si $A > B$ " o simplemente " $A > B$ ".
- Es válido hacer que dos flechas apunten a un símbolo, aunque es más estético hacer que la segunda flecha apunte a la primera que si está apuntando al símbolo, tal como se hace en el caso del "FIN".

5. SÍMBOLOS COMPUESTOS

En diagramación se acostumbra también el uso de símbolos compuestos que nos reduzcan la complejidad y el tamaño del diagrama. Estos símbolos representan estructuras de uso común y que ya están adecuadamente definidos. Este es el caso del símbolo de “Ciclo automático” que es como sigue:



El símbolo del Ciclo automático representa al ciclo “for” de la mayoría de los lenguajes. Para entender mejor este símbolo, veamos el siguiente ejemplo:



DIAGRAMAS DE FLUJO

En este caso inicialicé la variable “n” con el valor de 1. Y cambié la condición de salida para que comparara con “menor o igual” (\leq) en lugar de “menor que” ($<$).

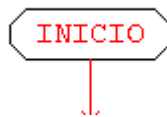
El mismo diagrama se puede también realizar con los objetos básicos, lo cual se puede dejar como ejercicio para apreciar la ventaja de utilizar símbolos compuestos

6. REPRESENTACIÓN DE UN DIAGRAMA EN PSEINT

Existen diversas herramientas para realizar diagramas de flujo, dos de las más conocidas son FreeDFD y PseInt. Esta última con la característica de poder trabajar en pseudocódigo y mantener sincronía con su respectivo diagrama de flujo. A continuación se explicará la notación en diagrama de flujo en PseInt, aunque es necesario resaltar que algunos de los símbolos difieren un poco del estándar.

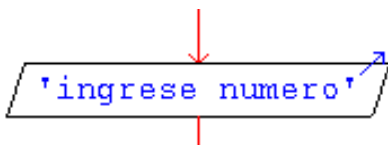
Como inicia la **ejecución**:

Ejemplo 1:



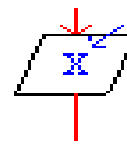
Que símbolo representa **escribir**:

Ejemplo 1:



Como se representa la **lectura (leer x)**:

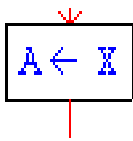
Ejemplo 1:



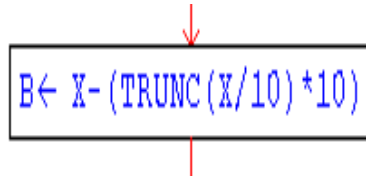
DIAGRAMAS DE FLUJO

Como se representa la **asignación**:

Ejemplo 1:



Ejemplo 2:

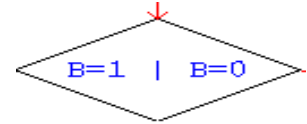


Como se representa una **comparación**:

Ejemplo 1:

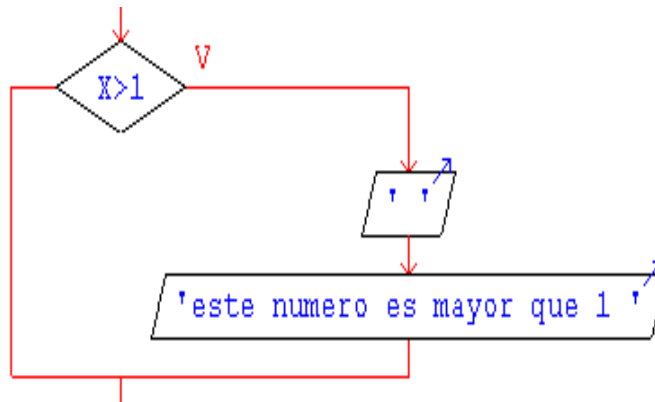


Ejemplo 2:



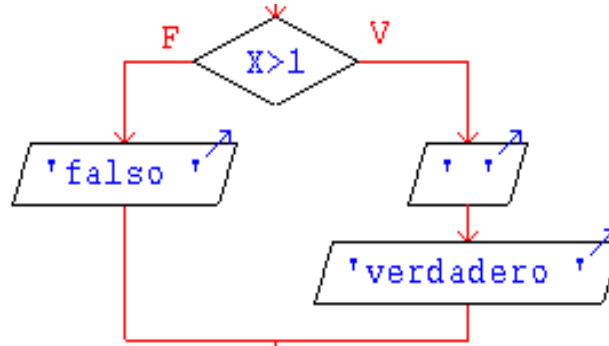
Como se representa el condicional **si**:

Ejemplo 1:



Como se representa el **si con el sino**:

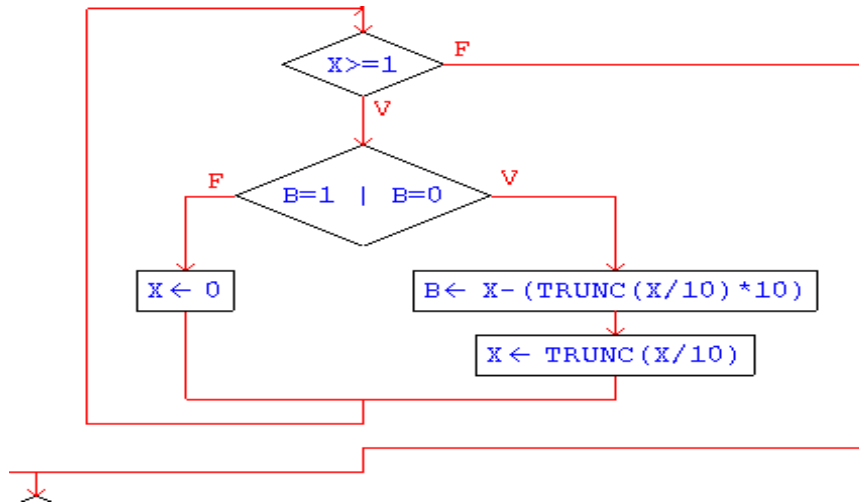
Ejemplo 1:



DIAGRAMAS DE FLUJO

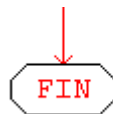
Como se representa el **mientras**:

Ejemplo 1:



Como termina la **ejecución**:

Ejemplo 1:



7. EJEMPLO EN PSEUDOCÓDIGO

A continuación se presenta un algoritmo en pseudocódigo, junto con su diagrama de flujo y su respectiva explicación. Tanto el pseudocódigo como el diagrama están creados en la herramienta Pselnt y corresponden a un aplicación que determina si un numero ingresado es binario para convertirlo a decimal o si es decimal para transformarlo en binario.

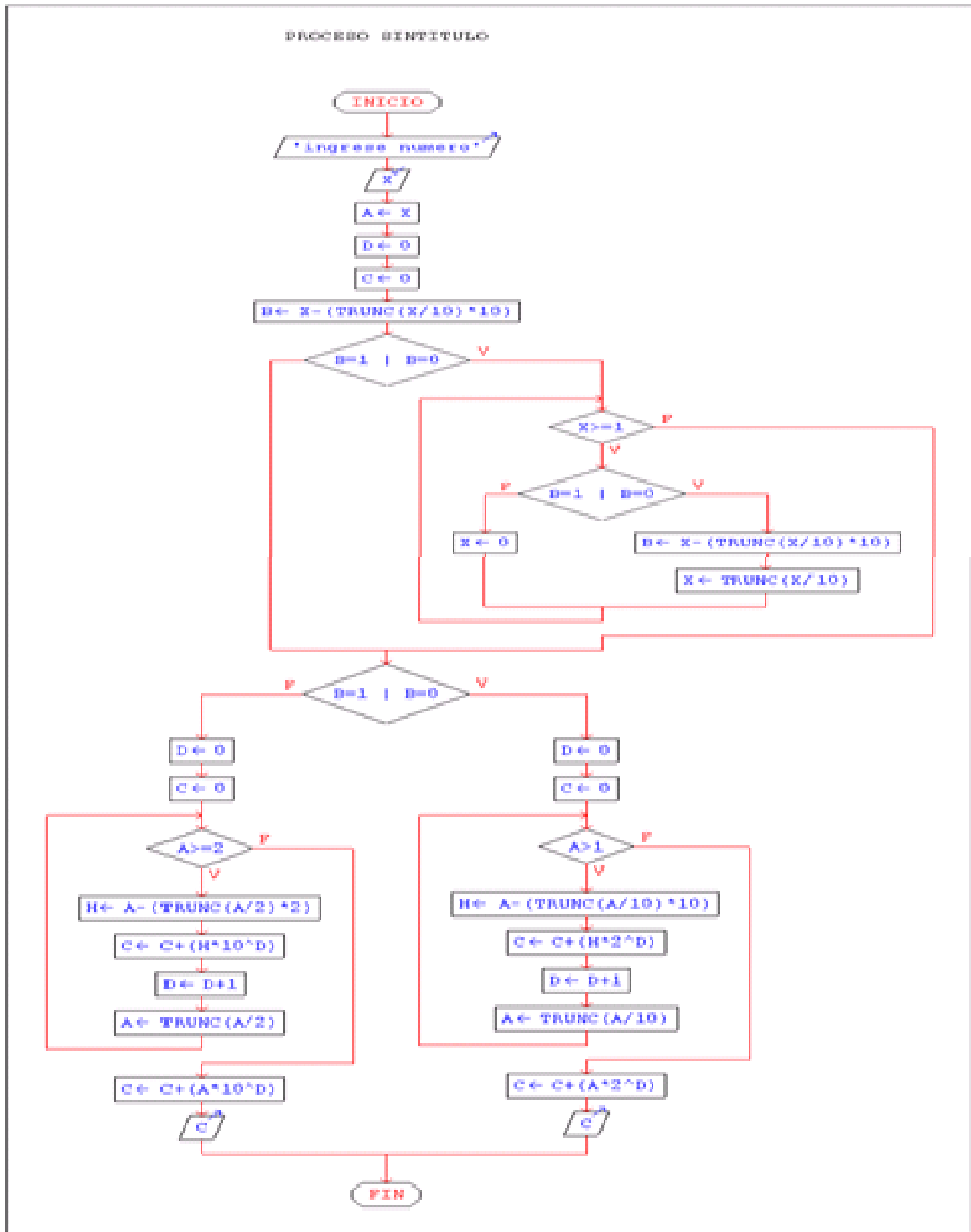
DIAGRAMAS DE FLUJO

Proceso SinTitulo

```
escribir "ingrese numero"
leer x;
a<-x
d<-0
c<-0
b<-x-(trunc(x/10)*10)
si b=1 | b=0 entonces
  mientras x>=1 hacer
    si b=1 | b=0 entonces
      b<-x-(trunc(x/10)*10)
      x<-trunc(x/10)
    sino
      x<-0
  finmientras
finmientras
finmientras
si b=1 | b=0 entonces
  d<-0
  c<-0
  mientras a>1 hacer
    h<-a-(trunc(a/10)*10)
    c<-c+(h*2^d)
    d<-d+1
    a<-trunc(a/10)
  finmientras
  c<-c+(a*2^d)
  escribir c;
sino
  d<-0
  c<-0
  mientras a>=2 hacer
    h<-a-(trunc(a/2)*2)
    c<-c+(h*10^d)
    d<-d+1
    a<-trunc(a/2)
  finmientras
  c<-c+(a*10^d)  escribir c;
finmientras
FinProceso
```

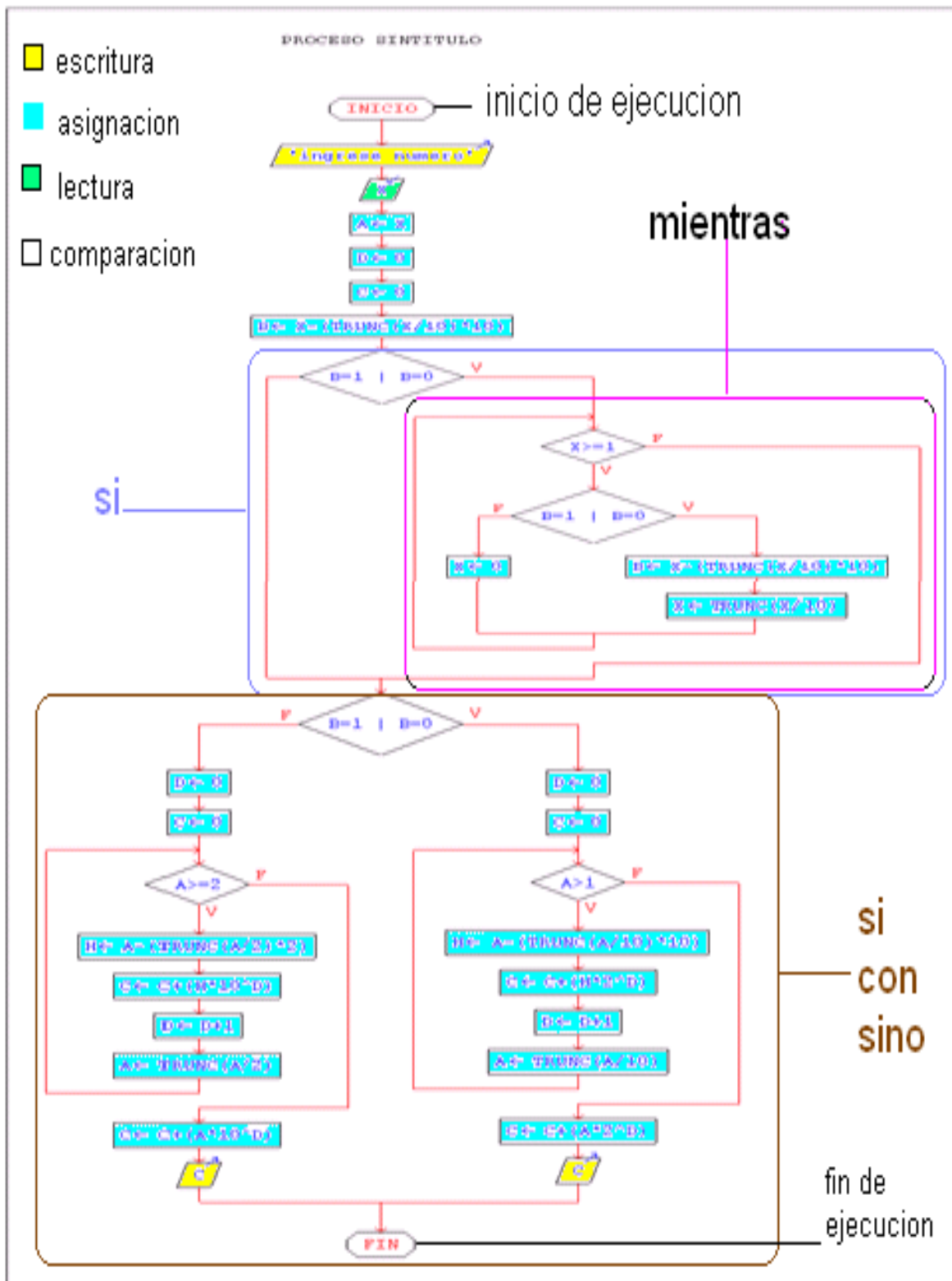
DIAGRAMAS DE FLUJO

Diagrama de flujo del algoritmo



DIAGRAMAS DE FLUJO

Explicación del diagrama de flujo de dicho algoritmo



8. REFERENCIAS

1. <http://www.elticus.com/?contenido=19>
2. www.mis-algoritmos.com/diagramas-flujo.html
3. <http://pseint.sourceforge.net/>