UT2 ALGORITMOS

MODULO: PROGRAMACIÓN

Alberto Ruiz

Contenido

1 Algoritmos. 3
2 Diagramas de flujo. 6
3 Ejercicios de diagramas de flujo. 16

UT2: Algoritmos

2.1. Algoritmos.
2.2. Diagramas de flujo.
2.3. Ejercicios de diagramas de flujo.

1 Algoritmos.

Un algoritmo de programación en informática está formado por **una serie de instrucciones que realizan una serie de procesos con el objetivo de resolver un problema**. A través de algoritmos, los programadores pueden dar respuesta a cualquier problema antes de modificarlo en el lenguaje elegido.

Un algoritmo tiene tres partes definidas.

1. **Input**. La entrada o input es donde se aportan todos los datos necesarios para realizar los procesos de resolución del problema.
2. **Procesamiento**. Son todas las acciones que se deben realizar para resolver el problema utilizando los inputs obtenidos.
3. **Output**. El output o salida muestra los resultados de resolución del problema obtenidos.

Los algoritmos se caracterizan por ser secuenciales, precisos, ordenados, concretos y finitos. Si un mismo algoritmo obtiene siempre los mismos valores de entrada, el output que muestra también debe ser siempre el mismo.

Para resolver un problema pueden existir diferentes algoritmos. En la programación informática siempre se utilizará el algoritmo que resuelva un problema de forma más sencilla y con menos pasos.

Algoritmos de programación, ejemplos

Si se quiere realizar un programa informático para **calcular el área de una circunferencia**, el algoritmo sería el siguiente:

- Input, que en este caso es el radio de la circunferencia.
- Proceso a realizar, que sería la fórmula matemática para el cálculo del área de una circunferencia ($A = \pi \times r^2$).
- Output, que en este caso es la solución obtenida al multiplicar π por el radio al cuadrado que se proporciona en el input.

Con este algoritmo se resuelve el problema y el programador puede codificar todo el proceso en el lenguaje de programación que utilice.

Para qué sirven estos algoritmos

Los algoritmos y los lenguajes de programación están directamente relacionados, pues los primeros **son herramientas fundamentales para poder codificar de forma más sencilla la resolución de problemas**.

Cuando se crea un programa informático se persigue ir resolviendo una serie de problemas. Para lograrlo se utiliza un algoritmo con el procedimiento óptimo para alcanzar la mejor solución de la forma más rápida y sencilla, haciendo que la codificación del programa sea más fácil y precisa.

Un algoritmo se puede ver como un **borrador donde se incluyen una serie de instrucciones para resolver un problema, y que sirve de guía al programador** para codificar un programa.

Tipos de algoritmos de programación

Los tipos de algoritmos en programación se pueden clasificar en cuatro diferentes:

Algoritmos computacionales.

En este tipo de algoritmos la resolución depende de un cálculo matemático por lo que puede ser resuelto por una computadora o calculadora.

Algoritmos no computacionales.

Son aquellos algoritmos que no pueden ser resueltos por una computadora y necesitan de la intervención humana para ello.

Algoritmos cualitativos.

En este tipo de algoritmo no se realizan cálculos numéricos para su resolución, sino secuencias lógicas o formales.

Algoritmos cuantitativos.

Este tipo de algoritmo depende de un cálculo numérico para poder mostrar el output.

Cómo se crean los algoritmos de programación

Para poder crear un algoritmo de programación que resuelva el programa y facilite el proceso de codificación, hay que seguir una serie de pasos:

Analizar el problema

Lo primero que debe realizarse es un estudio y análisis del problema que se quiere resolver, antes de hacer nada más. Se debe definir cuál es el objetivo que se persigue para poder diseñar el algoritmo de forma eficiente.

Definir las entradas necesarias

Conociendo bien cuál es el problema a solventar, se deben definir cuáles son los requisitos o entradas necesarias para poder resolverlo. Puede que solo sea una, o puede que sean varios los requisitos que deben proporcionarse en el input.

Implementar las instrucciones necesarias

Con los valores obtenidos en el input se deben establecer las instrucciones necesarias para dar resolución al problema. Cumpliendo las características de los algoritmos será más sencillo llevar estas instrucciones luego a un lenguaje de programación.

Mostrar los resultados

En esta parte se muestran los resultados obtenidos en todo el proceso y que son el objetivo final de la creación del algoritmo.

Verificar las características del algoritmo

Se debe verificar el funcionamiento del algoritmo y si cumple sus principales características (es finito, concreto, secuencial, etc.).

Codificar el algoritmo

Finalmente, tras comprobar que el algoritmo cumple con todos sus requisitos y que resuelve el problema, se puede pasar todo el proceso al lenguaje de programación seleccionado.

Los **algoritmos de programación** son herramientas fundamentales para poder construir programas más eficientes invirtiendo menos tiempo y esfuerzo.

Crear un algoritmo para solucionar un problema siempre debe ser el paso previo de todo programador antes de lanzarse a codificar en un lenguaje de programación.

2 Diagramas de flujo.

¿Qué es un Diagrama de Flujo?

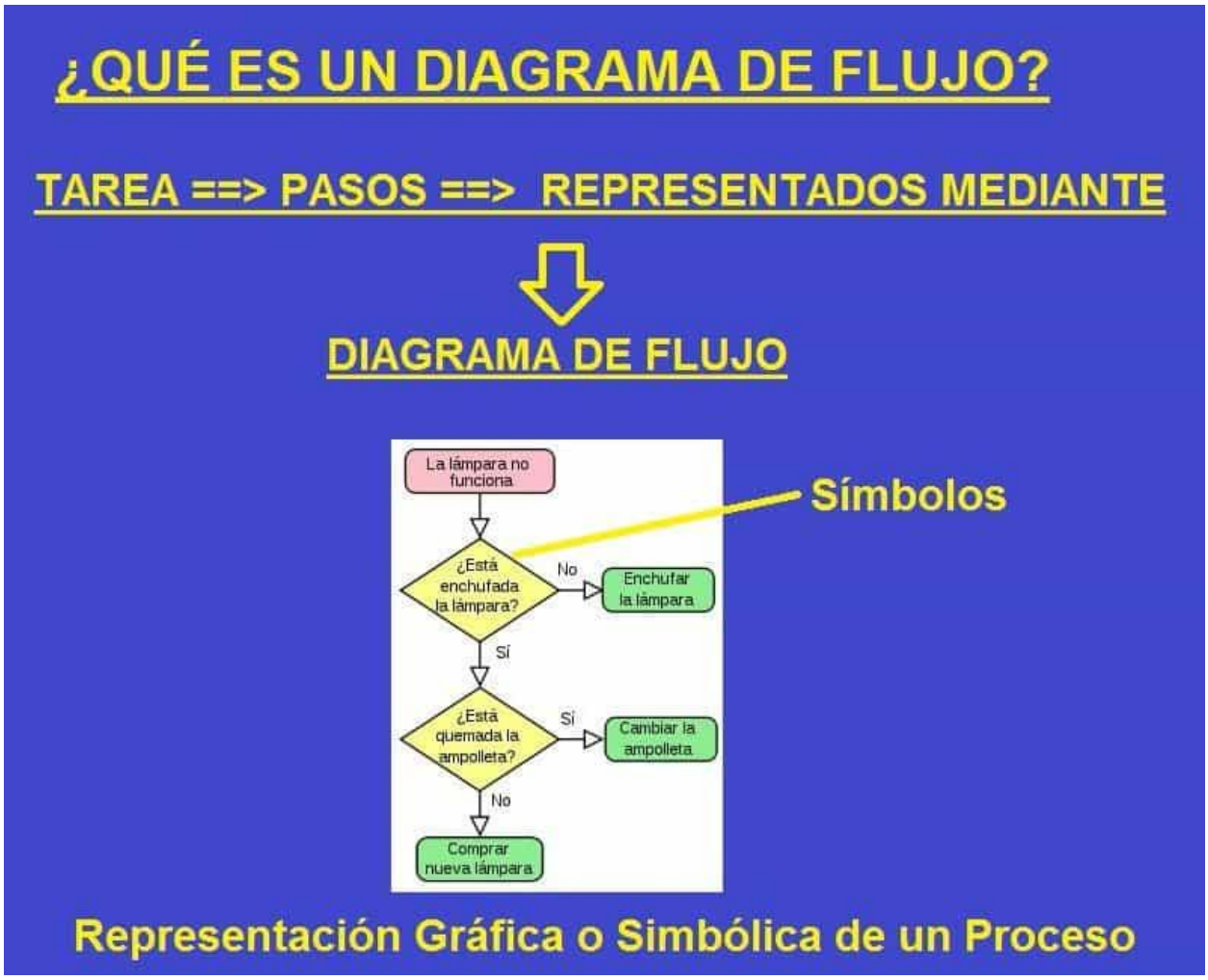
Un **diagrama de flujo**, también llamado **Flujograma de Procesos o Diagrama de Procesos**, representa la **secuencia** o los pasos lógicos (ordenados) **para realizar una tarea mediante unos símbolos**.

Dentro de los símbolos se escriben los pasos a seguir.

Los diagramas de flujo representan la secuencia lógica o los pasos que tenemos que dar para realizar una tarea mediante unos símbolos y dentro de ellos se describen los pasos ha realizar.

Un diagrama de flujo debe proporcionar una información clara, ordenada y concisa de todos los pasos a seguir.

Por lo dicho anteriormente, podríamos decir que: "**Un diagrama de flujo es una representación gráfica o simbólica de un proceso**".



El proceso o pasos que representa el diagrama de flujo puede ser de cualquier tipo, desde los pasos para freír un huevo, como luego veremos, hasta los pasos para realizar un enorme programa informático.

Muchas veces antes de realizar un diagrama de flujo se realiza un "Algoritmo" del problema o proceso a desarrollar.

Un **algoritmo** describe una secuencia de **pasos escritos** para realizar un tarea.

El **Diagrama de Flujo** es su representación esquemática.

Algoritmo: Escribir los pasos ordenados a realizar para solucionar el problema.

Diagrama de Flujo: Representación mediante un esquema con símbolos del algoritmo.

Los diagramas de flujo son una excelente herramienta para **resolver**

problemas, **comprender el proceso** a seguir así como para **identificar posibles errores** antes del desarrollo final de la tarea.

Cómo Hacer un Diagrama de Flujo

Normalmente para realizar un diagrama de flujo primero se hace el algoritmo.

Un ejemplo para cocinar un huevo para otra persona sería:

- Pregunto si quiere el huevo frito.
- Si me dice que si, lo frio, si me dice que no, lo hago hervido.
- Una vez cocinado le pregunto si quiere sal en el huevo.
- Si me dice que no, lo sirvo en el Plato, si me dice que si, le hecho sal y después lo sirvo en el plato.

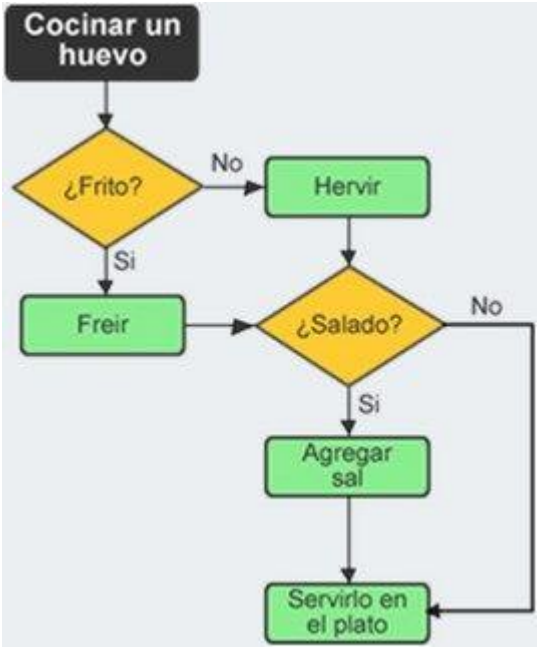
Si te fijas los pasos no pueden cambiar su posición.

Sería imposible preguntarle si lo quiere frito después de haberlo hervido, por ejemplo.

Es muy importante que los pasos sean una secuencia lógica y ordenada.

Ahora que ya sabemos todos los pasos, mediante el algoritmo, podemos hacer un esquema con estos pasos a seguir.

Este esquema será el **Diagrama de Flujo**.



Si uno tiene experiencia puede prescindir del algoritmo escrito pero siempre tendremos que tenerlo en mente para hacer el diagrama de flujo sin

equivocarnos.

Más abajo te dejamos varios ejemplos de diagramas de flujo.

¿Para Qué se Usan los Diagramas de Flujo?

Se usan para hacer un programa informático, para analizar lo que tiene que hacer un robot, en los procesos industriales, etc.

Un diagrama de flujo es útil en todo aquello que se necesite una previa organización antes de su desarrollo.

En la realización de un programa informático es imprescindible primero realizar el diagrama de flujo, independientemente del lenguaje de programación que usemos después.

Una vez que tenemos nuestro diagrama de flujo solo tendremos que conocer las órdenes del lenguaje que realizan esas tareas que se especifican en el diagrama

Reglas y Símbolos Para la Construcción de un Diagrama de Flujo

1. Todos los símbolos han de estar conectados

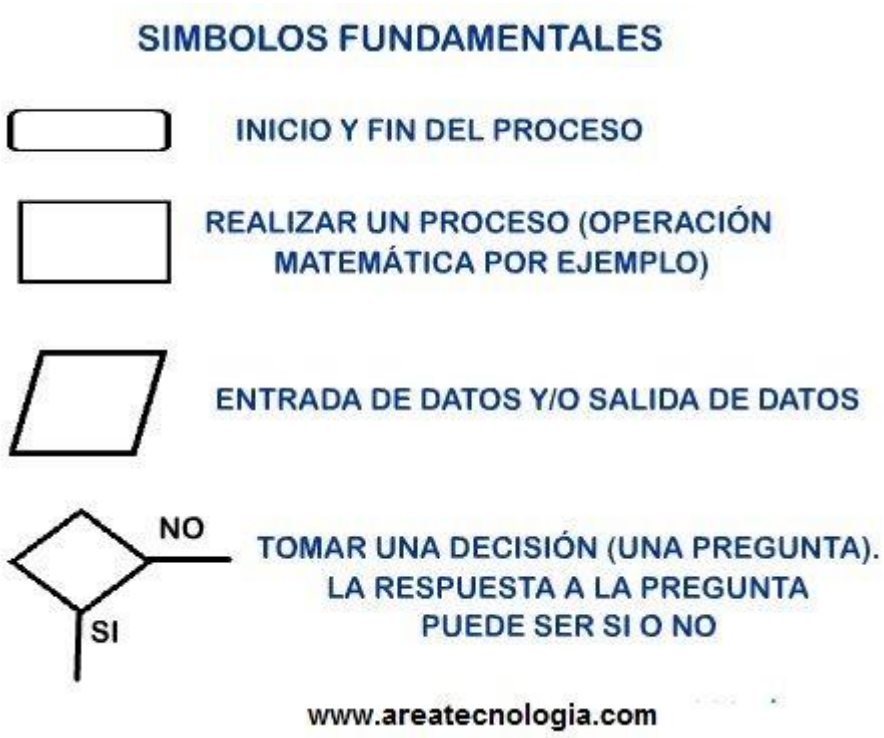
2. A un símbolo de proceso pueden llegarle varias líneas

3. A un símbolo de decisión pueden llegarle varias líneas, pero sólo saldrán dos (Si o No, Verdadero o Falso).

4. A un símbolo de inicio nunca le llegan líneas.

5. De un símbolo de fin no parte ninguna línea.

Los **símbolos que se usan** para realizar los diagramas de flujo son lo siguientes:



- En el Símbolo de decisión puede tomar los valores de salida SI o NO o también VERDADERO o FALSO.

- El símbolo de Inicio o Final del Diagrama puedes ser un cuadrado con los bordes redondeados o una elipse.

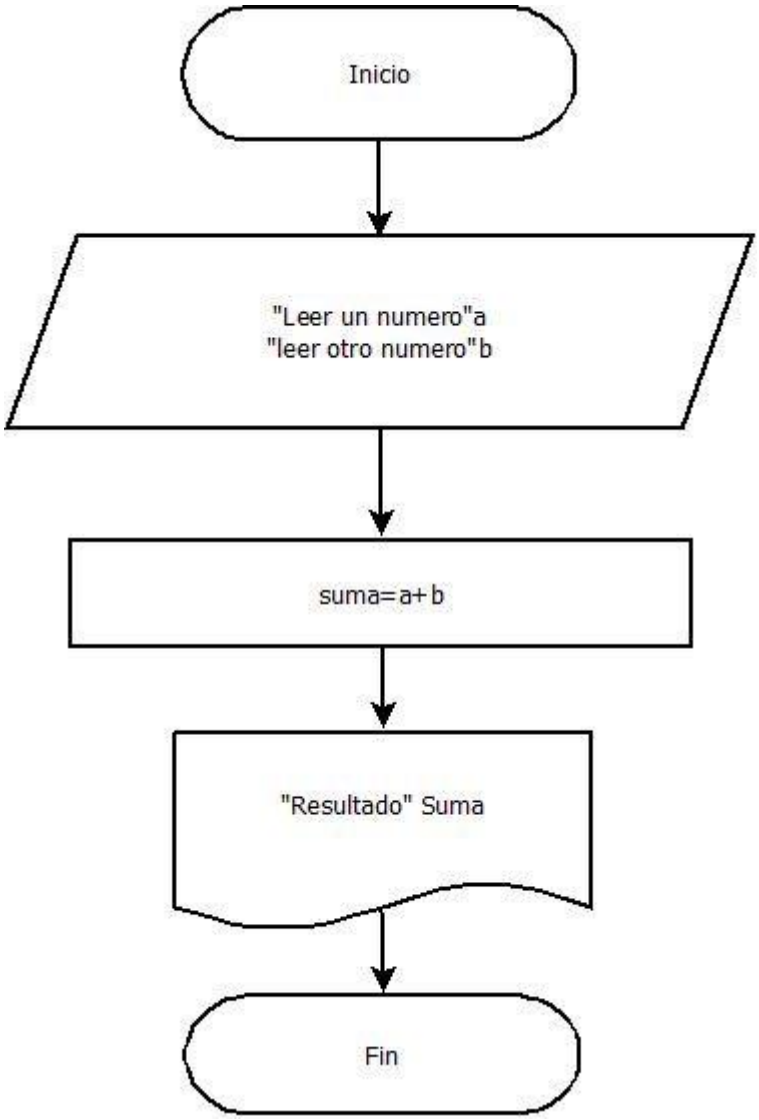
- Se pueden utilizar colores para lo símbolos.

Ejemplos de Diagramas de Flujo

Veamos **un primer ejemplo** muy sencillo.

Queremos hacer un programa informático que nos sume dos número y nos de el resultado en pantalla.

Solución del ejemplo:



El símbolo de resultado es un símbolo usado en los diagramas para soluciones con el ordenador.

Es el símbolo de salida del resultado por la pantalla del ordenador.

Ves que es muy sencillo, hay que ir poniendo los pasos lógicos que se deben seguir para realizar la tarea o el programa.

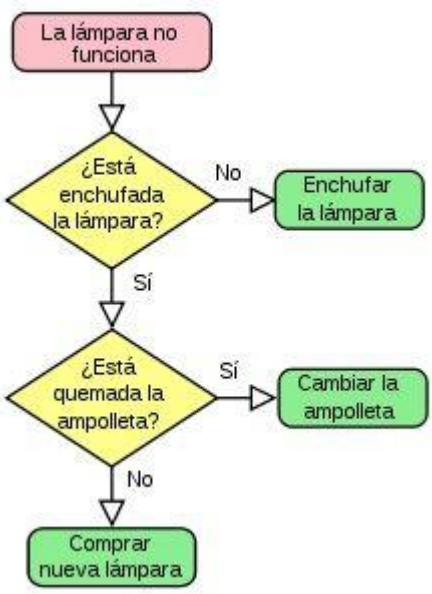
En el ejercicio tenemos el inicio y el fin, una entrada de datos, para meter los 2 números, una operación a realizar, la suma, y un resultado a mostrar.

Cada uno de esos pasos con su símbolo correspondiente en el diagrama.

Otro ejemplo de un diagrama de flujo para una operación sencilla.

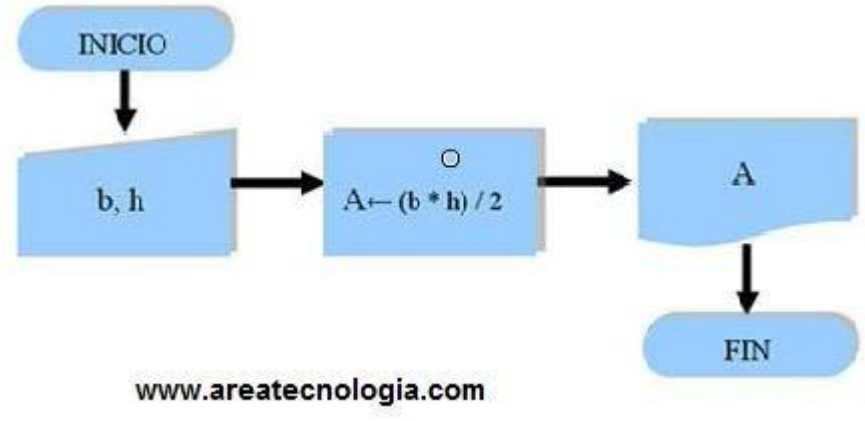
Imaginemos que tenemos una lámpara o bombilla y queremos hacer el

diagrama de flujo para saber que hacer cuando la lámpara no funciona.



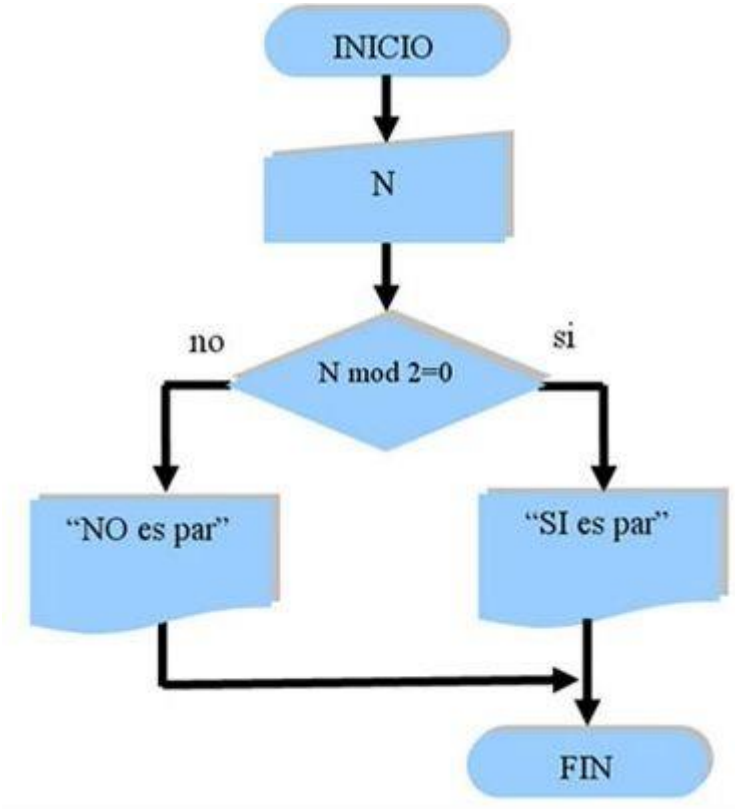
¿Hacemos otro?

Bueno vamos hacer uno que nos muestre el resultado del área de un triángulo en pantalla.



Como ves, en este ni siquiera hemos puesto las operaciones dentro de los símbolos, ya que con la forma del símbolo ya se entiende.

No hemos usado mucho el símbolo de tomar un decisión, por eso vamos hacer **uno en el que nos diga si el número es par o impar**:



La palabra "mod" significa dividir, por lo tanto "mod 2" es dividir entre 2.

Como ya debes saber si divido un número entre 2 y el resto es 0 el número es par, en caso contrario sería impar.

Hay tenemos la decisión.

¿Al dividirlo entre 2 el resto es 0? Hay 2 posibilidades.

Si lo es, se ve en pantalla "Si es par", si no lo es, se ve en pantalla "No es par".

Eso es la toma de decisiones.

Toma una salida en función del resultado de la entrada.

Además los diagramas de flujo no solo valen para informática, incluso podemos hacer uno para cocinar un huevo, como vimos al principio.

Bueno ahora hagamos uno un poco más complicado.

Tenemos que hacer **un diagrama de flujo para mostrar la suma de los 50 primeros números**.

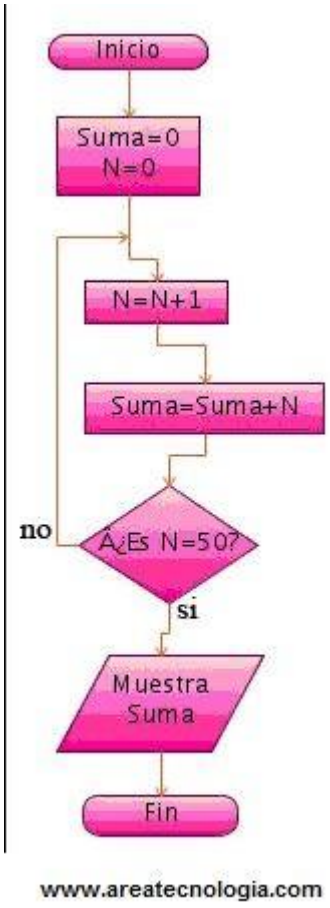
Lo primero es poner a cero la suma y dar el primer número a sumar que será el 0.

Fíjate que el diagrama acaba cuando N, que es el número en cada momento, es 50.

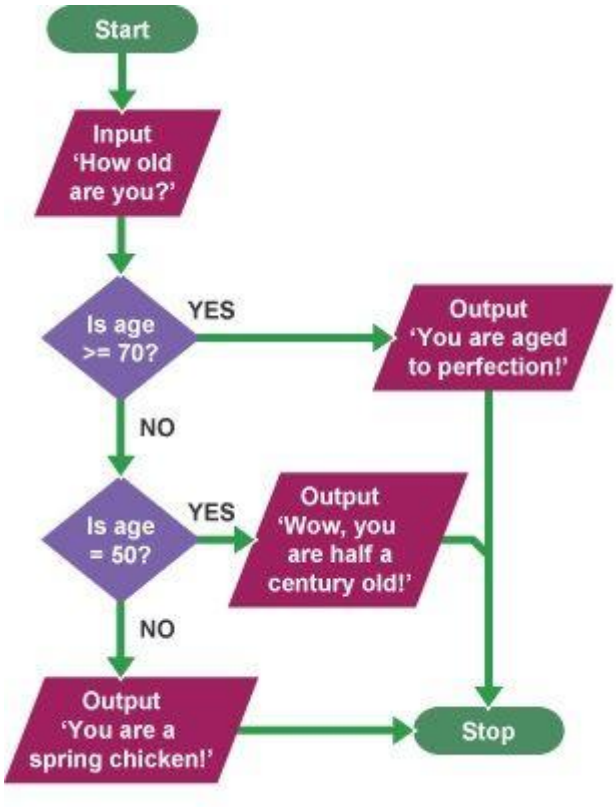
Mientras no sea 50 el programa vuelve a la tercera secuencia que será sumarle un número al anterior N = N + 1.

Intenta comprenderlo y ver lo que hace.

Puedes realizar mentalmente el diagrama para el número 0 y verás como lo acabas entendiendo.



¿Ponemos un diagrama de flujo en ingles?



En el siguiente enlace Ejemplos de Diagramas de Flujo te presentamos 15 diagramas de flujo resueltos.

Programa Para Crear Diagramas de Flujo

Ya existen programas que automatizan y simplifican la creación de los diagramas de flujo.

Si quieres descargarte gratis un programa para crear diagramas de flujo de forma sencilla aquí tienes este enlace: Programa Crear Diagramas de Flujo.

Recurso

draw.io

Wireflow

Lucidchart

Creately

Google Drawings

Diagrama de flujo de canva:

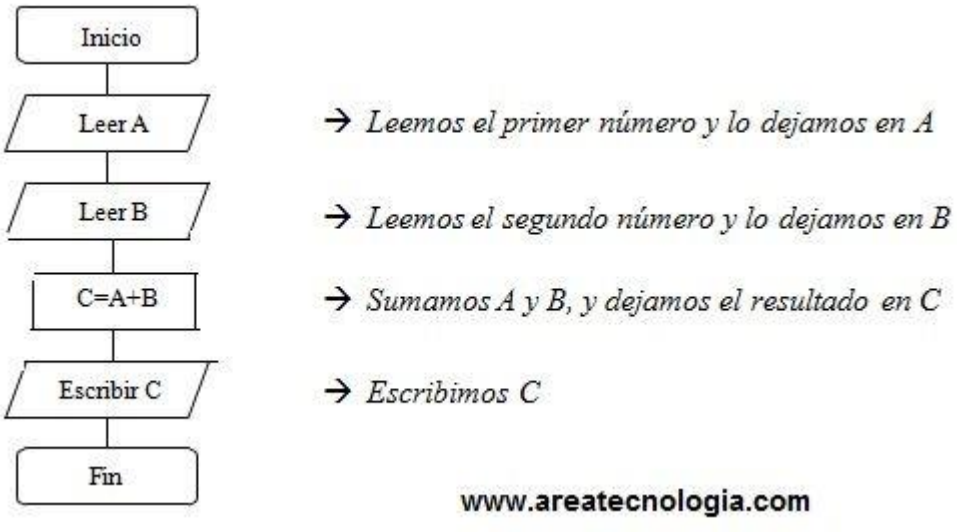
https://www.canva.com/es_es/graficos/diagramas-flujo/

3 Ejercicios de diagramas de flujo.

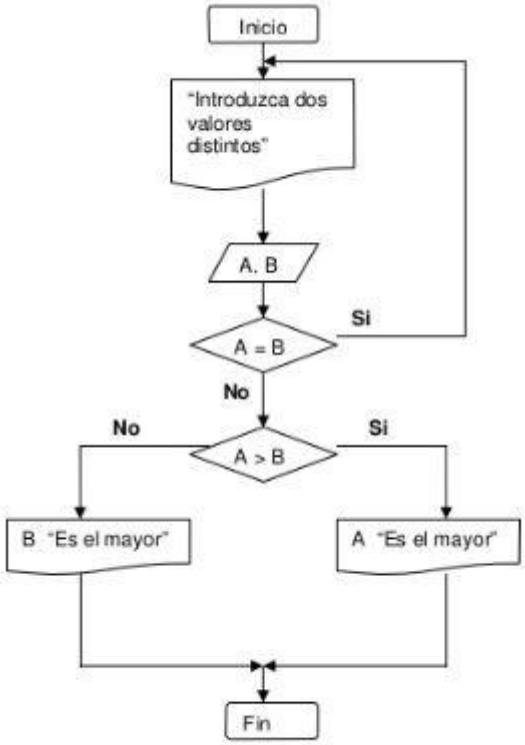
Aquí os dejamos una serie de **ejercicios resueltos** (15 en total) y **con su explicación** a modo de **ejemplos de construcción de diagramas de flujo** de procesos o **flujograma de procesos**.

Si no tienes claro la teoría te recomendamos primero que veas este enlace: Diagramas de Flujo.

1. Hacer el Diagrama de Flujo para sumar dos números leídos por teclado y escribir el resultado.



2. Hacer un diagrama de flujo que permita leer 2 números diferentes y nos diga cual es el mayor de los 2 números.



El pseudocódigo para este diagrama sería:

1.Inicio

2.Inicializar variables: A = 0, B = 0

3.Solicitar la introducción de dos valores distintos

4.**Leer** los dos valores

5.Asignarlos a las variables A y B

6.**Si** A = B **Entonces** vuelve a 3 porque los valores deben ser distintos

7.Si A>B **Entonces**

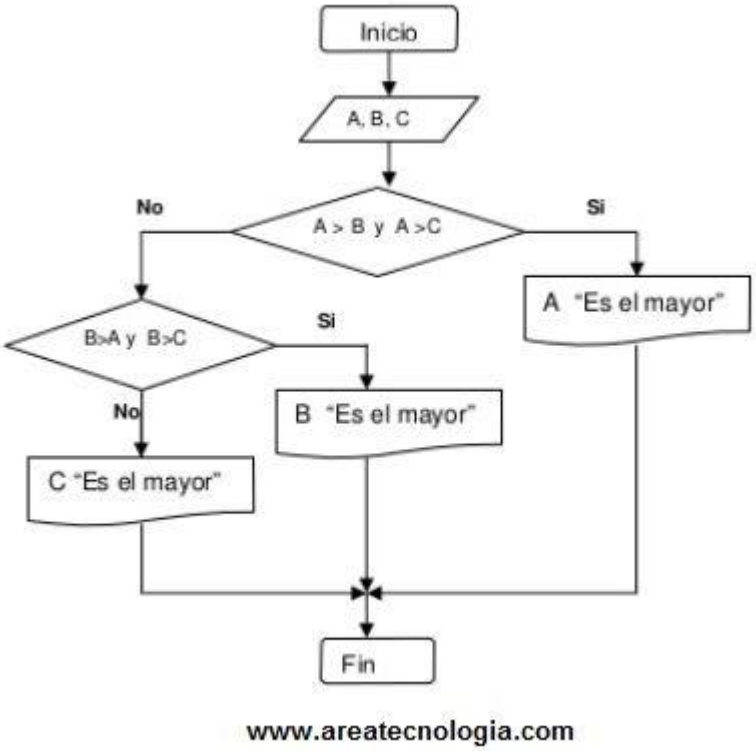
Escribir A, "Es el mayor"

8.**De lo contrario**: **Escribir** B, "Es el mayor"

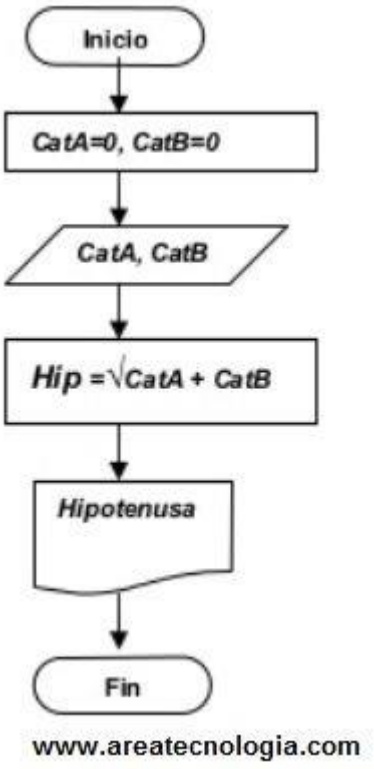
9.Fin_Si

10. Fin

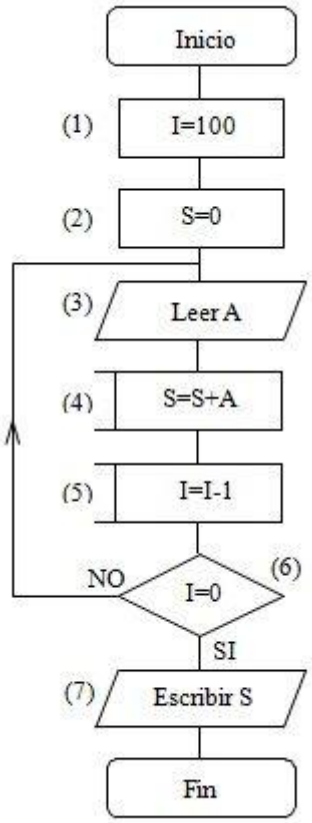
3. Crear un diagrama de flujo de procesos en el que se almacenen 3 números en 3 variables A, B y C. El diagrama debe decidir cual es el mayor y cual es el menor



4. Realizar el diagrama de flujo para que nos calcule la hipotenusa de un triángulo rectángulo, conocidos su dos catetos.



5. Diagrama de Flujo para sumar 100 números leídos por teclado.



Explicación:

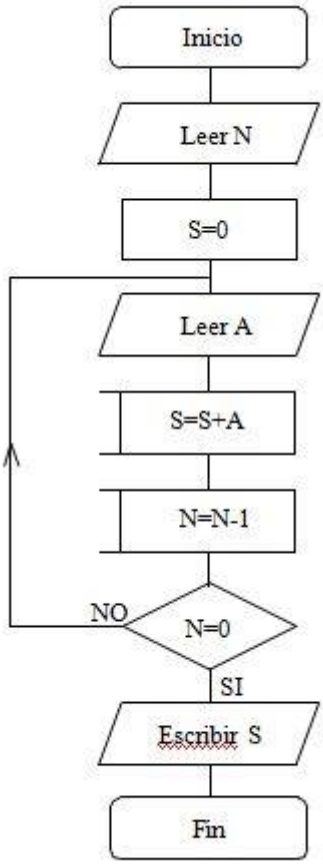
- En I contamos los números que quedan por sumar.
- En S calculamos la suma.
- A se emplea para leer temporalmente cada número.

Vamos a ver paso a paso como funciona. Supongamos que los datos son: 7, -1, 8, 5, ...

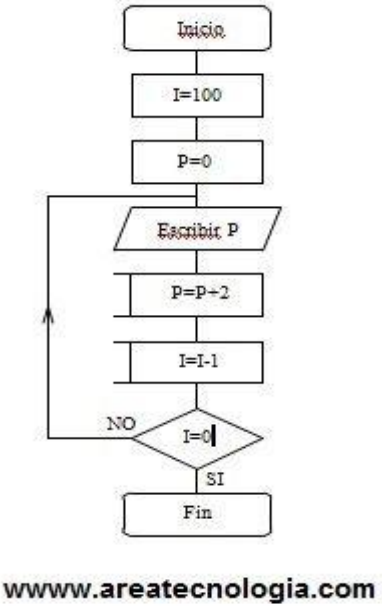
(1) I=100 (números a sumar)
(2) S=0 (suma, inicialmente 0) (3) Leer A. El primero es 7, luego A=7 (4) S=S+A=0+7=7
(5) I=I-1=100-1=99 (6) ¿I=0? ® NO
(3) Leer A, ahora A=-1
(4) S=S+A=7-1=6
(5) I=I-1=99-1=98
(6) ¿I=0? ® NO

Cuando I=0 habremos sumado los 100 números y pasaremos a: (7) Escribir S que será la suma.

6. Modificar el anterior para que permita sumar N números. El valor de N se debe leer previamente por teclado.



7. Hacer un diagrama de flujo que permita escribir los 100 primeros pares.



wwww.areatecnologia.com

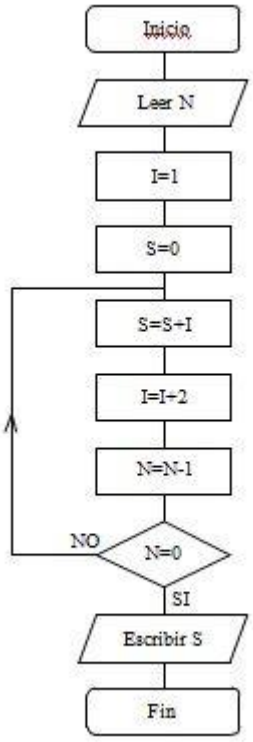
Explicación de la solución:

P: Variable para contener el siguiente par que se debe escribir.

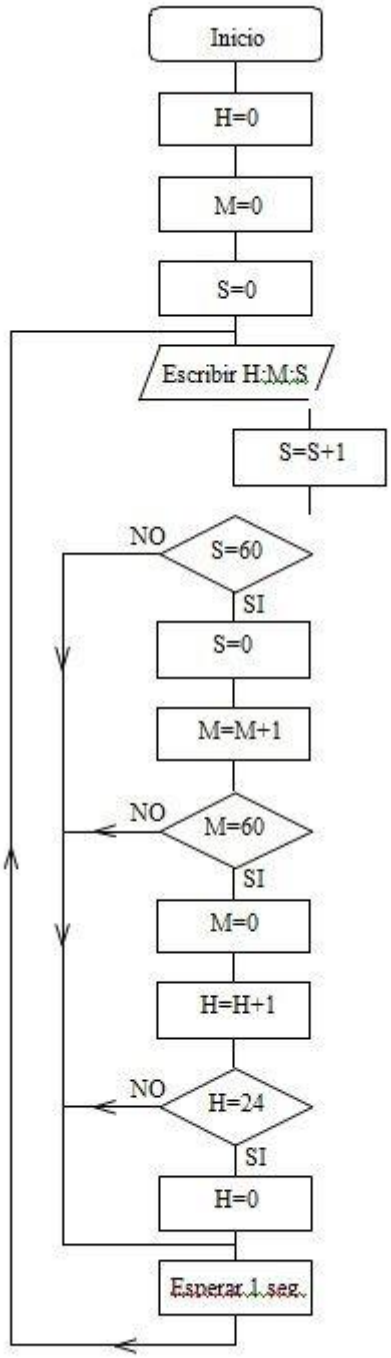
I: Contador de pares que quedan por escribir.

El proceso es similar al anterior. Necesitamos un bucle para contar 100 veces y dentro de él escribimos el par e incrementamos para obtener el siguiente.

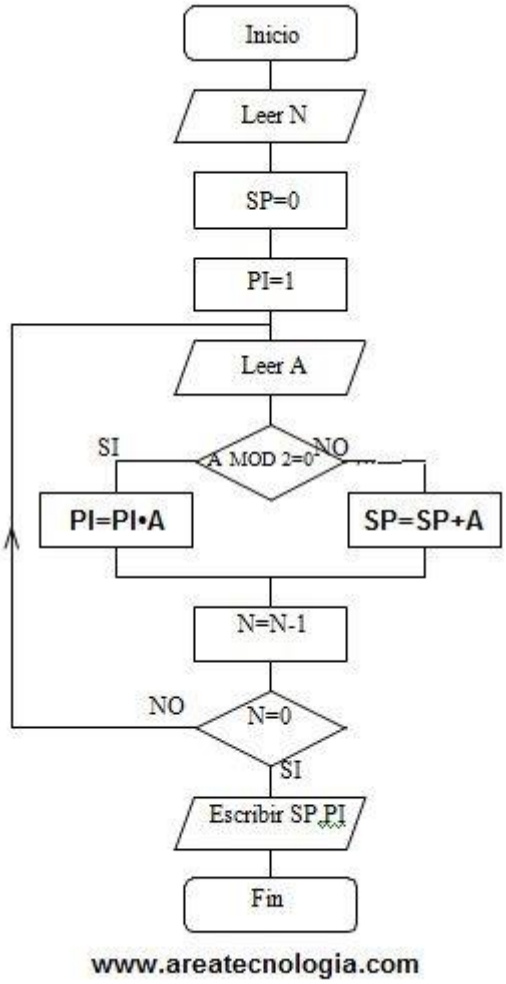
8. Hacer el diagrama de flujo para sumar los N primeros impares. Realizar después uno que haga lo mismo con los pares y otro con los múltiplos de 3.



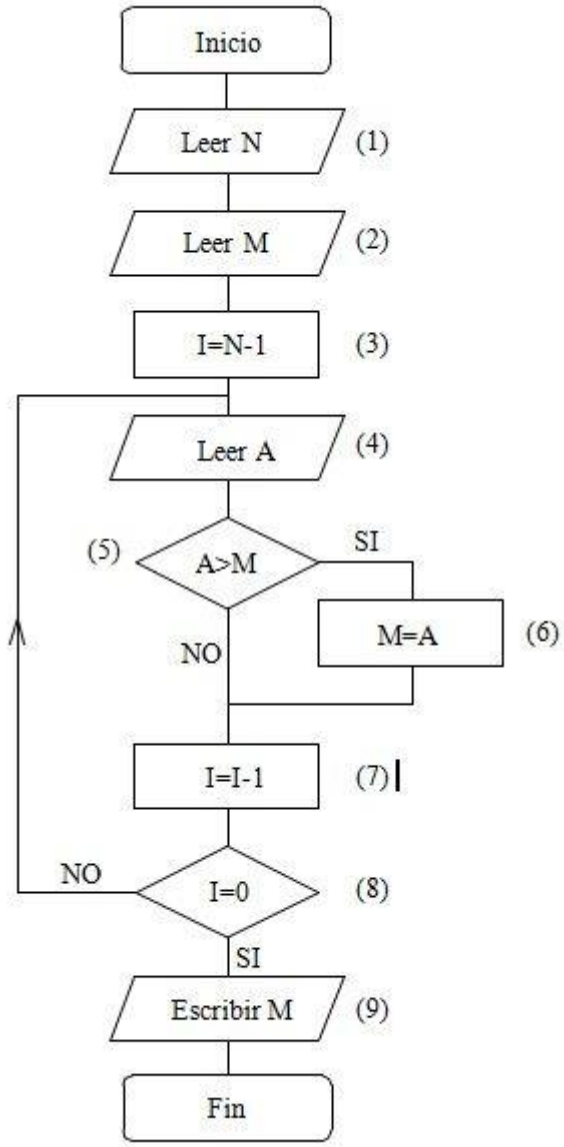
9. Hacer un diagrama de flujo que simule un reloj.



10. Hacer un organigrama que lea N números, calcule y escriba la suma de los pares y el producto de los impares.



11. Calcular el máximo de N números leídos desde teclado.



Explicación del Ejemplo de Diagrama:

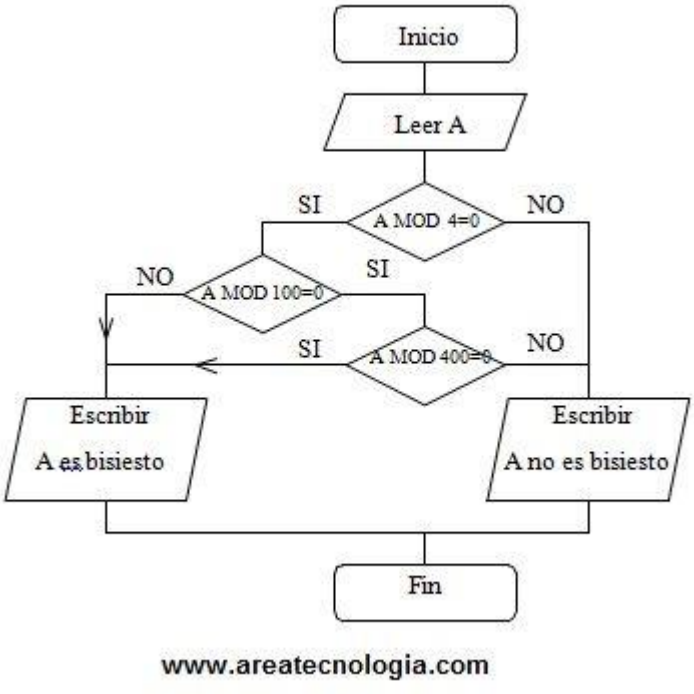
Vamos a almacenar en M el máximo de los números que se hayan leído, el primero va directamente a M y los N-1 restantes los leemos en A, comparamos con M y si son mayores cambiamos el máximo temporal.

Al final se escribe el resultado.

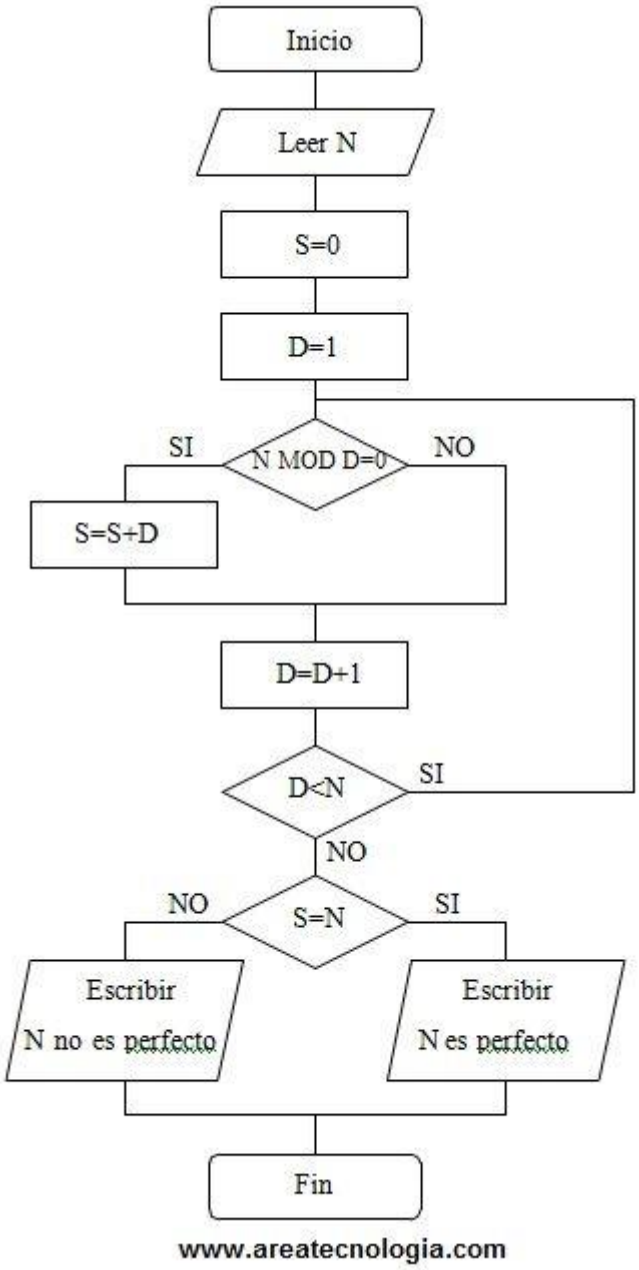
Vamos a ejecutarlo paso a paso para N=4, empleando como datos: 2, 3, -1, 7.

(1) Leer N ® N=4
(2) Leer M ® M=2
(3) I=N-1=3
(4) Leer A ® A=3 (5) ¿A>M? ® SI (6) M=A=3
(7) I=I-1=3-I=2
(8) ¿I=0? ® NO (4) Leer A ® A=-1

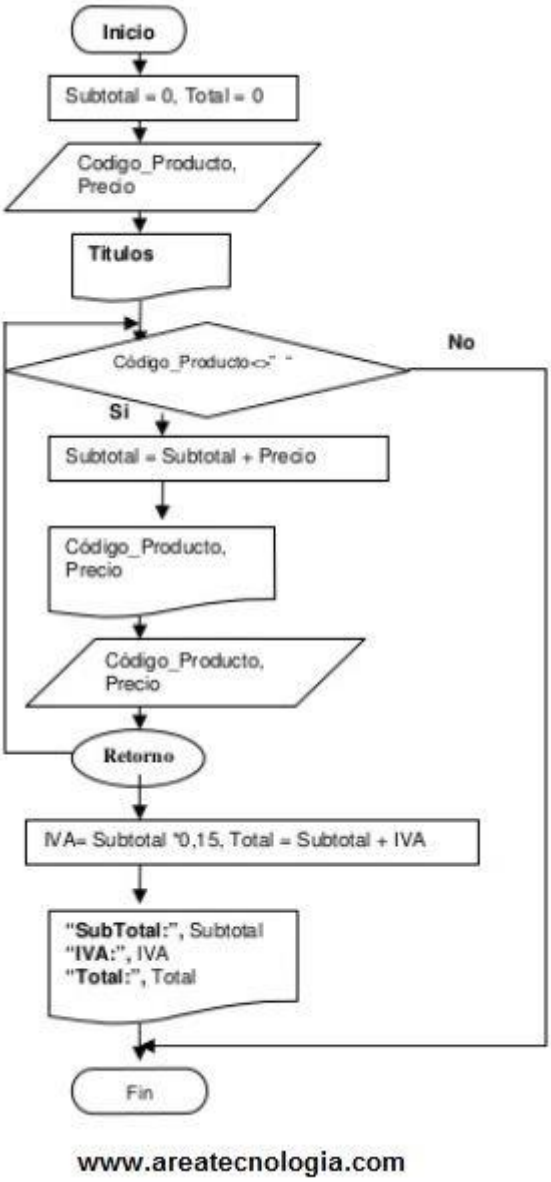
12. Un año es bisiesto si es múltiplo de 4, exceptuando los múltiplos de 100, que sólo son bisiestos cuando son múltiplos además de 400, por ejemplo el año 1900 no fue bisiesto, pero el año 2000 si lo será. Hacer un organigrama que dado un año A nos diga si es o no bisiesto.



13. Dados dos números enteros positivos N y D, se dice que D es un divisor de N si el resto de dividir N entre D es 0. Se dice que un número N es perfecto si la suma de sus divisores (excluido el propio N) es N. Por ejemplo 28 es perfecto, pues sus divisores (excluido elv28) son: 1, 2, 4, 7 y 14 y su suma es 1+2+4+7+14=28. Hacer un organigrama que dado un número N nos diga si es o no perfecto.



14. Realiza el diagrama de flujo que simule una caja registradora.



El pseudocódigo para esta caja registradora es:

1. **Inicio**
2. Declaración de Variables:
   Sub_total=0,Total = 0
3. **Ingrese** "Código de Producto y Precio:"
4. **Almacenar** Codigo_Producto, Precio
5. **Imprimir** líneas de títulos del recibo de pago
6. **Mientras** Código_Producto <> " "
7. Subtotal = Subtotal + Precio
8. **Imprimir** Codigo_Producto, Precio
9. **Ingrese** "Código de Producto y Precio:"
10. **Fin_Mientras**
11. IVA = Subtotal * 0,15
12. Total = Subtotal + IVA
13. **Imprimir** "**Sub Total :** ", Subtotal
14. **Imprimir** " **IVA :** ", IVA
15. **Imprimir** "**Total:** ", Total
16. **Fin**