

DIAGRAMACION Y PROGRAMACION

Índice de Contenidos:

- Pasos para la solución de problemas con programas de computador
- Símbolos y convenciones
- ¿Qué es un Diagrama de Flujo?
- Cómo Hacer un Diagrama de Flujo
- Para qué se usan los Diagramas de Flujo
- Reglas y Símbolos en los Diagramas de Flujo.
- Ejemplos de Diagramas de Flujo
- Programa Para Crear Diagramas de Flujo

Pasos para la solución de un problema por computador

1. UNIVERSIDAD TÉCNICA DE AMBATO FACULTAD DE INGENIERÍA CIVIL Y MECÁNICA CARRERA DE INGENIERÍA CIVIL SEGUNDO "D" Empleo de Nuevas Tecnologías de Información y Comunicación PASOS PARA LA SOLUCIÓN DE UN PROBLEMA POR COMPUTADOR● Eduardo Zavala● Talía Cevallos

2. Índice:● Caratula.● Índice.● Objetivos.● Resumen.● Desarrollo.1. Definición y delimitación del Problema.2.Análisis del Problema3.Diseño del Algoritmo4.Codificación5.Compilación6.Depuración7.La documentación8.Ejecución9.MantenimientoConclusiones.● Bibliografía.

3. Resumen: El siguiente trabajo contiene los pasos que se deben seguir para la resolución de un problema por computador.

4. Objetivos:● Determinar la manera de resolver pasos para solucionar problemas por el computador.● Encontrar las diferentes maneras de solucionar los problemas en el computador.● Explicar cada uno de los elementos dados por el documento para la solución de problemas en el computador.

5. Desarrollo: Definición y Delimitación del Problema Es fundamental conocer el problema por completo. Si no se tiene No hay que Con el enunciado claro lo que se confundir el podremos enunciado observar si el pretende porque se problema tiene o solucionar no puede terminar no solución y/o si tiene caso dando una se le podría dar continuar con solución a un una alternativa de la siguiente problema solución. Fase. Inexistente.

6. Análisis del Problema Es necesario definir los datos de entrada, los datos de salida, cálculos y fórmulas para la resolución del problema. Es conveniente ponerse en el lugar del computador para plantear las condiciones que satisfagan las condiciones dadas para la resolución del problema.

7. Diseño del Algoritmo Se determinan los pasos o instrucciones, en orden lógico de ejecución. De esta fase depende el éxito o el fracaso de la solución de un problema. ● Las instrucciones están determinadas por el computador.● Puede utilizar técnicas de representación de un logaritmo.

8. Codificación Es escribir la solución del problema en un lenguaje comprendido por la computadora. Basado en el diseño del algoritmo, conocido como código fuente. Se escribe en un lenguaje de programación o lenguaje de alto nivel.

9. Compilación El computador hace una revisión de las instrucciones colocadas por el programador, teniendo en cuenta que las instrucciones están escritas correctamente en sintaxis y la gramática según el lenguaje de programación. Por último se transcribe el programa al interior de la memoria.

10. Depuración Los errores dentro de la AI identificar errores en el programación son muchos programa o código fuente, y aumentan con la se debe hacer una complejidad y dificultad del depuración y corrección de problema. Errores, Es de mucha ayuda documentar nuestros algoritmos y los programas que desarrollemos, ya que son una guía o una comunicación escrita entre el código y cualquier usuario.

11. La documentación INTERNA EXTERNA Comentarios o Normalmente se le mensaje que conoce como Manuales añaden al código del usuario, es un fuente para hacer documento escrito más claro el con la descripción entendimiento de un del Problema, proceso. Nombre del Autor, algoritmo, diccionario de datos y código fuente.

12. Ejecución Se utilizan los dispositivos de entrada y de salida. La forma de estar seguro de la efectividad de un algoritmo es comparar los Requerimientos con las Soluciones

13. Mantenimiento Se realiza después de terminado el programa, cuando se detecta que es necesario hacer algún cambio, ajuste o complementación al programa para que siga trabajando de manera correcta. Para poder realizar este trabajo se requiere que el programa está correctamente documentado.

14. Conclusiones:
● Codificación: Es escribir la solución del problema en un lenguaje comprendido por la computadora.
● Definición y delimitación del Problema: Para resolver un problema es fundamental conocerlo por completo.

Los algoritmos pueden ser expresados de muchas maneras, incluyendo al lenguaje natural, pseudocódigo, diagramas de flujo y lenguajes de programación entre otros. Las descripciones en lenguaje natural tienden a ser ambiguas y extensas. El usar pseudocódigo y diagramas de flujo evita muchas ambigüedades del lenguaje natural. Dichas expresiones son formas más estructuradas para representar algoritmos; no obstante, se mantienen independientes de un lenguaje de programación específico.

La descripción de un algoritmo usualmente se hace en tres niveles:

1. **Descripción de alto nivel.** Se establece el problema, se selecciona un modelo matemático y se explica el algoritmo de manera verbal, posiblemente con ilustraciones y omitiendo detalles.

2. **Descripción formal.** Se usa pseudocódigo para describir la secuencia de pasos que encuentran la solución.
3. **Implementación.** Se muestra el algoritmo expresado en un lenguaje de programación específico o algún objeto capaz de llevar a cabo instrucciones.

También es posible incluir un teorema que demuestre que el algoritmo es correcto, un análisis de complejidad o ambos.

Lenguaje natural

Son aquellas lenguas que han sido generadas espontáneamente en un grupo de hablantes con propósito de comunicarse, a diferencia de otras lenguas, como puedan ser una lengua construida, los lenguajes de programación o los lenguajes formales usados en el estudio de la lógica formal, especialmente la lógica matemática.

Los algoritmos escritos en este lenguaje se caracterizan por seguir un proceso de ejecución común y lógico, describiendo textualmente paso a paso cada una de las actividades a realizar para resolver un problema determinado.

Los inconvenientes de escribir algoritmos en este lenguaje son: la ambigüedad, la no universalidad y la amplitud del mismo lo cual genera errores y no puede ser interpretado correctamente por el computador.

Pseudocódigo

Es una descripción de alto nivel de un algoritmo que emplea una mezcla de lenguaje natural con algunas convenciones sintácticas propias de lenguajes de programación, como asignaciones, ciclos y condicionales, aunque no está regido por ningún estándar. Es utilizado para describir algoritmos en libros y publicaciones científicas, y como producto intermedio durante el desarrollo de un algoritmo, como los diagramas de flujo, aunque presentan una ventaja importante sobre estos, y es que los algoritmos descritos en pseudocódigo requieren menos espacio para representar instrucciones complejas.

El pseudocódigo está pensado para facilitar a las personas el entendimiento de un algoritmo, y por lo tanto puede omitir detalles irrelevantes que son necesarios en una implementación. Programadores diferentes suelen utilizar convenciones distintas, que pueden estar basadas en la sintaxis de lenguajes de programación concretos. Sin embargo, el pseudocódigo, en general, es comprensible sin necesidad de conocer o utilizar un entorno de programación específico, y es a la vez suficientemente estructurado para que su implementación se pueda hacer directamente a partir de él.

Ejemplo

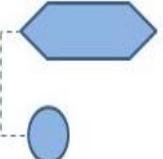
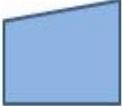
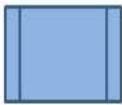
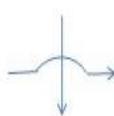
- El problema es calcular la suma de los números naturales desde 1 hasta **n**.
- **Inicio**
- Entero **i,n,Suma**
- Leer(**n**)
- **i=1**
- **Suma =0**
- **MIENTRAS (i<=n)**
 - **Suma=Suma+i**
 - **i=i+1**
- **Fin_mientras**
- Escribir (Suma)
- **Fin**

Diagramas de Flujo

Son descripciones gráficas de algoritmos; usan símbolos conectados con flechas para indicar la secuencia de instrucciones y están regidos por ISO.

Los diagramas de flujo son usados para representar algoritmos pequeños, ya que abarcan mucho espacio y su construcción es laboriosa. Por su facilidad de lectura son usados como introducción a los algoritmos, descripción de un lenguaje y descripción de procesos a personas ajenas a la computación.

SIMBOLOS Y CONVENCIONES

Inicio/Final 	Se utiliza para indicar el inicio y el final de un diagrama; del 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 General 	Entrada/Salida de datos en General(en esta guía, solo la usaremos para la Entrada).	Iteración (repetición) 	Indica que una instrucción o grupo de instrucciones deben ejecutarse varias veces..
Entrada por teclado 	Instrucción de 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.	Salida Impresa 	Indica la presentación de uno o varios resultados en forma impresa.
Llamada a subrutina 	Indica la llamada a una subrutina o proceso predeterminado.	Salida en Pantalla 	Instrucción de presentación de mensajes o resultados en pantalla
Acción/Proceso General 	Indica una acción o instrucción general que debe realizar el computador (cambios de valores de variables, asignaciones, operaciones aritméticas etc.).	Conector 	Indica el enlace de dos partes de un diagrama dentro de la misma página.
Flujo 	Indica el seguimiento lógico del diagrama. También indica el sentido de ejecución de las operaciones.	Conector 	Indica el enlace de dos partes de un diagrama en páginas diferentes.

ALGORITMOS

Un proceso de trabajo consiste en una serie de acciones que son necesarias para resolver una tarea definida.

El diagrama de flujo es una herramienta muy útil para mejorar los procesos de trabajo. Mapear e investigar los procesos es importante para descubrir cómo se puede mejorar.

Además, los diagramas de flujo son fáciles de crear y dado que las formas son simples y visuales, también son fáciles de entender.

¿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

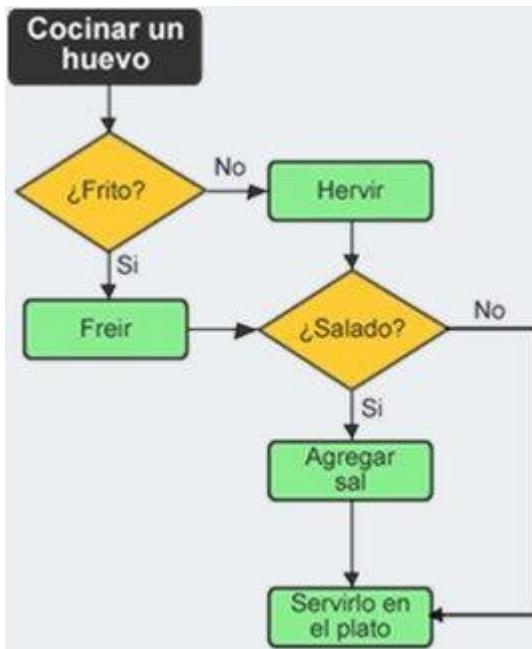
Normalmente para realizar un diagrama de flujo primero se hace lo que se llama el algoritmo. **Un algoritmo** es una **secuencia** de PASOS LÓGICOS a seguir para resolver un problema **de forma escrita**.

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 los siguientes:

SIMBOLOS FUNDAMENTALES



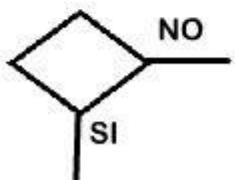
INICIO Y FIN DEL PROCESO



REALIZAR UN PROCESO (OPERACIÓN MATEMÁTICA POR EJEMPLO)



ENTRADA DE DATOS Y/O SALIDA DE DATOS



TOMAR UNA DECISIÓN (UNA PREGUNTA).
LA RESPUESTA A LA PREGUNTA
PUEDE SER SI O NO

- 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.

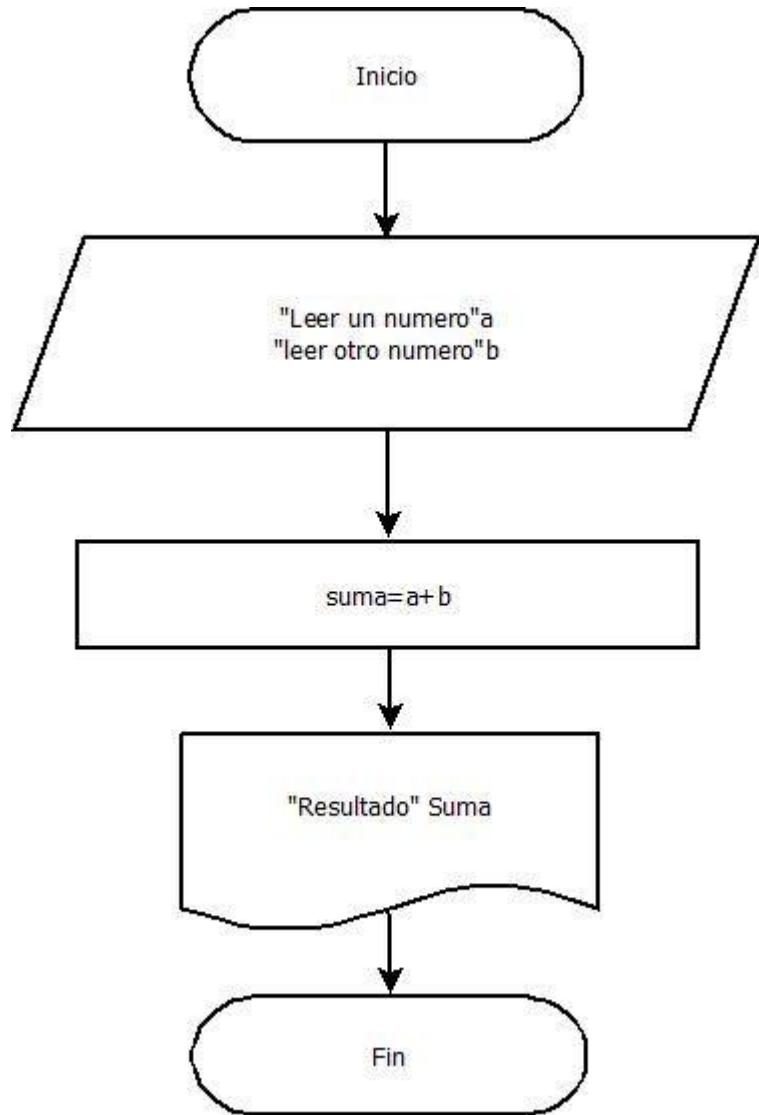
- Se pueden utilizar colores para los símbolos.

Ejemplos de Diagramas de Flujo

Veamos **un primer ejemplo** muy sencillo.

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

Solución del ejemplo:



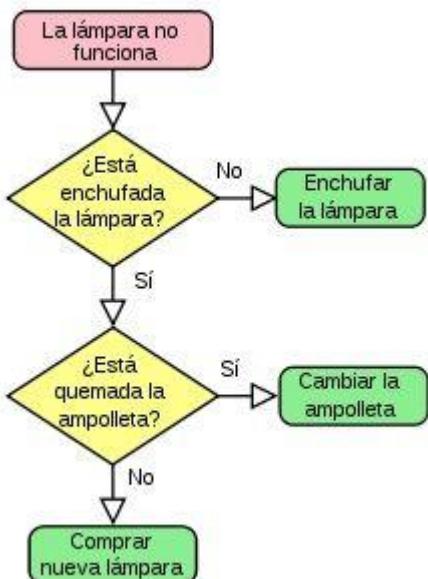
El símbolo de resultado es un símbolo usado en los diagramas para soluciones con el ordenador. Es el símbolo de salida del resultado por la pantalla del ordenador.

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

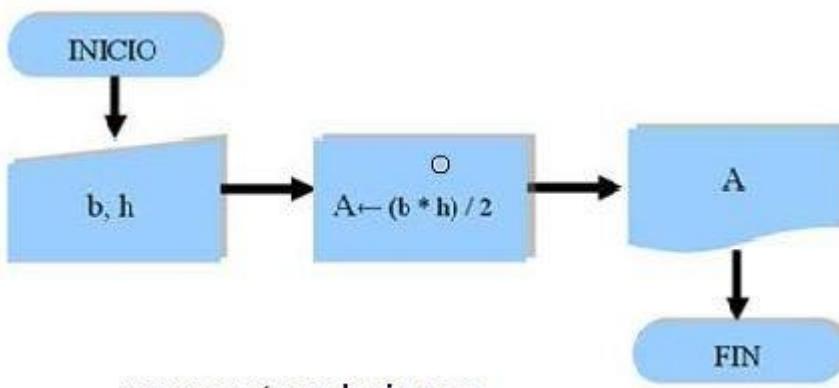
correspondiente en el diagrama.

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 qué hacer cuando la lámpara no funciona.



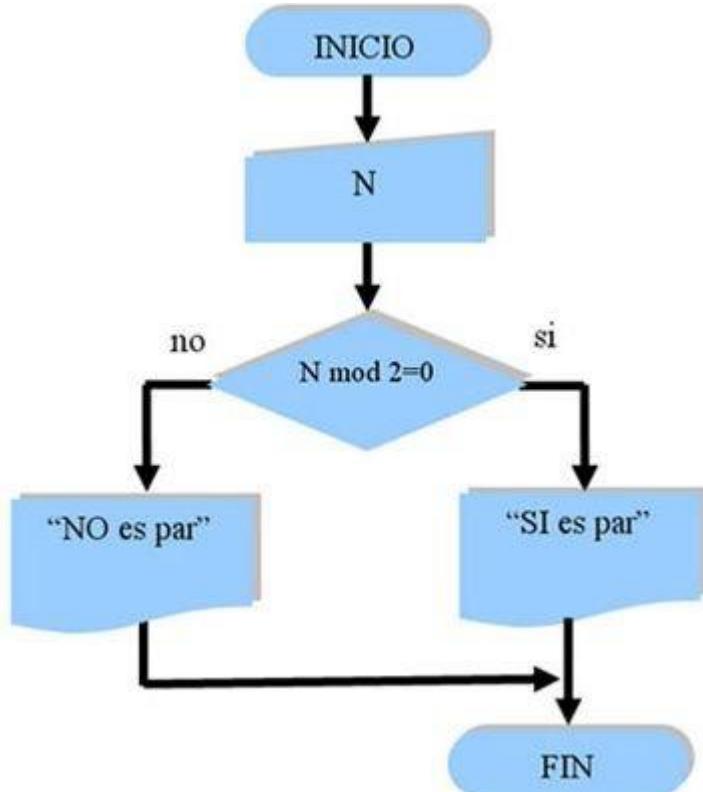
¿Hacemos otro?

Bueno vamos a 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 una 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. Bien pues hay esta la decisión.

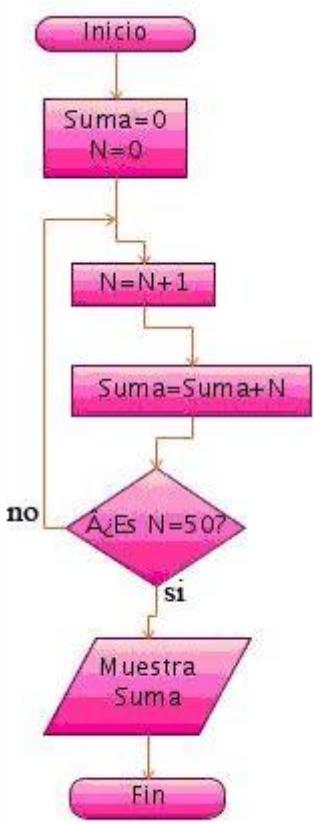
¿Al dividirlo entre 2 el resto es 0? Hay 2 posibilidades. Si lo es, se ve en pantalla "Si es par", si no lo es, se ve en pantalla "No es par". Eso es la toma de decisiones. Toma una salida en función del resultado de la entrada.

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 primeros 50 números.**

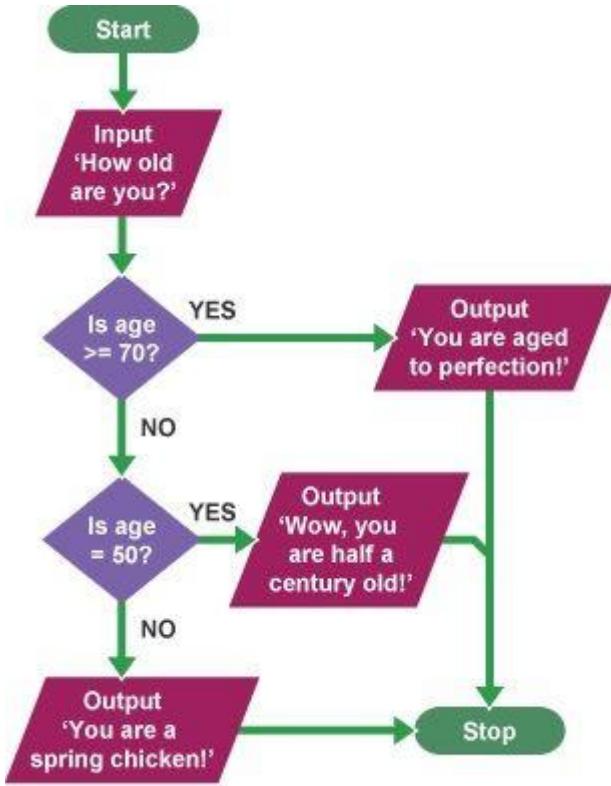
Lo primero es poner a cero la suma y dar el primer número a sumar que será el 0.

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.



www.areatecnologia.com

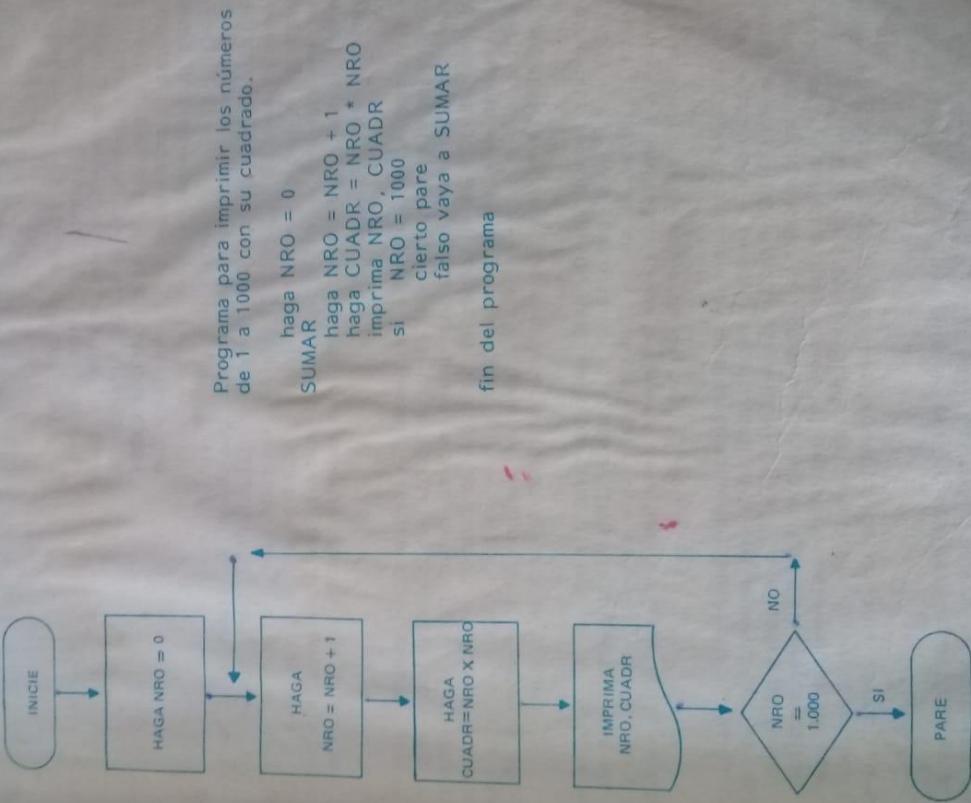
¿Ponemos un diagrama de flujo en inglés?

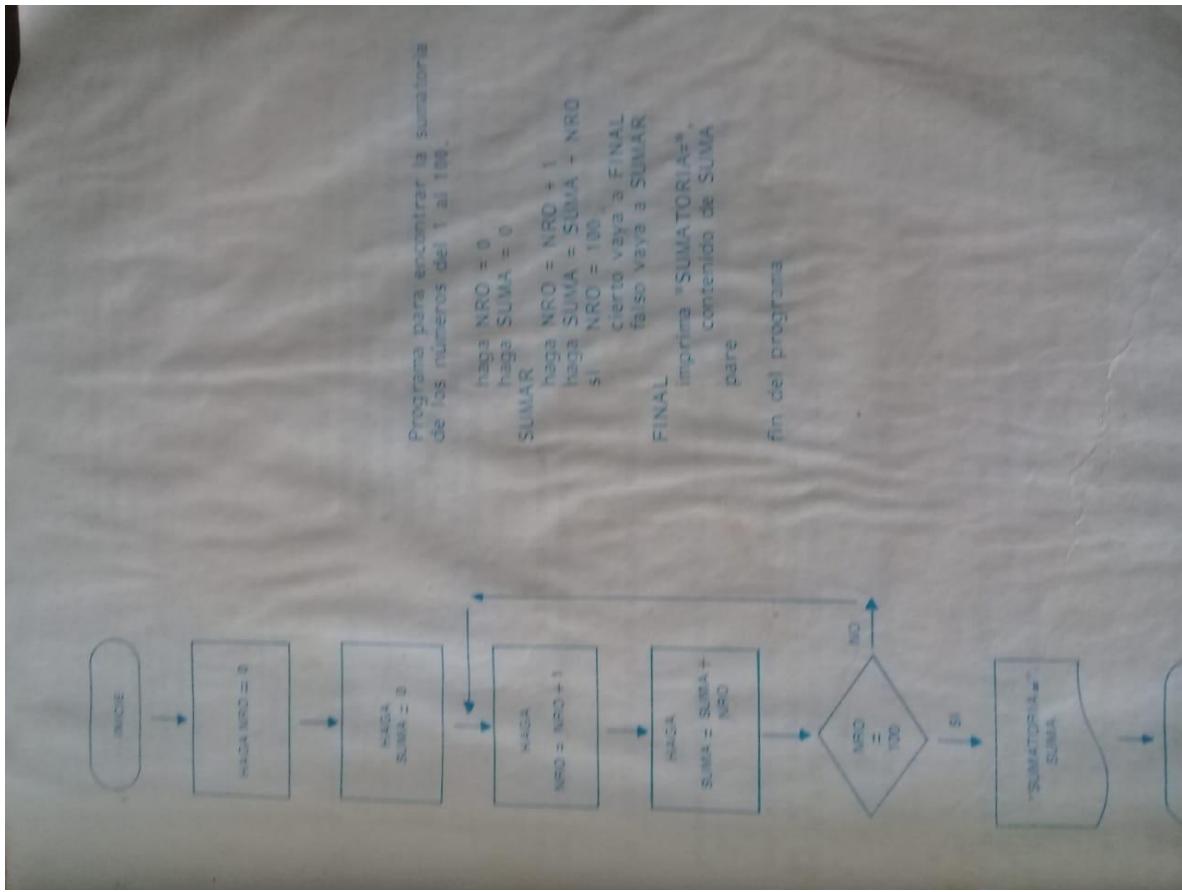


EJERCICIOS RESUELTOS.

Instrucción que determinaba si el programa volvía a repetir la evaluación de una condición o ya no la hacía más.

En la figura 2,2,4 podemos analizar una solución gráfica del problema.





demos
n que
e y la
nos lo
(tado)
e que
ermina

do los

erar e

nbre lo
ma. En
esentan
ICAS o
símbolos
imíamos
cas las

a figura

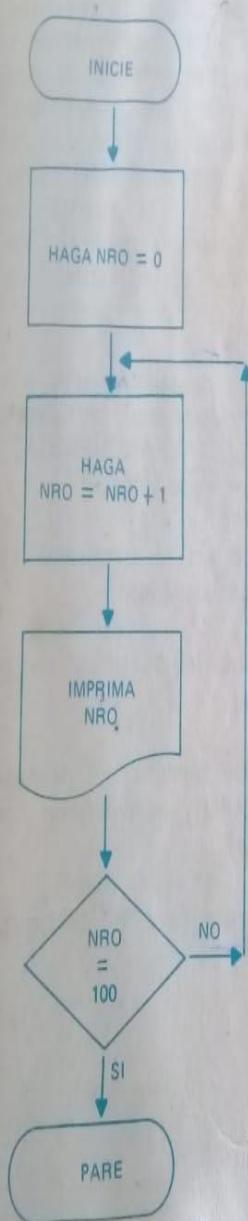


Fig. 2.2.2

Programa para imprimir los números
del 1 al 100.

haga NRO = 0
SUMAR
haga NRO = resultado de NRO + 1
imprima contenido de NRO
si NRO = 100
cierto pare
falso vaya a SUMAR

fin del programa

Programa para calcular el área
de un triángulo.

```
leia BASE, ALTURA
Haga AREA = resultado de ((BASE*ALTURA)/2)
imprima "RESULTADO=" y contenido de AREA
pare
```

fin del programa

```
INICIE
    LEA
        BASE, ALTURA
        HAGA
            AREA =
                BASE * ALTURA
                -----
                    2
IMPRIMA
    "RESULTADO = "
        AREA
PARE
```

Fig. 2.2.1