FUNDAMENTOS DE PROGRAMACIÓN

Resolución de problemas

Objetivo: El alumno resolverá problemas mediante la especificación algorítmica.

Contenido:

2.1 Definición, planteamiento y modelado del problema.

- **2.1.1** Formular el problema.
- **2.1.2** Analizar el problema.
- **2.1.3** Diseñar una estrategia de búsqueda de la solución.

- **2.2 Algoritmos para la resolución del problema.**
- **2.2.1** Definición y representación de algoritmos.
- **2.2.2** Conversión del planteamiento del problema al algoritmo.

- **2.3 Definición del modelo computacional.**
- **2.3.1** Máquina de Von Neumann.
- **2.3.2** Máquina de Turing.
- **2.3.3** Manejo del Sistema Binario

- **2.4 Refinamiento del algoritmo paso a paso.**
- **2.4.1** Planteamiento de la solución del problema.
- **2.4.2** Descomposición de la solución del problema en submódulos.
- **2.4.3** Aplicación de las estructuras básicas de control: secuencial, condicional e iterativo.

CICLO DE VIDA DE UN PROGRAMA

Al igual que en la resolución de problemas, existen ciertos pasos que debemos seguir para la creación de programas, estos son:

Análisis del problema
Elaborar el Algoritmo
Codificación del algoritmo
Depuración de código
Mantenimiento de programa

Análisis del problema

Consiste en estudiar minuciosamente el problema que queremos solucionar, considerando los requerimientos que se piden y los elementos con los que contamos para realizarlo, etc.

Elaborar el Algoritmo

El algoritmo son las instrucciones para resolver el problema.
Puede ser de texto (Pseudocódigo) o gráfico (Diagrama de Flujo).

De manera más específica:
Un algoritmo es el conjunto de instrucciones que emplean **estructuras de control y** nos permiten realizar un programa que ejecute una o varias actividades especificas. Para elaborarlo, debe pensarse detallada y ordenadamente todos los pasos que realizará el programa.

Codificación del algoritmo

Después de tener el algoritmo, el siguiente paso es codificarlo en el lenguaje que seleccionamos, para este caso en específico es el **Lenguaje "C"**.

Depuración de código

Después de codificar el programa, generalmente quedan pequeños detalles a corregir, por lo tanto, el siguiente paso es la depuración del programa.

Depurar; es pulir el programa para que todo funcione como nosotros deseamos. Para esto, es necesario probar el programa cuantas veces sea necesario, para asegurarnos de que funcione correctamente.

Mantenimiento de programa.

Este es el último paso del ciclo de vida de un programa. Se realiza después de que el programa ya ha tenido vida útil y debido a las nuevas necesidades de los usuarios, es necesario hacer modificaciones al programa.

Cuando estos cambios son demasiados, se debe elaborar un nuevo programa.

- El software es una parte esencial de sistemas convencionales y de tecnologías de la información, tales como sistemas de transporte, militares, médicos y financieros.
- Hay una proliferación de normas, procedimientos, métodos, herramientas y entornos para desarrollar y gestionar el software. Esta proliferación ha creado dificultades en la gestión y en la ingeniería de software, especialmente en la integración de productos y servicios. La disciplina del software necesita evolucionar desde esta proliferación, hacia un marco de referencia común que pueda ser usado por los profesionales del software para "hablar el mismo lenguaje", a la hora de crear y gestionar el software.
- La NTP-ISO/IEC 12207 tiene como objetivo principal proporcionar una estructura común para que compradores, proveedores, desarrolladores, personal de mantenimiento, operadores, gestores y técnicos involucrados en el desarrollo de software usen un lenguaje común.

Ingeniería de Software

En la construcción y desarrollo de proyectos se aplican métodos y técnicas para resolver los problemas, la informática aporta herramientas y procedimientos sobre los que se apoya la ingeniería de software:

- Mejorar la calidad de los productos de software.
- Aumentar la productividad y trabajo de los ingenieros del software.
- Facilitar el control del proceso de desarrollo de software.
- Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.
- Definir una disciplina que garantice la producción y el mantenimiento de los productos de software, desarrollados en el plazo fijado y dentro del costo estimado.

Método de ”Ciclo de Vida Clásico”

El método de ciclo de vida de un programa, para el desarrollo de sistemas, es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implementar un sistema de información.

El método del ciclo de vida para el desarrollo de sistemas consta de las siguientes actividades:

- 1) Investigación preliminar.
- 2) Determinación de los requisitos del sistema.
- 3) Diseño del sistema.(diseño lógico).
- 4) Desarrollo de software (diseño físico).
- 5) Prueba de sistemas.
- 6) Implantación y evaluación.

DIAGRAMAS DE FLUJO

DEFINICIÓN:

Es la representación gráfica de las secuencias lógicas, que se realizan para la resolución de un problema (algoritmo).

Elementos esenciales:

Comienzo del diagrama (parte superior)

- **Operaciones**
- **Secuencia en que se realizan**

Fin del diagrama (parte inferior)

Recomendación adicionales:

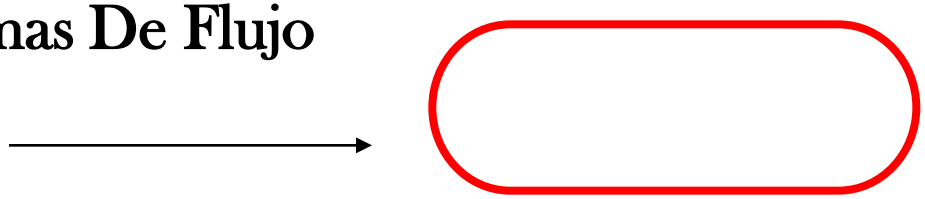
El símbolo de inicio y fin deben estar sólo una vez.

El flujo de las secuencias deben de ir de arriba-abajo

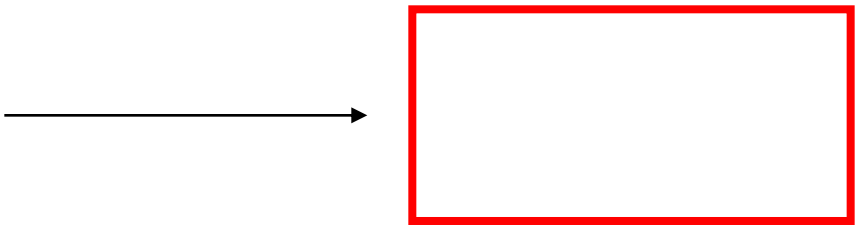
Evitar cruces de líneas de flujo.

Elementos de Diagramas De Flujo

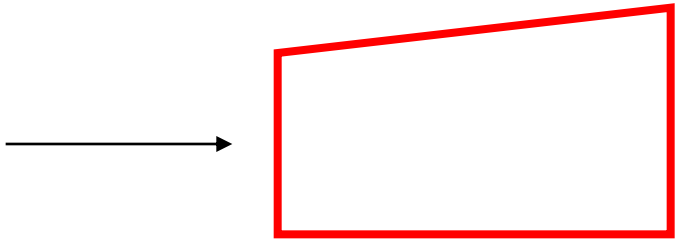
Inicio y fin



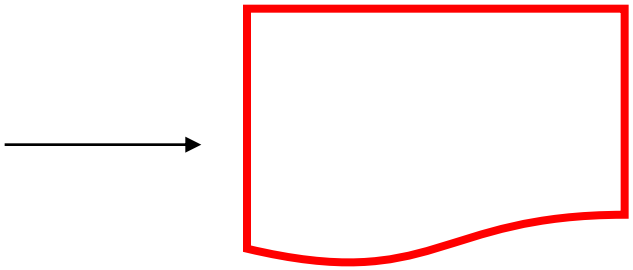
Proceso



Entrada de Datos



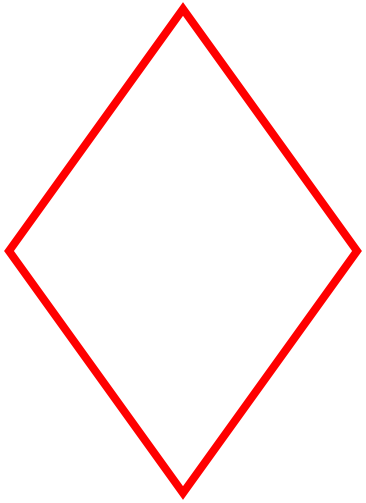
Salida de Datos



conector

Flujo de datos

Elemento de Decisión



OPERADORES ARITMÉTICOS:

SUMA	+
RESTA	-
DIVISIÓN	/
MULTIPLICACIÓN	*
POTENCIA	**
RESIDUO	**mod**
PARÉNTESIS	**()**
OPERADOR DE ASIGNACIÓN	=

RECUERDEN QUE LOS OPERADORES TIENEN UNA JERARQUÍA

PSEUDOCÓDIGO

DEFINICIÓN:

Es la representación ESCRITA de las secuencias lógicas, que se realizan para la resolución de un problema (algoritmo).

Elementos esenciales:

Comienzo al inicio -parte superior-

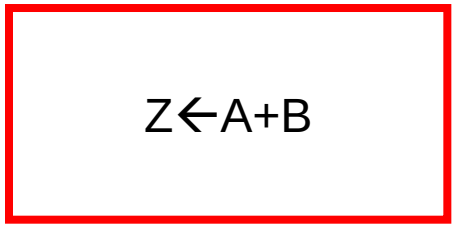
Operaciones

Secuencia en que se realizan

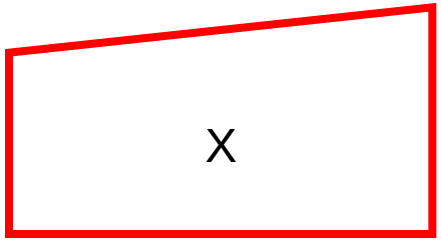
Fin del algoritmo -parte inferior-

Elementos Escritos

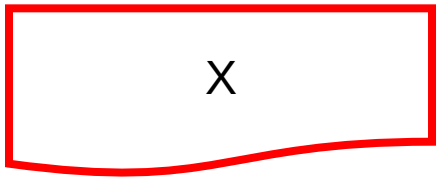
INICIO



ASIGNAR Z←A+B



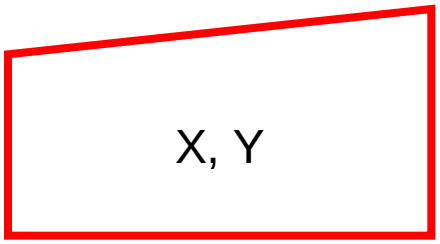
LEER X



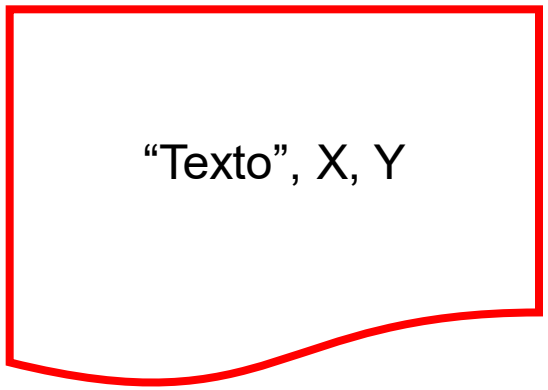
MOSTRAR X

Elementos Escritos

FIN



LEER X, Y



MOSTRAR “TEXTO”, X, Y

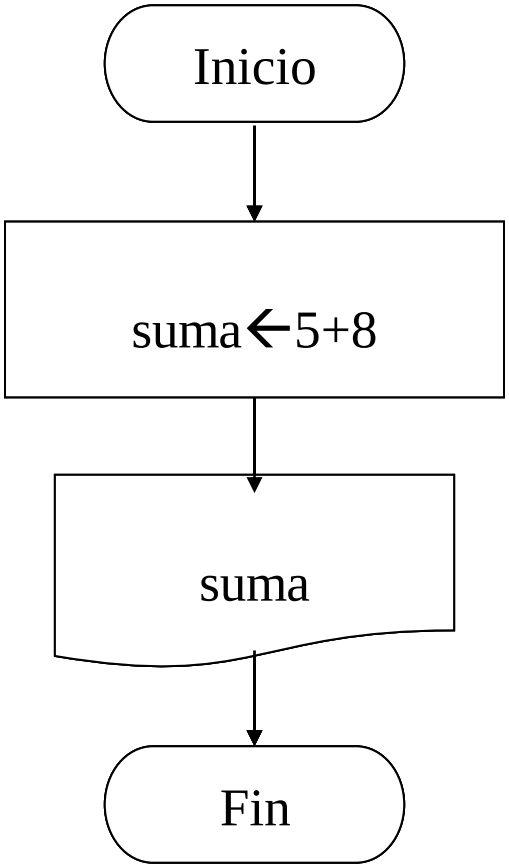
Asignaciones:

Ejer. Obtener la suma de dos constantes de tipo numérico.

Pseudocódigo
Nombre de Algoritmo: Suma
Variables:
suma, real
Inicio
Asignar suma←5+8
Mostrar suma
Fin

Ejer. Obtener la suma de dos constantes de tipo numérico.

Diagrama de Flujo



url´s:

http://unfviso12207.webcindario.com/

http://www.monografias.com/trabajos99/articulo-ntp-iso-iec-12207/articulo-ntp-iso-iec-12207.shtml

http://www.monografias.com/trabajos93/desarrollo-software-sistema-informacion/desarrollo-software-sistema-informacion.shtml

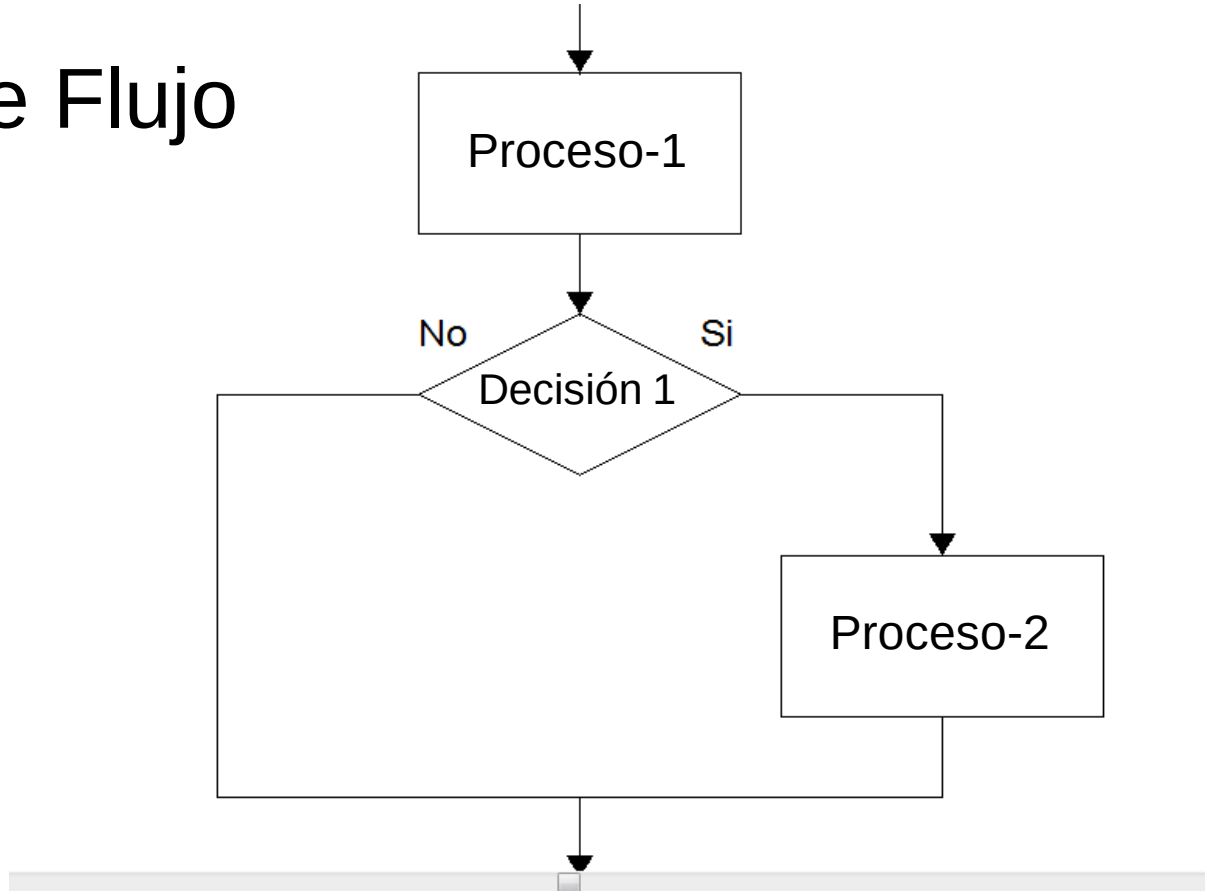
http://www.monografias.com/trabajos5/inso/inso.shtml

Variables

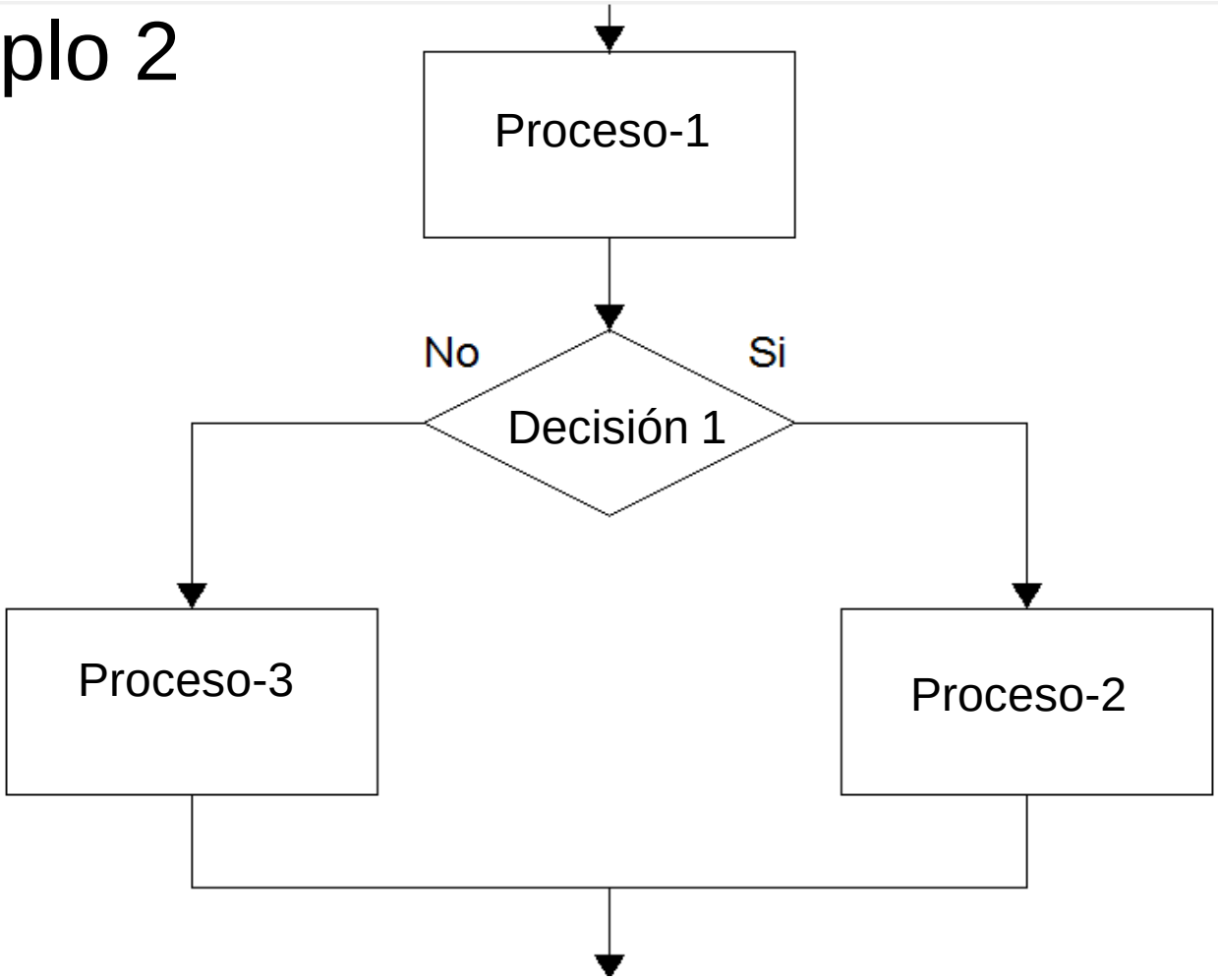
- Tipo de Datos
  - Numéricos
    - Enteros
    - Reales
  - Alfanuméricos
    - Caracteres
  - Datos lógicos
    - Booleanos

Decisiones

- Diagrama de Flujo
- Ejemplo 1



- Diagrama de Flujo
- Ejemplo 2



Decisiones

- Pseudocódigo
- Ejemplo 1

```
ASIGNAR PROCESO 1
SI DECISIÓN 1 (ES VERDADERA) ENTONCES
        ASIGNAR PROCESO 2
FIN DE DECISION
```

- Pseudocódigo
- Ejemplo 2

```
ASIGNAR PROCESO 1
SI DECISIÓN 1 (ES VERDADERA) ENTONCES
        ASIGNAR PROCESO 2
SI NO
        ASIGNAR PROCESO 3
FIN DE DECISION
```

Ejer. Leer un número X e imprimir si es cero o no.

Pseudocódigo

Nombre de Algoritmo: Num_Cero

Variables:

num, entero

Inicio

Mostrar “Ingrese un número entero”

Leer NUM

Si NUM=0 entonces

Mostrar “El número es cero”

Fin decisión

Fin

OPERADORES RELACIONALES

MAYOR	>
MENOR	<
IGUAL	=
MAYOR E IGUAL	>=
MENOR E IGUAL	<=
DIFERENTE	<>

OPERADORES LÓGICOS

AND (Y)
OR (Ó)

Comparación de tres variables
A=B=C incorrecto

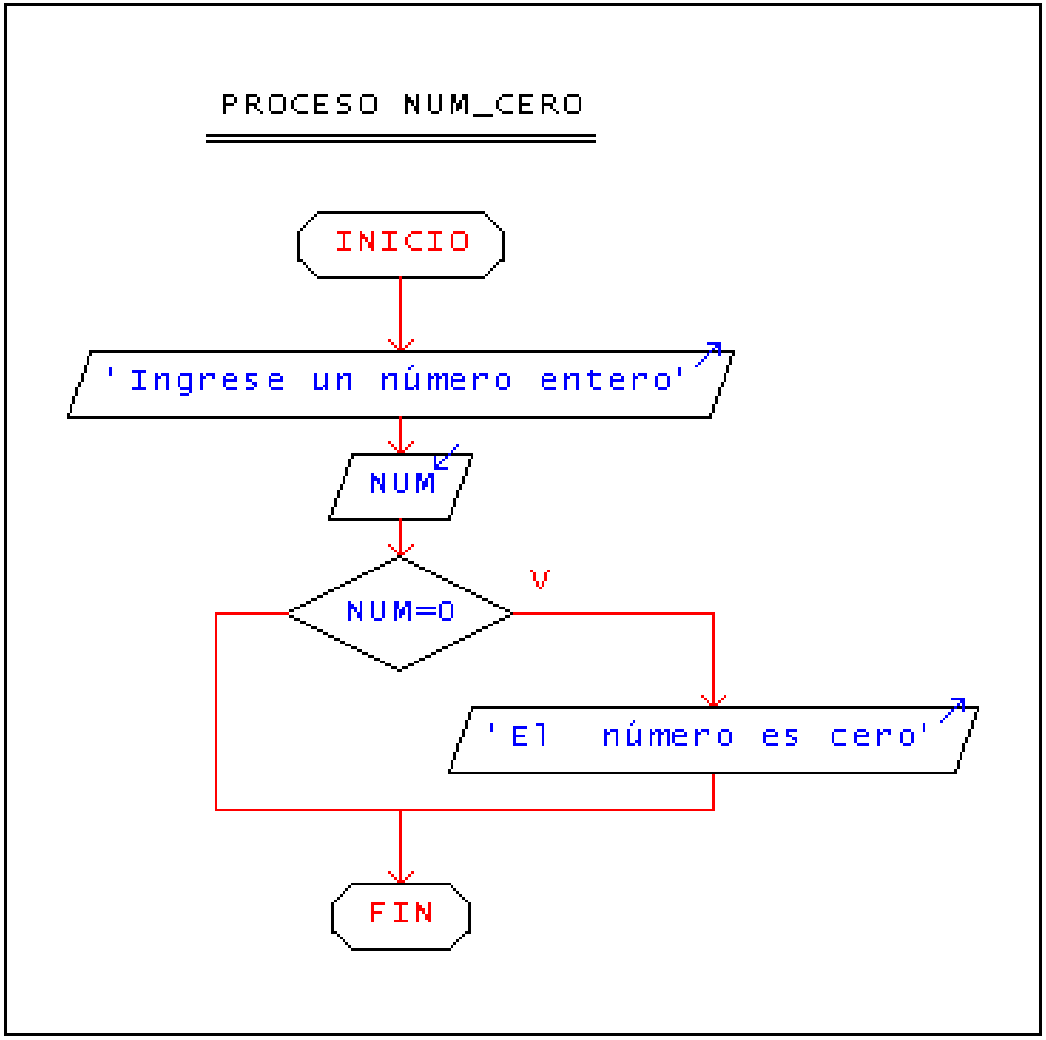
A=B y B=C correcto

O bien

A=B ó B=C ó A=C correcto

Ejer. Leer un número X e imprimir si es cero o no.

Diagrama de Flujo



Estructura de repetición: Mientras

- Diagrama de Flujo
- Ejer.

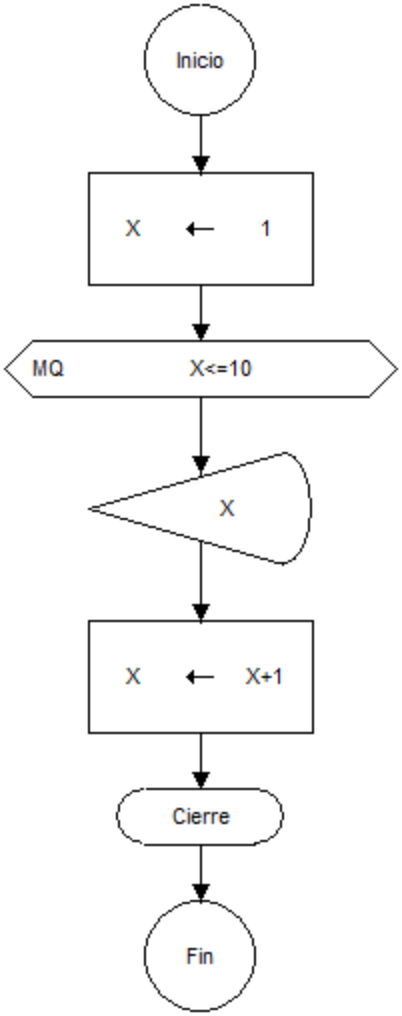
Mostrar en pantalla

La numeración de 1 al 10,

Es decir: 1, 2, 3,…10

Usar una sola variable

y un ciclo mientras.



Estructura de repetición: Mientras

- Diagrama de Flujo
- Ejer.

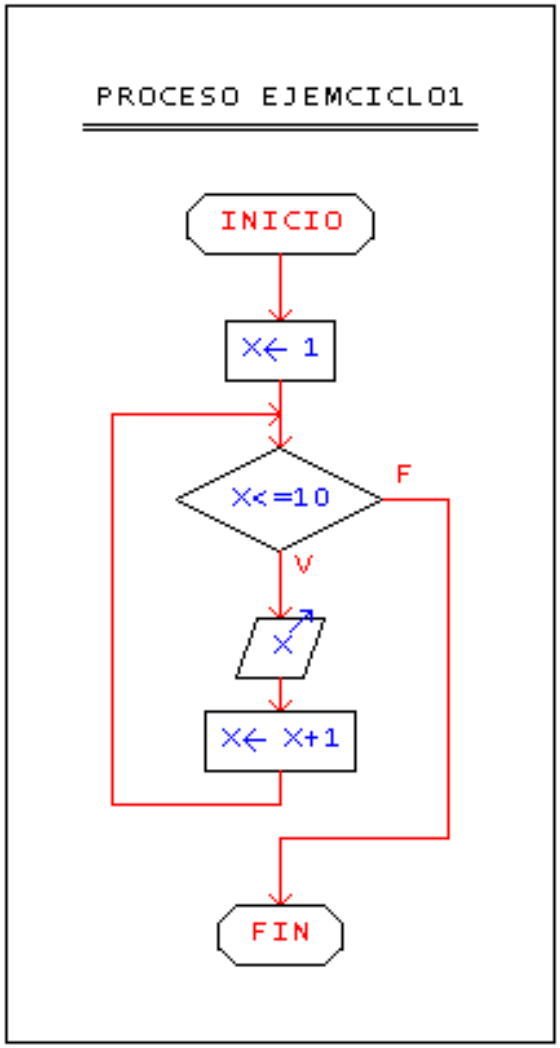
Mostrar en pantalla

La numeración de 1 al 10,

Es decir: 1, 2, 3,…10

Usar una sola variable

y un ciclo mientras.



Estructura de repetición: Mientras

- Pseudocódigo
- Ejer.

Mostrar en pantalla
La numeración de 1 al 10,
Es decir: 1, 2, 3,…10
Usar una sola variable
y un ciclo mientras.

Nombre de algoritmo: ciclo1

Variables: X, entero

Inicio

asignar X←1

mientras X<=10 repetir

mostrar X

asignar X←X+1

fin ciclo

Fin

Estructura de repetición: Mientras

- Pseudocódigo
- Ejer.

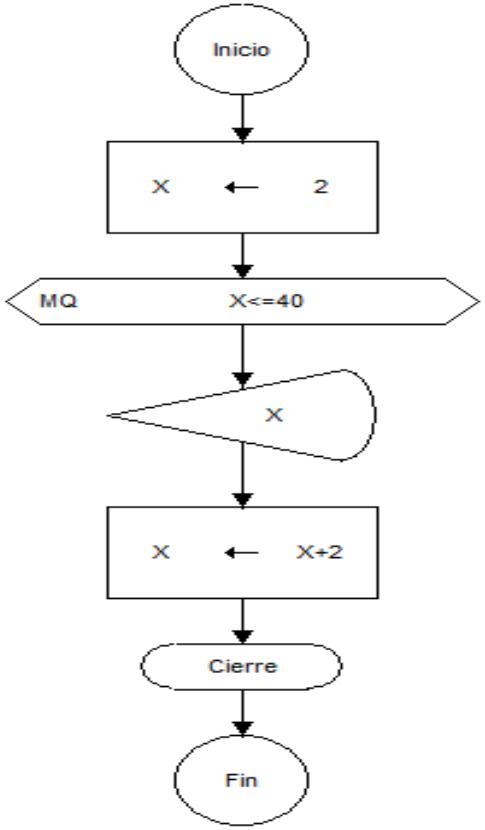
Mostrar en pantalla
La numeración de 1 al 10,
Es decir: 1, 2, 3,…10
Usar una sola variable
y un ciclo mientras.

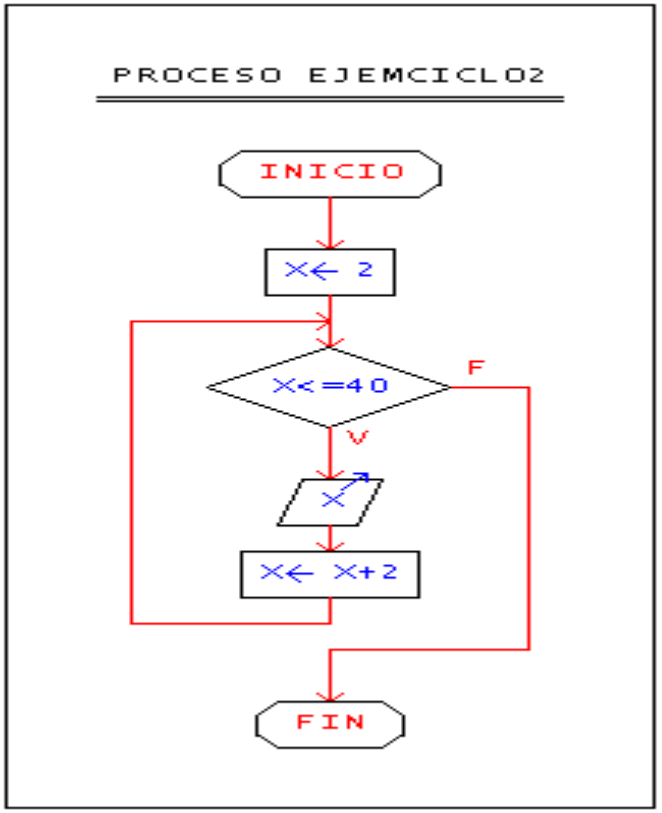
```
Proceso EjemCiclo1
        Definir X como entero;
        X<-1;
        Mientras X<=10 Hacer
                Escribir X;
                X<-X+1;
        FinMientras

FinProceso
```

Ejer:

Generar e imprimir los primeros 20 números pares. Usar una variable y un ciclo mientras.





Ejer:

Generar e imprimir los primeros 20 números pares. Usar una variable y un ciclo mientras.

Nombre del algoritmo: Ciclo2
variables: X, entero
Inicio
    Asignar X←2
    mientras X<=40 repetir
        mostrar X
        asignar X←X+2
    fin de ciclo
Fin

Ejer:

Generar e imprimir los primeros 20 números pares. Usar una variable y un ciclo mientras.

```
Proceso EjemCiclo2
        Definir X como entero;
        X<-2;
        Mientras X<=40 Hacer
                Escribir X;
                X<-X+2;
        FinMientras

FinProceso
```

Tipos de Variables

- **Contador:** Se utiliza para llevar la cuenta de determinadas acciones que se pueden solicitar durante la resolución de un problema.

- **Acumulador:** Su tarea es almacenar cantidades variables.

- La principal diferencia entre ambas es que el incremento o decremento de cada suma es ***variable*** en lugar de un valor ***constante***, como en el caso del contador.

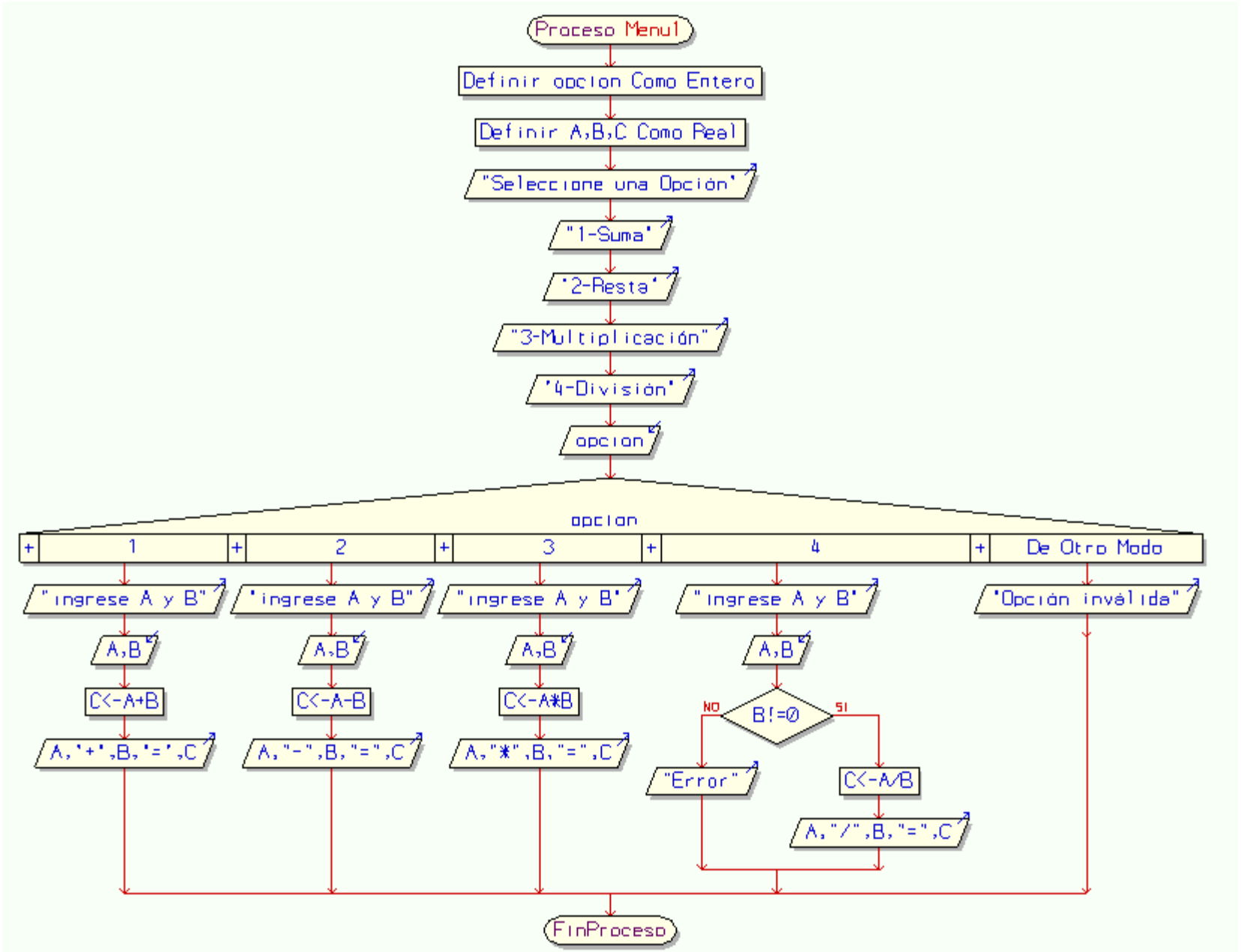
Estructura de Selección Múltiple

- Proceso Ejercicio

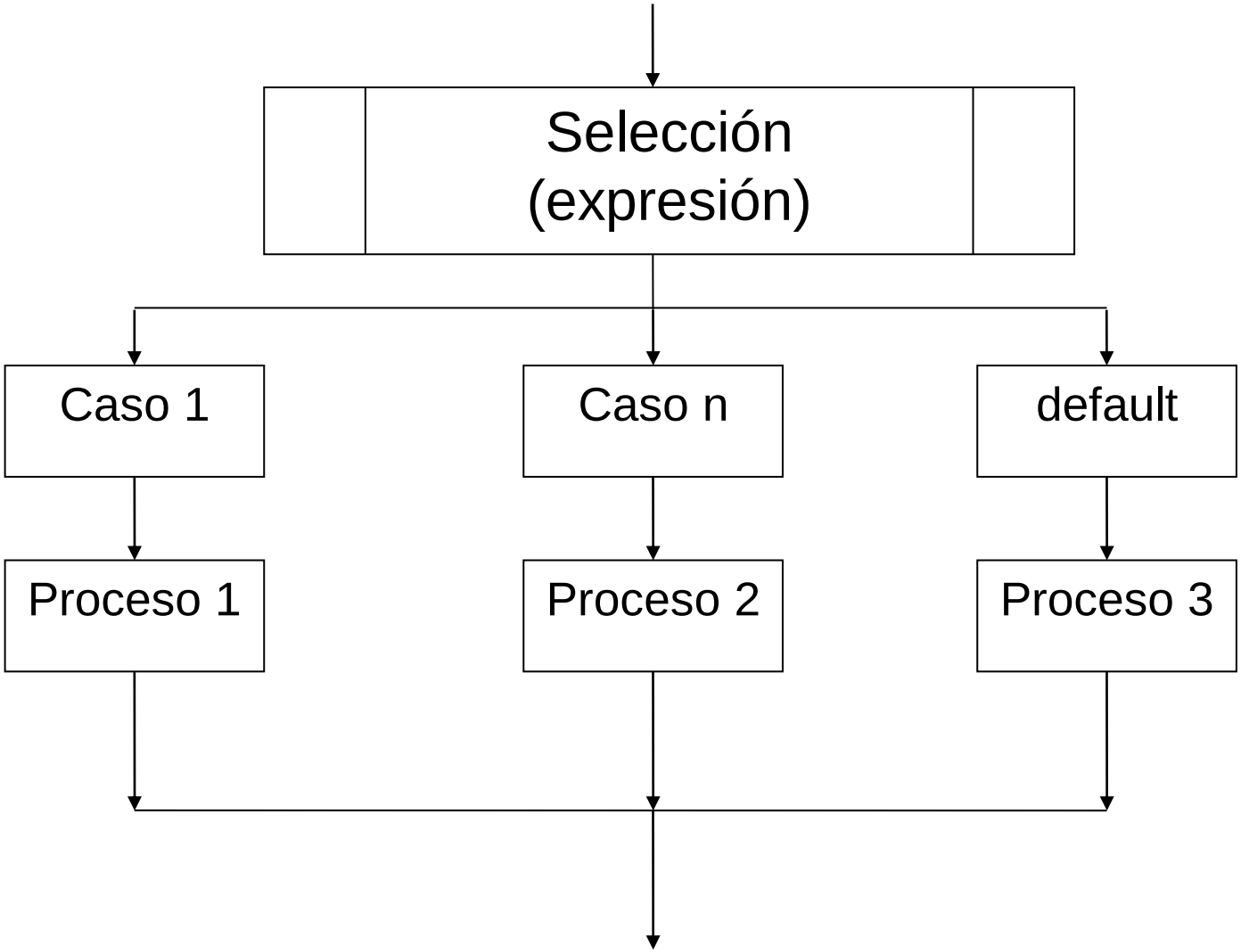
```
Definir opcion como entero
Definir A, B, C como real
Escribir 'Seleccione una Opción'
Escribir '1-Suma'
Escribir '2-Resta'
Escribir '3-Multiplicación'
Escribir '4-División'
Leer opcion
```

```
Segun opcion Hacer
        1:      Escribir 'ingrese A y B'
                Leer A,B
                C<-A+B
                Escribir A,'+',B,'=',C
        2:      Escribir 'ingrese A y B'
                Leer A,B
                C<-A-B
                Escribir A,'-',B,'=',C
        3:      Escribir 'ingrese A y B'
                Leer A,B
                C<-A*B
                Escribir A,'*',B,'=',C
```

```
            4:      Escribir 'ingrese A y B'
                    Leer A,B
                    Si B<>0 Entonces
                            C<-A/B
                            Escribir A,'/',B,'=',C
                    Sino
                            Escribir 'Error'
                    Fin Si
                De Otro Modo:
                    Escribir 'Opción inválida'
        Fin Segun
FinProceso
```


Proceso Menu1
Definir opcion Como Entero
Definir A,B,C Como Real
"Seleccione una Opción"
"1-Suma"
"2-Resta"
"3-Multiplicación"
"4-División"
opcion
opcion
1
2
3
4
De Otro Modo
"ingrese A y B"
"ingrese A y B"
"ingrese A y B"
"ingrese A y B"
"Opción inválida"
A,B
A,B
A,B
A,B
C<-A+B
C<-A-B
C<-A*B
B!=0
NO
SI
A,"+",B,"=",C
A,"-",B,"=",C
A,"*",B,"=",C
"Error"
C<-A/B
A,"/",B,"=",C
FinProceso

ELEMENTO DE SELECCIÓN MULTIPLE



PSEUDOCÓDIGO PARA ESTRUCTURA DE SELECCIÓN

```
SI SELECTOR IGUAL
        VALOR 1: HACER ACCIÓN 1
        VALOR 2: HACER ACCIÓN 2
        VALOR 3: HACER ACCIÓN 3
        VALOR 4: HACER ACCIÓN 4
         DE OTRA FORMA: HACER ACCIÓN N
FIN DEL CONDICIONAL (SELECTOR)
```

SERIE DE DISEÑO DE ALGORITMOS

EJERCICIOS DE CLASE

OBTENER EL DIAGRAMA DE FLUJO Y EL PSEUDOCÓDIGO, PARA LOS SIGUIENTES INCISOS, MOSTRAR LOS RESULTADOS SOLICITADOS.