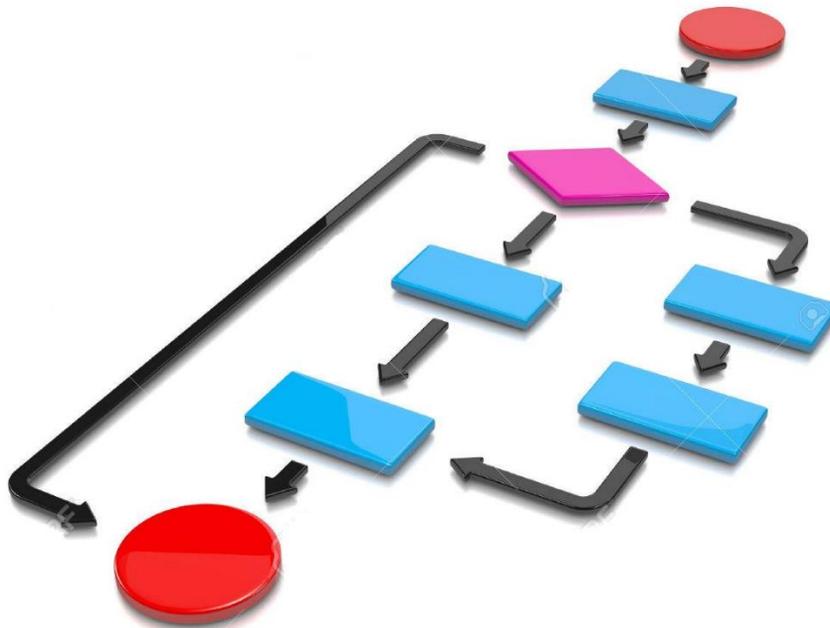




# TECNOLOGÍA E INFORMÁTICA

## Módulo II

### Grado 10°



2020

## Guía 5

### Las Variables

#### ¿Qué es una variable?

El término variable es utilizado en varios campos de las ciencias como son las matemáticas y su rama de la estadística y en ciencias de la computación. En este caso vamos a centrarnos en la rama de las ciencias de la computación pero la idea que hay detrás de cada una de estas ciencias no es muy distinta.

En computación, una Variable es un espacio de la memoria del computador que permite almacenar información de un determinado tipo de dato.

El tipo de dato indica como es el dato que se almacena en la variable, en programación los tipos de datos básicos son los numéricos, los carácter y los lógicos.



#### Asignar un dato a una variable

Para asignar un dato o un valor a una variable se utiliza el símbolo igual "=", por ejemplo:

**base** = 20

La variable **base** tendrá un valor de 20

**altura** =4

La variable **altura** tendrá un valor de 4

**area** = (base \* altura)/2

La variable **area** tendrá el valor del resultado de la operación correspondiente, en este caso 40

#### Variable tipo numérico

Las variables de tipo numérico se utilizan para almacenar valores numéricos, ya sean enteros o reales con los cuales se pueden realizar operaciones aritméticas como la suma, resta, multiplicación, división entre otras.

#### Variable tipo carácter

Las variables de tipo carácter se usan para almacenar uno solo o varios caracteres como los que forman una palabra o una frase, estos deben estar encerrados entre comillas dobles o simples, con el fin de que el lenguaje de programación las pueda reconocer como variables de tipo carácter.

Desde el punto de vista de la programación, si no se ponen restricciones al alfabeto, una cadena podrá estar formada por cualquier combinación finita de todo el juego de caracteres disponibles (las letras de la 'a' a la 'z' y de la 'A' a la 'Z', los números del '0' al '9', el espacio en blanco ' ', símbolos diversos '!', '@', '%', entre otros). Un caso especial de cadena es la que contiene cero caracteres, a esta cadena se le llama cadena vacía.

Algunas operaciones con cadenas de caracteres:

- **Asignación:** Consiste en asignarle una cadena a otra.
- **Concatenación:** Consiste en unir dos cadenas o más (o una cadena con un carácter) para formar una cadena de mayor tamaño.
- **Comparación:** Se utiliza para comparar dos cadenas.

## Variables lógicas

Las variables de tipo lógicas se conocen como boolean o booleano y permiten almacenar uno de los dos estados lógicos (verdadero o falso).

Los nombres de Variables se conocen también como identificadores, un identificador es una secuencia de caracteres de letras, dígitos o subrayados “\_”, en donde el primer carácter debe ser una letra o un subrayado y después de este pueden ir otras letras, subrayados o dígitos. Las letras mayúsculas y minúsculas son diferentes.

## Jerarquía de Operadores

En programación una expresión es una fórmula aritmética que permite calcular un valor, cuando se construye la formula se debe tener en cuenta la jerarquía de operadores. La jerarquía de operadores determina el orden en el que se resuelven las expresiones cuando se involucran operaciones aritméticas como la suma, resta, multiplicación, división, potencia, raíz y módulo de la división. Este orden es el que permite que una expresión aritmética cualquiera siempre tenga la misma interpretación ya sea resolviéndola en papel, en calculadora o en el computador.

Operador	Descripción
()	Paréntesis
^	Exponenciación
/ y *	División y multiplicación
+ y -	Suma y resta

## Operadores aritméticos y su prioridad

Prioridad	Operador	Significado	Ejemplo
1	()	Paréntesis	$(2 + 3) * 5 = 25$
2	^	Potenciación	$4 ^ 2 = 16$
	√	Radicación	$\sqrt{9} = 3$
	MOD	Residuo de la división	$5 \text{ MOD } 2 = 1$
3	*	Multiplicación	$2 * 4 = 8$
	/	División	$5 / 2 = 2.5$
4	+	Suma	$3 + 4 = 7$
	-	Resta	$8 - 5 = 3$

## Ejemplo

$$\begin{aligned} X &= 6 + 4 / 2 \\ X &= 6 + 2 \\ X &= 8 \end{aligned}$$

Cuando dos operadores tienen el mismo nivel de prioridad, dentro de una expresión se evalúan de izquierda a derecha.

$$\begin{aligned} A &= 2 + 2 \wedge 2 * 2 + 2 \text{ MOD } 2 \\ A &= 2 + 4 * 2 + 2 \text{ MOD } 2 \\ A &= 2 + 8 + 2 \text{ MOD } 2 \\ A &= 2 + 8 + 0 \\ A &= 10 \end{aligned}$$

Cuando se desea asignar un orden específico de ejecución en una expresión aritmética, se debe emplear los paréntesis para agrupar, de esta manera, las operaciones que se encuentren dentro del paréntesis serán las primeras en ejecutarse.

$$\begin{aligned} B &= 5 * (7 + 3) / 2 + 4 \\ B &= 5 * 10 / 2 + 4 \\ B &= 50 / 2 + 4 \\ B &= 25 + 4 \\ B &= 29 \end{aligned}$$

## Actividad

1. Asigna un tipo de variable a los campos del siguiente formulario de datos de empleados de una empresa, justifica tu respuesta.

Campo	Tipo de variable	Justificación
<b>Nombres</b>		
<b>Dirección</b>		
<b>Salario</b>		
<b>Activo</b>		

2. Considera las siguientes variables tipo caracter y su valor para realizar las siguientes operaciones, escriba los resultados:

<b>Nombres</b>	Julio Cesar
<b>Apellidos</b>	Londoño Posada
<b>Ciudad</b>	La Llanada

<b>Barrio</b>	El Progreso
<b>Calle</b>	12
<b>NroCasa</b>	7-64

- a. A la variable **Nombres** se le concatena la variable **Apellidos**

<b>Nombres</b>	
----------------	--

- b. En la variable **DirUsuario** se concatenan las variables Ciudad, Calle, NroCasa y Barrio

<b>DirUsuario</b>	
-------------------	--

3. Solucione los siguientes ejercicios teniendo en cuenta la prioridad de los operadores aritméticos.

$$x = 7+3*4-3*2+1*5$$

$$y = (7+3)*4-3*(2+1*5)$$

$$a = (77+3)/2 - 2*(4/2*5)$$

$$b = 100 \bmod (2^5 + \sqrt{9-5})$$

$$z = x*y + (a-x)^b$$

$$w = a \bmod x + (y \bmod 5)$$

$$t = (\sqrt{9+7} + 3 * 2) \bmod 8$$

$$p = (2+5)^2 - \sqrt{(100 \bmod 75)} + 6$$

$$j = (13 \bmod 7)^2 + (2^4 + 10)$$

$$k = (2^5 \bmod 5)^3 + 5 * 4 / 5$$

## Guía 6

### Leer un dato (entrada) ¿Qué es una entrada?

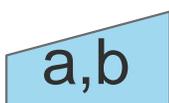
En programación, leer o entrar un dato hace referencia a la captura de un dato (normalmente por el teclado) y almacenarlo en una variable para posteriormente realizar con él operaciones que permitan obtener resultados.

En diagramas de flujo de datos para indicar una entrada se representa de la siguiente manera:



Significa que vamos a leer una entrada desde teclado en una variable llamada **a**

Para representar en el mismo símbolo más de una entrada, se procede como se muestra a continuación:



Significa que vamos a leer dos entradas desde teclado, una en una variable llamada **a** y otra en una variable llamada **b**

### Escribir un dato (salida) ¿Qué es una salida?

En programación, escribir hace referencia a mostrar un mensaje o el contenido de una variable generalmente en la pantalla, o también, en un medio impreso. Una salida generalmente hace referencia al resultado final esperado luego de haber realizado una serie de operaciones.

En diagramas de flujo de datos para indicar una salida en pantalla se representa de la siguiente manera:



Significa que vamos a mostrar en pantalla el valor que está almacenado en una variable llamada **x**

Para representar en el mismo símbolo más de una salida en pantalla se procede como se muestra a continuación:



Significa que vamos a mostrar en pantalla el valor que está almacenado en una variable llamada **x**, y el valor que está almacenado en una variable llamada **y**

En diagramas de flujo de datos para indicar una salida en medio impreso se representa de la siguiente manera:



Significa que vamos a mostrar en un documento impreso el valor que está almacenado en una variable llamada **x**

Para representar en el mismo símbolo más de una salida en pantalla se procede como se muestra a continuación:



Significa que vamos a mostrar en medio impreso el valor que está almacenado en una variable llamada **x**, y el valor que está almacenado en una variable llamada **y**

## Transformar un dato (proceso) ¿Qué es un proceso?

En programación, un proceso hace referencia a realizar operaciones con los datos para transformarlos y obtener resultados, las operaciones pueden ser, por ejemplo, matemáticas o de asignación directa.

En diagramas de flujo de datos para indicar un proceso se representa de la siguiente manera:

$$x = a + b$$

Significa que la variable **x** va a almacenar el resultado de la operación matemática de la suma de los valores que contienen las variables **a** y **b**. En este caso se asigna un valor mediante una operación matemática.

$$x = 5$$

Significa que a la variable **x** va a almacenar el valor que se le asigna (en este caso 5), en este caso es una asignación directa.

### Actividad

1. Solucione los siguientes ejercicios teniendo en cuenta la jerarquía de operadores.

$$a = 18 + 5 * 3 + 4 * 6 = \underline{\hspace{2cm}}$$

$$b = 5 * 2 / 2 * 2 + 6 * 4 / 2 * 2 - 4 * 6 * 2 / 2 * 3 / 2 = \underline{\hspace{2cm}}$$

$$c = 3 ^ 2 * 4 / 5 ^ 2 - 4 * 6 * 2 / 3 * 2 + 5 * 4 / 2 * 7 = \underline{\hspace{2cm}}$$

$$d = 4 * 2 - 5 * 6 / 2 * 3 + 4 * 3 / 6 * 2 - 6 * 2 / 2 * 3 * 2 = \underline{\hspace{2cm}}$$

$$e = 2 * 3 - 4 * 5 / 6 * 2 + 3 * 4 / 5 * 6 - 6 * 5 - 4 * 3 / 2 * 4 = \underline{\hspace{2cm}}$$

$$f = 8 * 7 * 6 / 5 * 4 * 3 + 5 * 6 / 3 * 2 - 6 * 5 / 4 * 3 = \underline{\hspace{2cm}}$$

$$g = 9 / ( 1 + 2 ) * 2 - 4 / 2 + 5 = \underline{\hspace{2cm}}$$

$$h = 10 - 2 * 3 / 2 + ( 6 * 2 ) / ( 2 + 2 ) * 2 + 5 = \underline{\hspace{2cm}}$$

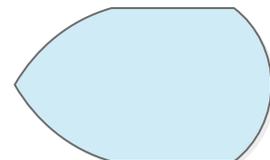
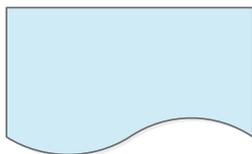
$$i = 10 / 2 * 2 * 3 + 6 + ( 8 / ( 6 - 2 ) * 10 ) = \underline{\hspace{2cm}}$$

$$j = 5 * 6 + 7 - 10 / 2 - 4 * ( 2 + 6 ) / 2 = \underline{\hspace{2cm}}$$

2. Con los resultados anteriores, solucione las siguientes expresiones. Represente la variable "Promedio" como una salida impresa y la variable "Sumatoria" como una salida en pantalla.

$$\text{Sumatoria} = a + b + c + d + e + f + g + h + i + j \quad \rightarrow \text{Sumatoria} = \underline{\hspace{2cm}}$$

$$\text{Promedio} = (a + b + c + d + e + f + g + h + i + j) / 10 \quad \rightarrow \text{Promedio} = \underline{\hspace{2cm}}$$



## Guía 7

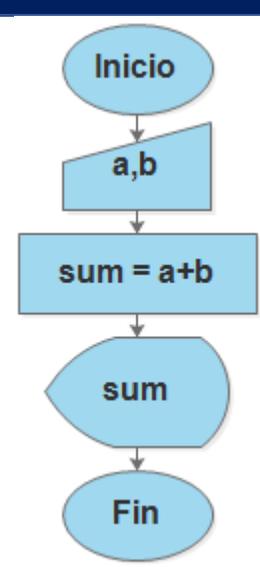
### Diseño de un diagrama de flujo de datos (DFD)

Para diseñar un DFD, se debe tener muy claro el algoritmo que da solución al problema que se plantea. Una vez se tenga el algoritmo, se identifica los símbolos que permitan representarlo gráficamente, es entonces donde se procede a diseñar el DFD asegurándose que sea acorde al algoritmo.

#### Ejemplo 1

Diseñar un diagrama que permita obtener la suma de dos números ingresados por el usuario y muestre su resultado en pantalla.

Con el problema anterior, procedemos a realizar en primer lugar el algoritmo y luego diseñamos el DFD.

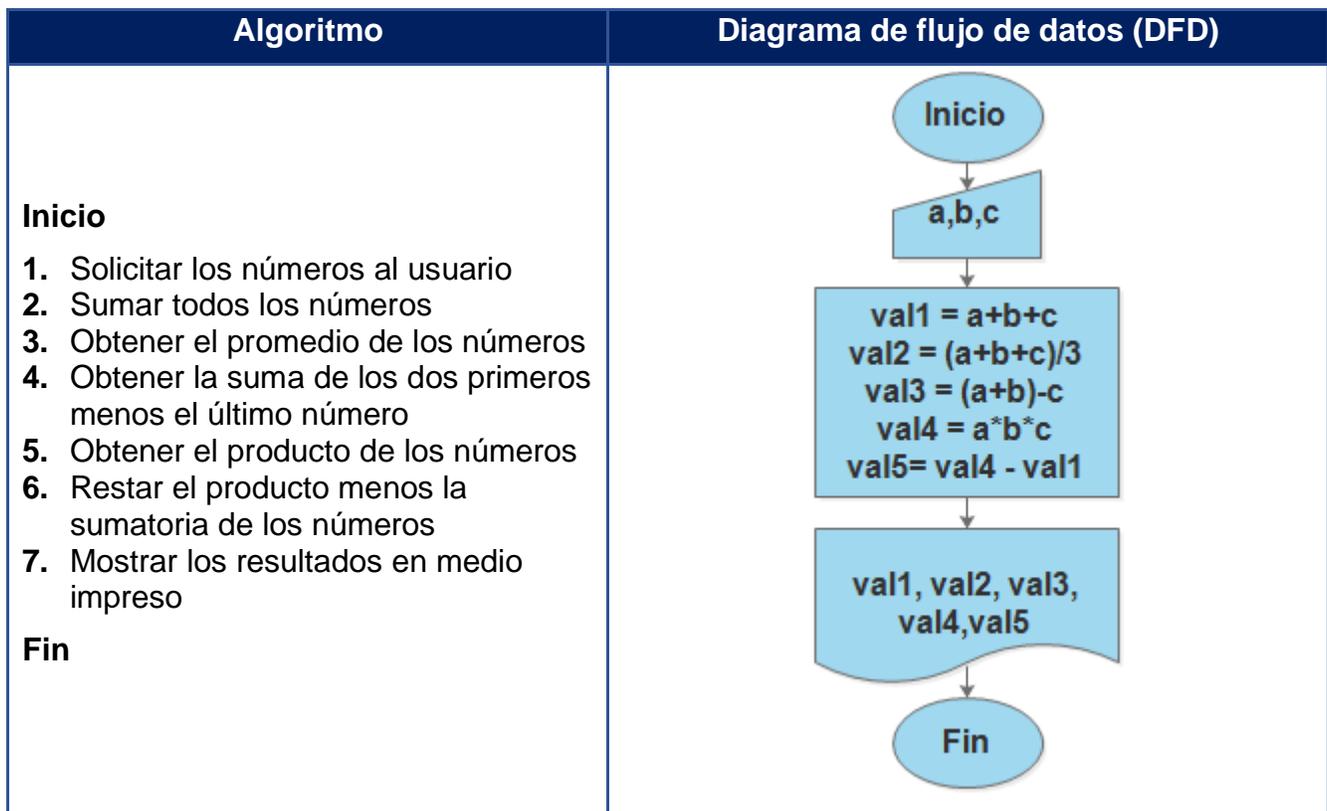
Algoritmo	Diagrama de flujo de datos (DFD)
<p><b>Inicio</b></p> <ol style="list-style-type: none"><li>1. Solicitar el primer número al usuario</li><li>2. Solicitar el segundo número al usuario</li><li>3. Sumar los dos números</li><li>4. Mostrar el resultado en pantalla</li></ol> <p><b>Fin</b></p>	 <pre>graph TD; Inicio([Inicio]) --&gt; Entrada[/a,b/]; Entrada --&gt; Procesamiento[sum = a+b]; Procesamiento --&gt; Salida{{sum}}; Salida --&gt; Fin([Fin]);</pre>

Como podemos darnos cuenta, en el diagrama se usaron las variables **a** y **b** para almacenar los números ingresados desde teclado por el usuario, el resultado de la suma de los dos números se almacena en la variable **sum**, por lo tanto, esta variable es la que se muestra en pantalla.

#### Ejemplo 2

Diseñar un diagrama de flujo de datos que permita a partir de tres (3) números ingresados por el usuario obtener los siguientes resultados y mostrarlos en medio impreso:

- La sumatoria de todos los números.
- El promedio de los números.
- La suma de los dos primeros números menos el último número.
- El producto o multiplicación de los tres números
- El producto menos la sumatoria de los números



En este diagrama se usaron las variables **a**, **b** y **c** para almacenar los valores ingresados por el usuario y se utilizaron las variables **val1**, **val2**, **val3**, **val4** y **val5** para almacenar los resultados de las operaciones que se solicitan y al final se muestran en medio impreso estas variables.

En **val5** se toma la variable **val4** y se le resta **val1** porque **val4** contiene el producto de los números y **val1** contiene la sumatoria, así evitamos volver a realizar esas operaciones.

También se usaron paréntesis en algunos procesos para garantizar que el resultado sea el esperado, esto teniendo en cuenta la jerarquía de los operadores.

**Actividad**

1. Coloque al frente de cada expresión el resultado que se obtenga.

Expresión	Resultado
$(\sqrt{10 + 15} + 4 + 5) * 2 - 3^2 + 4 \div 2 + 5$	
$\sqrt{10 + 15} + (4 + 5 * 2) - 3^2 + 4 \div 2 + 5$	
$(\sqrt{10 + 15} + 4) + 5 * 2 - (3^2 + 4 \div 2) + 5$	

Que puedes concluir del anterior ejercicio: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_



## Guía 8

### Estructuras de control

Las estructuras de control permiten determinar la secuencia de ejecución de las sentencias o instrucciones de un programa. En programación existen tres tipos de estructuras de control, la secuencia, las de decisión (también llamadas selectivas o condicionales) y las repetitivas. La estructura secuencial permite ejecutar las instrucciones de un programa en forma de secuencia, y se ejecutan una después de la otra.

Las estructuras de decisión (condicionales) determinan si se ejecutan un grupo de instrucciones u otras según se cumpla o no una determinada condición.

Las estructuras repetitivas permiten ejecutar de forma repetida un bloque específico de instrucciones.

### Manejo de operadores en programación

Los operadores en programación se dividen en aritméticos, relacionales y lógicos.

#### Operadores aritméticos

Se usan para realizar operaciones de la matemática básica como la suma, resta, multiplicación, división.

Operador	Símbolo	Acción
Suma	+	Suma dos operandos
Resta	-	Resta el segundo operando del primero
Multiplicación	*	Multiplica los dos operandos
División	/	Divide el primer operando por el segundo

#### Operadores relacionales

Los operadores relacionales se usan para evaluar expresiones condicionales y se basan en el concepto de verdadero o falso.

Operador	Símbolo	Ejemplo
Mayor que	>	8 > 4 Verdadero 2 > 4 Falso
Mayor o igual que	>=	4 >= 4 Verdadero 2 >= 4 Falso
Menor que	<	2 < 6 Verdadero 9 < 4 Falso
Menor o igual que	<=	6 <= 6 Verdadero 8 <= 4 Falso
Igual	=	7 = 7 Verdadero 5 = 7 Falso
Diferente de	!=	2 != 7 Verdadero 3 != 3 Falso

## Operadores lógicos

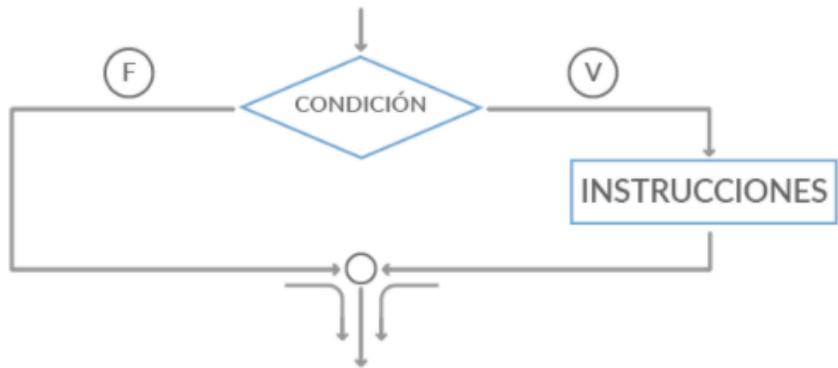
Los operadores lógicos son usados para concatenar dos o más expresiones con operadores relacionales.

Operador	Símbolo
AND	&&
OR	
NOT	!

## Estructura de decisión SI simple (if)

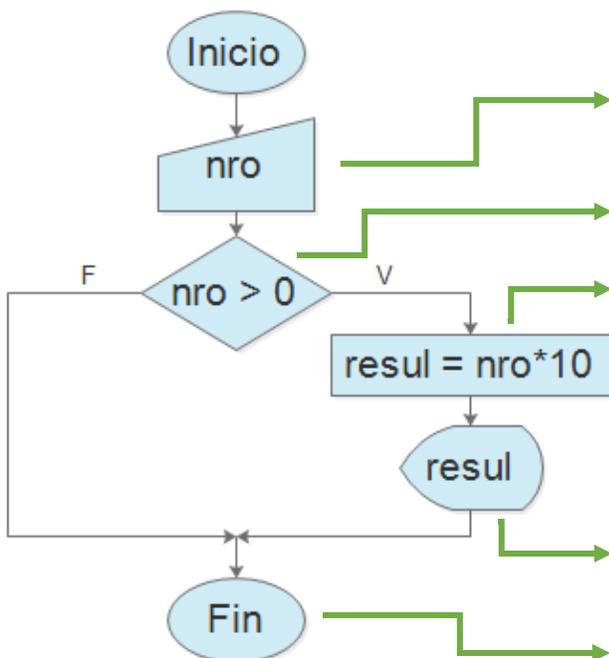
En este tipo de estructura, se evalúa la condición y si ésta se cumple se ejecuta un determinado grupo de instrucciones. En caso contrario se saltan dicho grupo de instrucciones.

Para evaluar la condición y saber cuál es el resultado de ésta, se utilizan los operadores relacionales. En una estructura de decisión cuando la respuesta es verdadera, el flujo de los datos se dirige por la derecha, y si la respuesta es falsa, el flujo de los datos se dirige hacia la izquierda.



Cuando la estructura condicional se termina, el flujo de los datos continúa hacia abajo.

**Ejemplo:** Diseñar un DFD que permita ingresar un número, si el número ingresado es mayor que 0, lo multiplique por 10 y muestre el resultado en pantalla.



En este DFD podemos darnos cuenta que el número que se ingresó se almacena en la variable **nro**.

Luego este valor se compara en la condición usando el operador relacional **Mayor que (>)**.

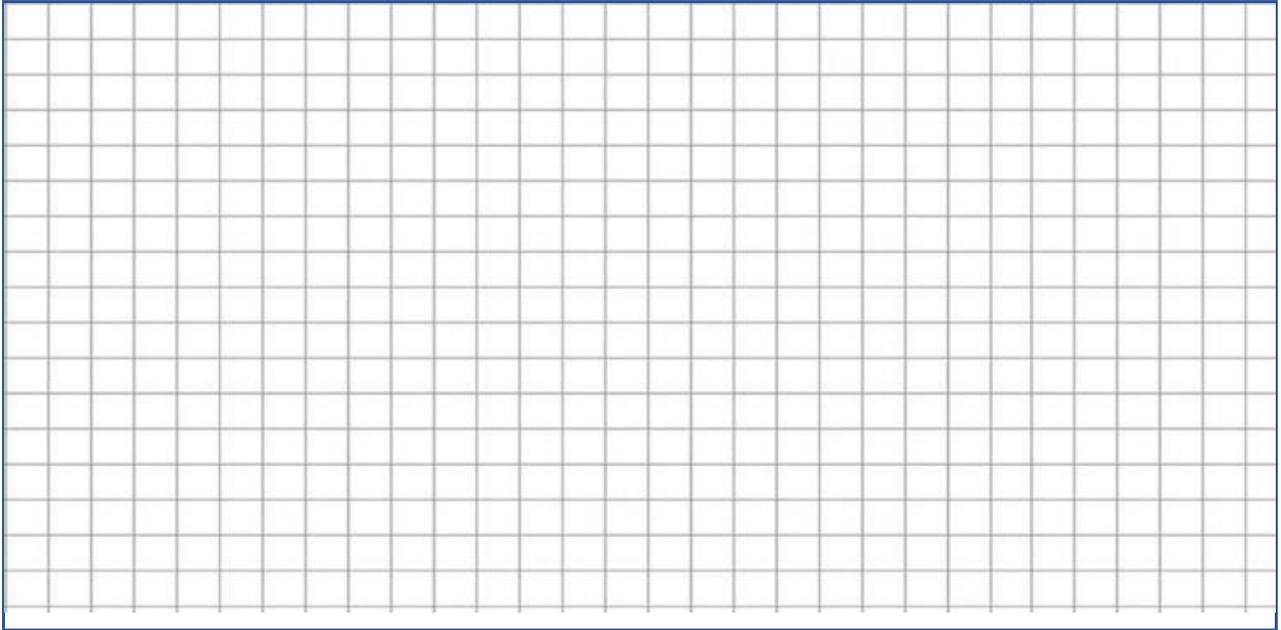
Cuando la condición de que el número sea mayor que 10 se cumpla, es decir sea verdadera, el flujo de los datos continúa por la derecha y se utiliza una variable llamada **resul** para almacenar el valor que resulta de multiplicar el número ingresado por 10, como solicita el ejemplo.

Una vez hecha esa operación se procede a mostrar en pantalla el resultado obtenido que se almacenó en la variable **resul**.

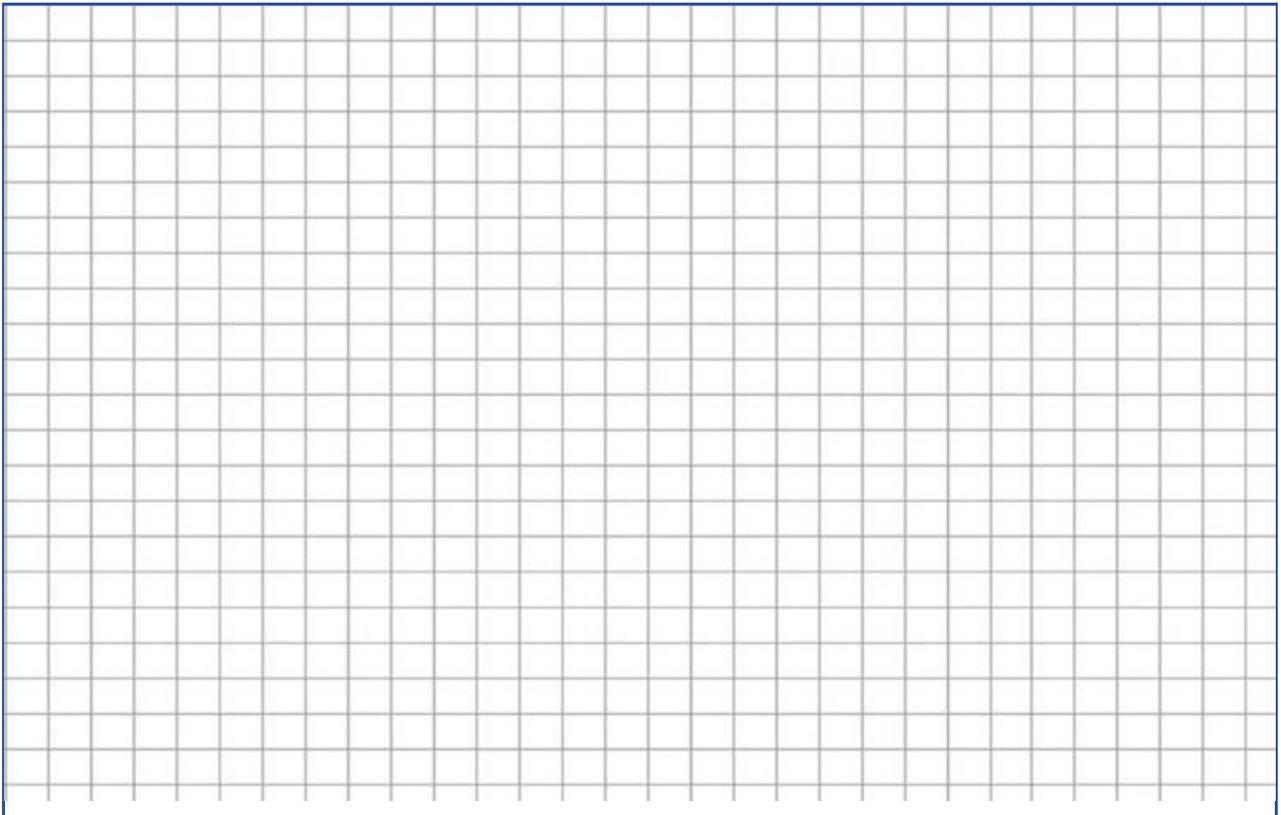
Se finaliza el flujo de los datos

## Actividad

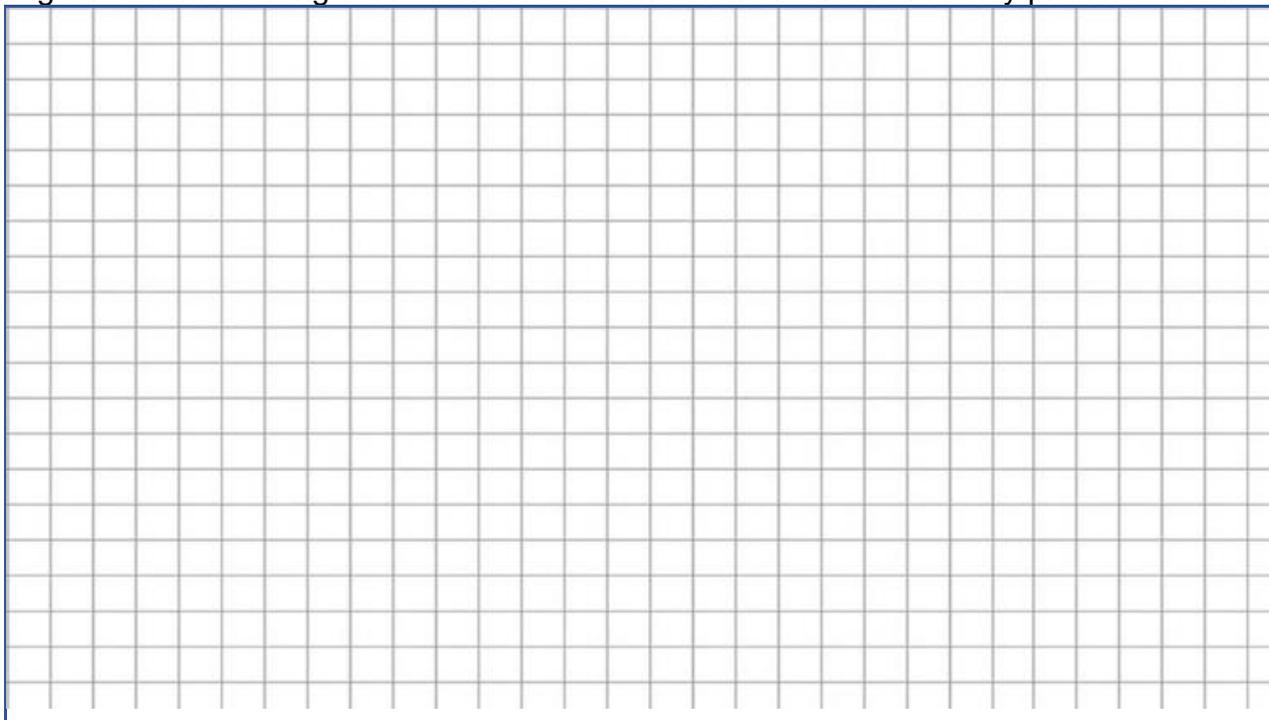
1. Realice un DFD que permita a partir de un número obtener el resultado de elevarlo a la tercera potencia, siempre y cuando el número sea menor que 10. El resultado se mostrará en medio impreso



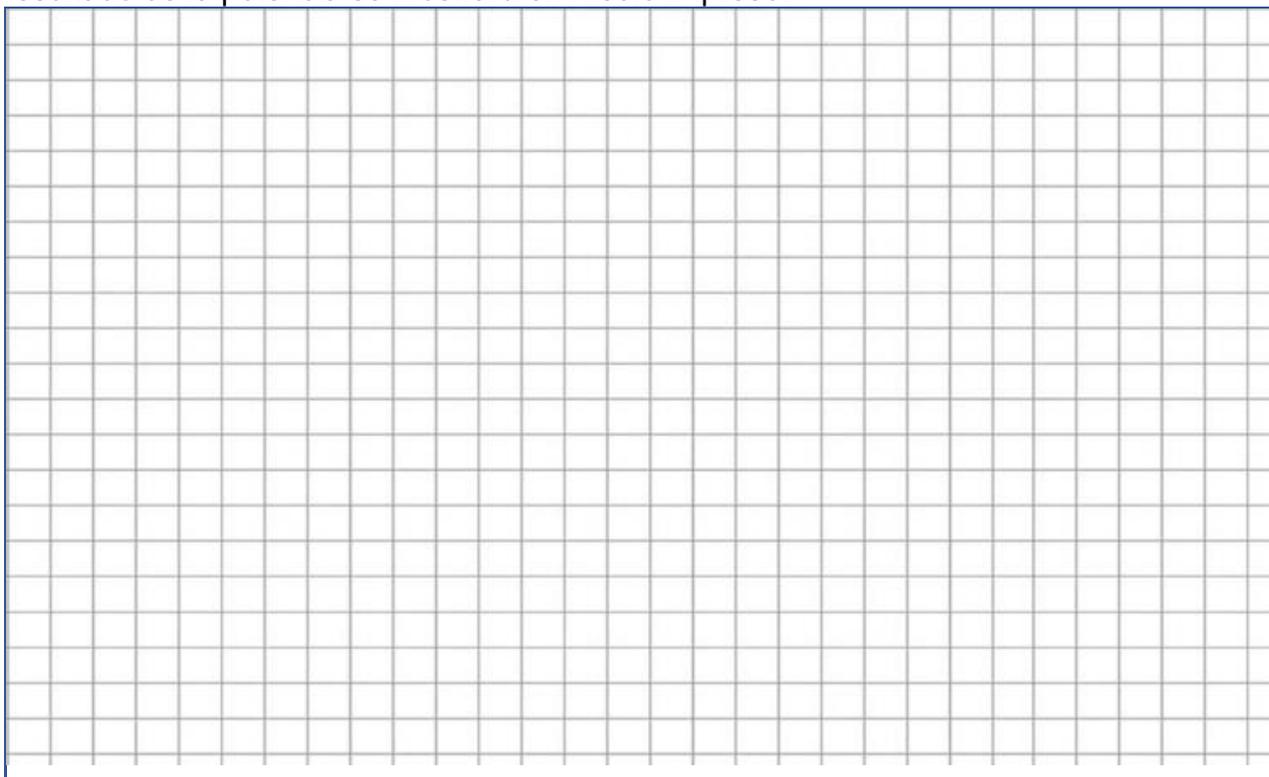
2. Diseñe un diagrama que con el ingreso de dos números obtenga la sumatoria y el promedio de los números siempre y cuando los números sean diferentes. Los resultados se mostrarán en pantalla.



3. Se necesita un diagrama que permita ingresar tres números y si el primer número es mayor o igual al último obtenga el resultado de la suma de todos los números y producto de ellos.



4. Diseñe un diagrama que ingresando cuatro números calcule el promedio de ellos y lo muestre en pantalla, además, si el promedio obtenido es mayor a 10, calcule una potencia donde la base sea el segundo número y el exponente sea el último número ingresado. El resultado de la potencia se mostrará en medio impreso.

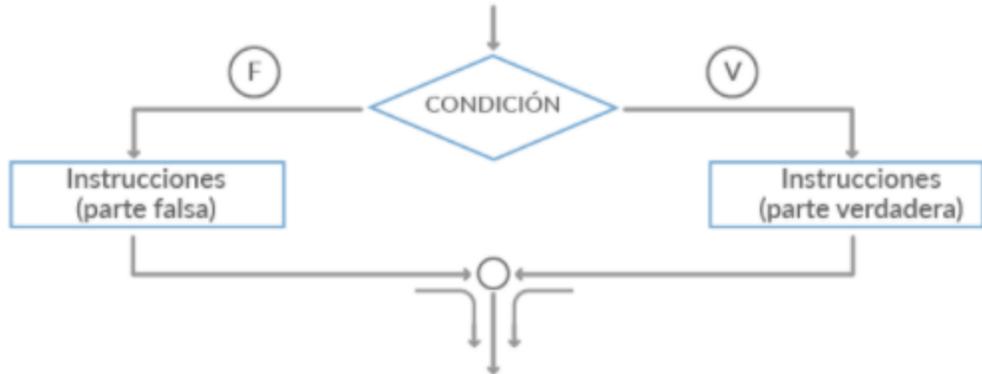


## Guía 9

### Estructura de decisión SI compuesta (if – else)

En este tipo de estructura, se evalúa la condición y si ésta se cumple se ejecuta un determinado grupo de instrucciones. En caso contrario se ejecuta otro grupo de instrucciones. Nunca se ejecutan los dos grupos de instrucciones a la vez, pues si la condición es verdadera ejecuta las instrucciones establecidas por el lado derecho del flujo de los datos, pero si es falsa, se ejecutan las instrucciones establecidas por el lado izquierdo del flujo de los datos.

Igualmente, la condición se evalúa con los operadores relacionales y cuando la estructura condicional se termina, el flujo de los datos continua hacia abajo por una sola línea.



**Ejemplo:** Diseñar un diagrama que a partir de dos números realice las siguientes operaciones:

- Si el primer número es mayor que el segundo, obtenga el residuo que resulta al dividir el número mayor entre el menor, muestre el resultado en medio impreso.
- En caso contrario, obtenga la suma de los dos números y muestre el resultado en pantalla.

Analizando el diagrama podemos observar que:

Los números ingresados se almacenan en las variables **a** y **b**.

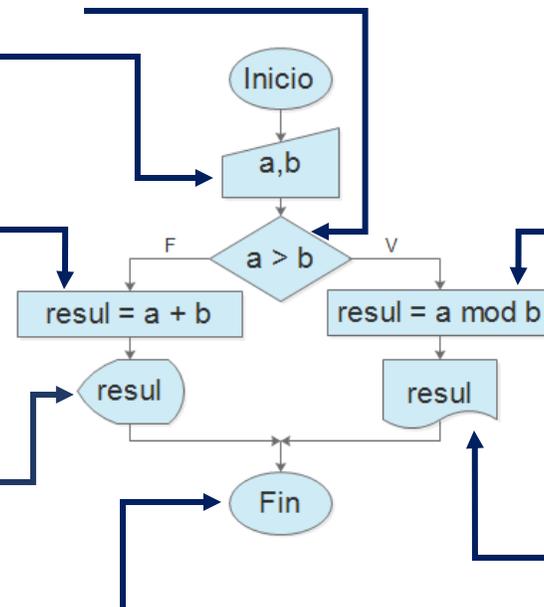
Se hace la comparación entre los dos números para saber si el primero es mayor que el segundo en una condición haciendo uso del operador relacional

**Mayor que (>).**

Si el resultado de la comparación es falso:

Se procede a calcular la suma de los dos números, el resultado se lo almacena en la variable **result**.

Se muestra en pantalla la variable **result** que contiene el resultado de la operación realizada.



Si el resultado de la comparación es verdadero:

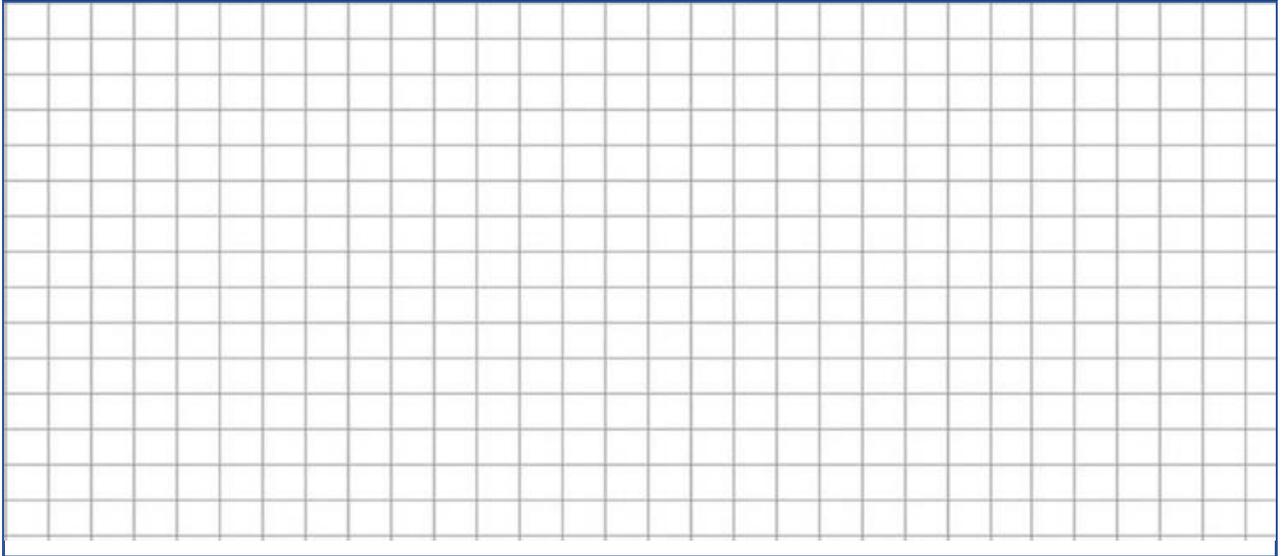
Se procede a calcular el residuo de la división entre los dos números, el resultado se lo almacena en la variable **result**.

Se muestra en medio impreso la variable **result** que contiene el resultado de la operación realizada.

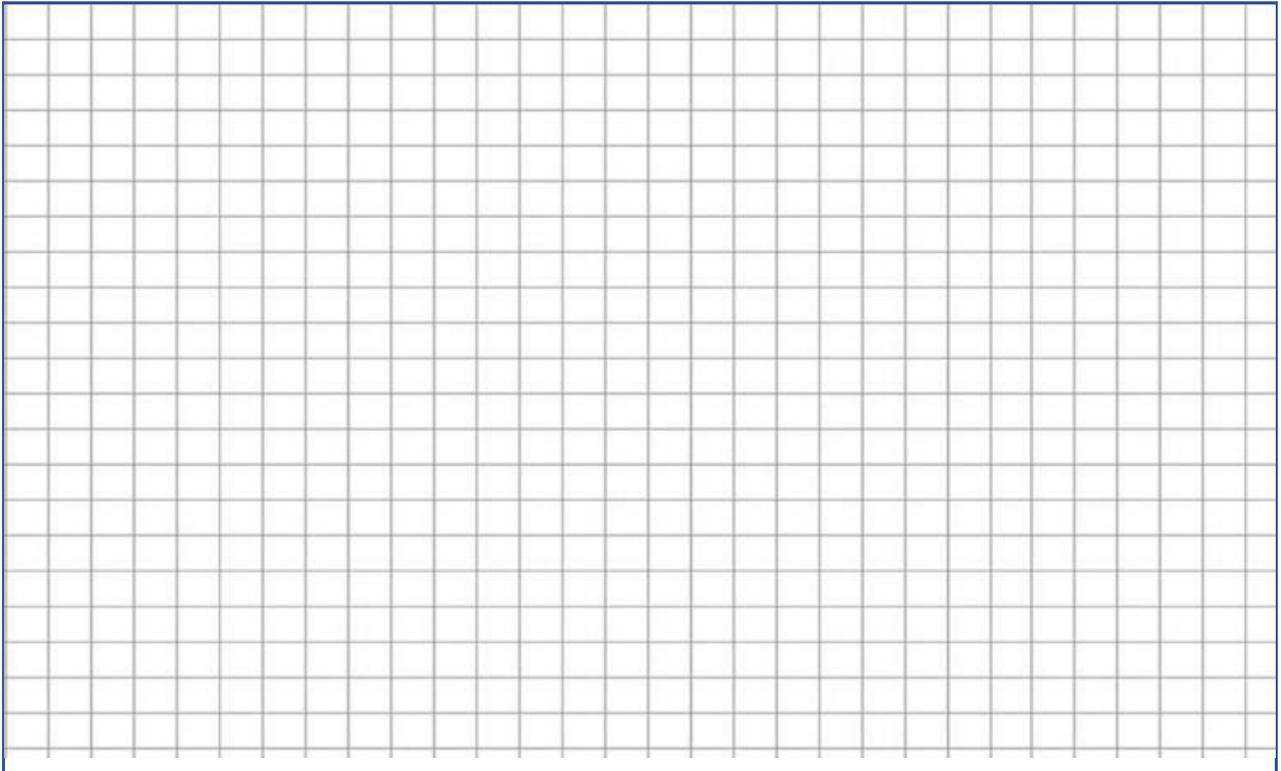
Se finaliza la estructura condicional por una sola línea de flujo.

## Actividad

1. Realice un DFD que permita ingresar dos números y realice lo siguiente:
  - Si el segundo número es menor que el primero obtenga el promedio de los números y muestre el resultado en pantalla.
  - En caso contrario calcule una potencia donde la base sea 4 y la potencia el número mayor. Muestre el resultado en medio impreso.



2. Diseñe un diagrama que con el ingreso de dos números obtenga lo siguiente:
  - Si los números son diferentes calcule el producto de los números y muestre el resultado en pantalla.
  - En caso contrario, obtenga el cuadrado del número. Muestre el resultado en pantalla.

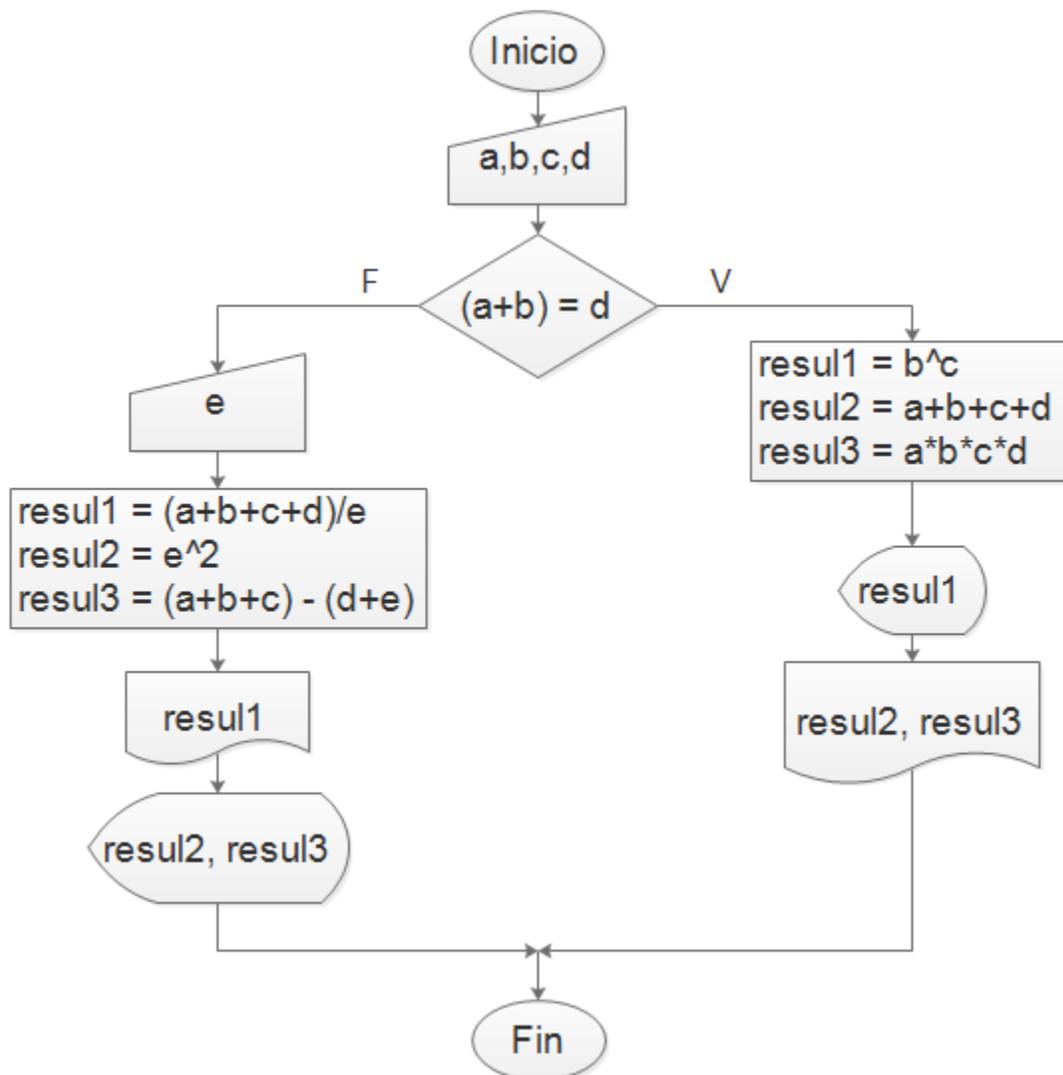


## Guía 10

Tome como ejemplo el siguiente diagrama de flujo de datos para dar solución a los problemas planteados en la guía.

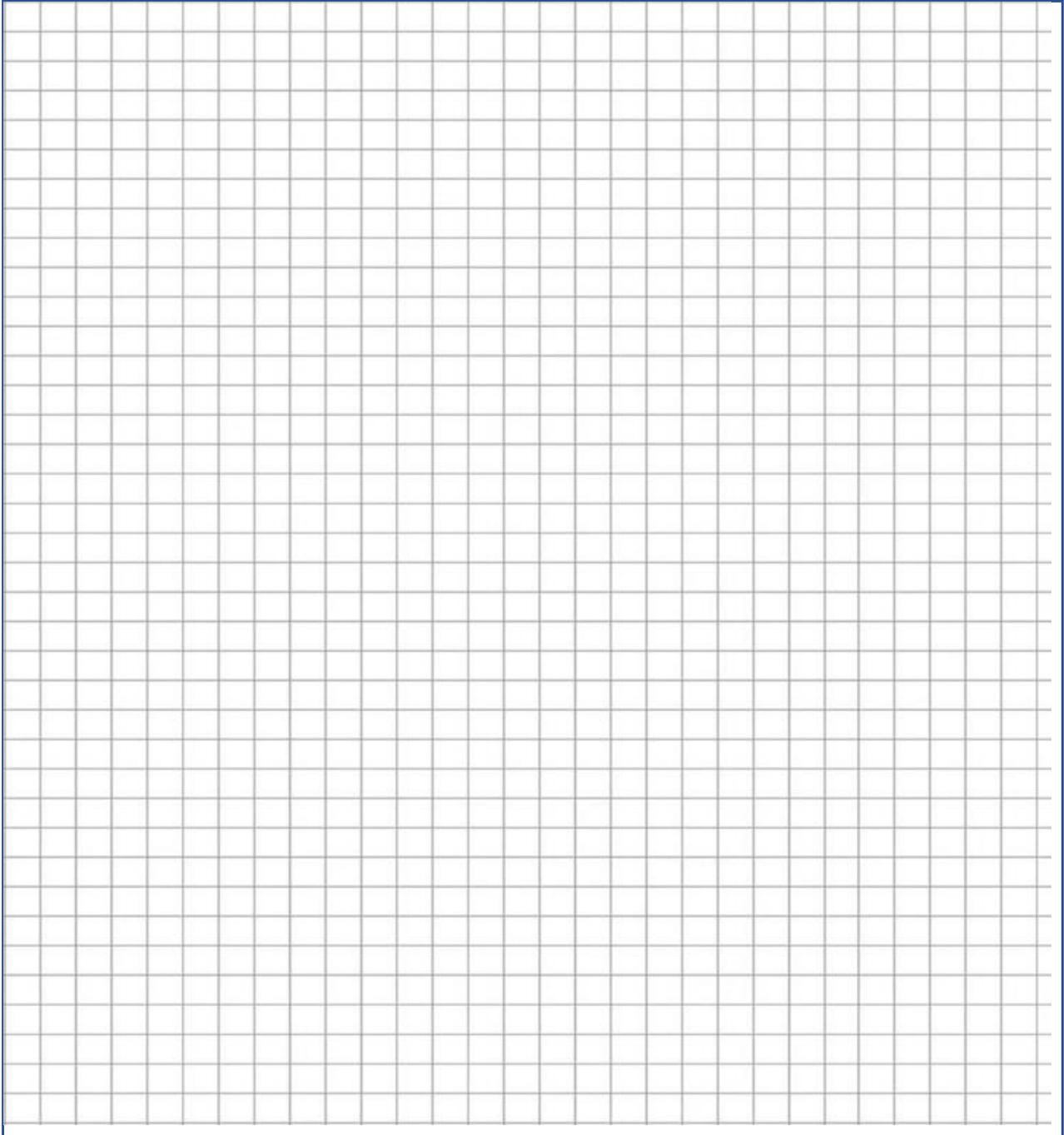
**Problema:** Diseñar un diagrama de flujo de datos que a partir de la entrada de cuatro números realice los siguientes cálculos:

- Si la suma de los dos primeros números es igual al cuarto número, obtenga una potencia donde la base sea el segundo número y el exponente el tercer número, además obtenga la sumatoria y el producto de los números. El resultado de la potencia se mostrará en pantalla y la sumatoria y el promedio en documento impreso.
- En caso contrario, ingrese un número más y obtenga el resultado de la división entre la sumatoria de los primeros cuatro números y el último número ingresado, este resultado muéstrelolo en un documento. También calcule el cuadrado del último número ingresado y la suma de los tres primeros números menos la suma de los dos últimos. Estos últimos resultados se mostrarán en pantalla.



## Actividad

1. Diseñe un diagrama de flujo que permita ingresar tres números y realice lo siguiente:
  - Si el tercer número es mayor que el primero, obtenga el residuo de dividir estos dos números, además calcule el cuadrado de la suma los dos primeros números, obtenga el producto y la sumatoria de todos los números. Estos resultados se mostrarán en documento.
  - En caso contrario, obtenga el residuo de dividir el primer numero entre el tercero, también calcule una potencia donde la base sea el primer número y el exponente sea la suma del segundo con el tercer número. Muestre los resultados en pantalla.



2. Diseñe un diagrama que permita ingresar un número, si el número ingresado es menor que 10 entonces:
- Ingrese dos números más y calcule el promedio de los números, además obtenga una potencia donde la base sea la suma de los dos nuevos números y el exponente sea el primer número ingresado. También obtenga el residuo de dividir 20 entre el primer número ingresado. El promedio y la potencia se mostrarán en documento, el residuo se mostrará en pantalla.
  - Si la condición es falsa, ingrese un número mas y calcule una potencia donde la base sea el primer número y el exponente el segundo número ingresado, también obtenga el residuo de dividir 100 entre la suma de los dos números ingresados y por último obtenga el producto de los dos números. Muestre todos los resultados en pantalla.

