

Apellido y Nombre:

Fecha:/...../.....

CONCEPTOS TEÓRICOS A TENER EN CUENTA

La **Programación Estructurada** es un paradigma o modelo de programación que define cómo debe construirse un programa. Este modelo utiliza, entre otras herramientas, lenguajes algorítmicos para diseñar la lógica de un programa:

- Pseudocódigo** es un lenguaje (escrito) de especificación de algoritmos
- Diagrama de flujo (flowchart)** es una técnica de representación gráfica de algoritmos que utiliza símbolos (cajas) estándar y que tiene los pasos del algoritmo escritos en esas cajas unidas por flechas, denominadas líneas de flujo, que indican la secuencia en que se deben ejecutar.

LENGUAJE ALGORÍTMICO: es todo recurso que permita describir con mayor o menor nivel de detalle los pasos que componen un algoritmo. Son lenguajes algorítmicos, los lenguajes de programación, el pseudocódigo, los diagramas de flujo, los diagramas de Nassi-Shneiderman, etc.

TEOREMA DE LA PROGRAMACIÓN ESTRUCTURADA: El Teorema de la Programación Estructurada establece que un programa propio puede ser escrito utilizando solamente las siguientes estructuras de control:

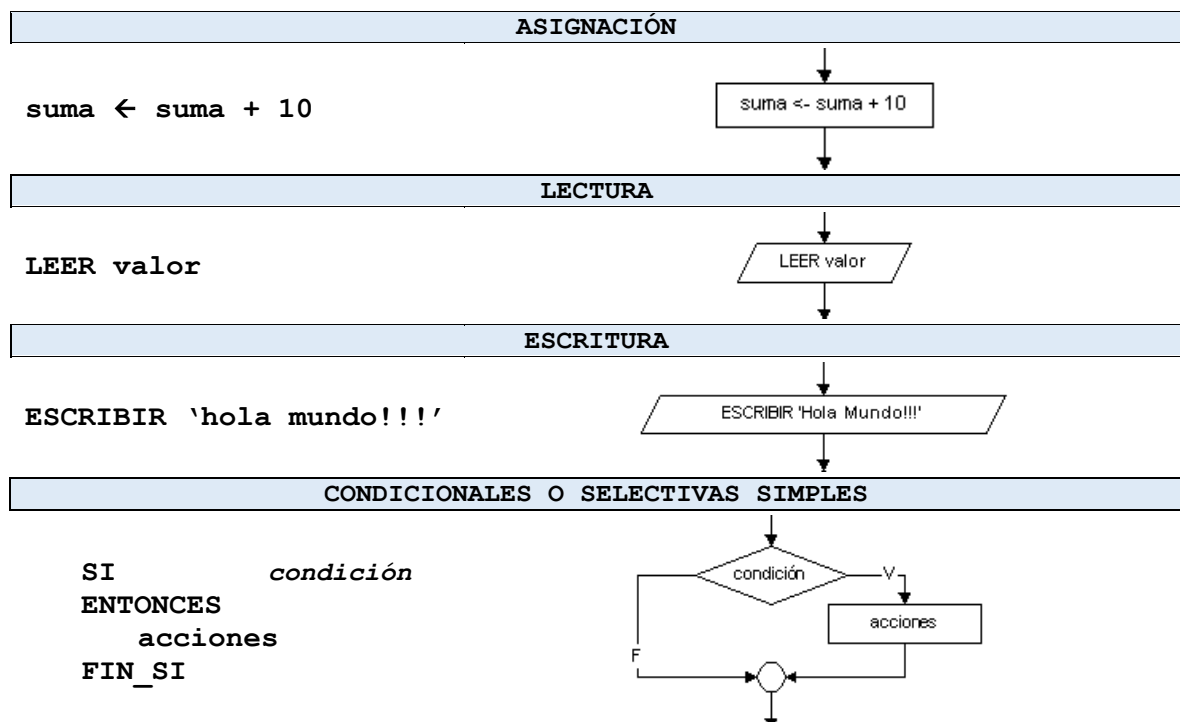
- **Secuenciales**
- **Selectivas**
- **Repetitivas**

Un programa se define como propio si cumple con las siguientes características:

- **tiene exactamente una entrada y una salida para control del programa,**
- **existen caminos que se pueden seguir desde la entrada hasta la salida que conducen por cada parte del**

ESTRUCTURAS DE CONTROL SECUENCIALES Y SELECTIVAS

EQUIVALENCIAS DE LAS OPERACIONES Y ESTRUCTURAS DE CONTROL ENTRE DIAGRAMA DE FLUJO Y PSEUDOCÓDIGO.

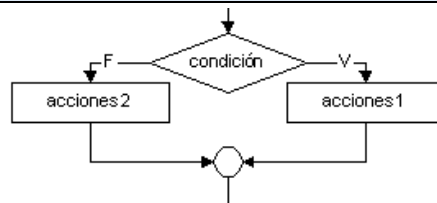


Funciona de la siguiente forma:

1. Se evalúa la expresión lógica *condición*.
2. Si la condición toma el valor Verdadero, se ejecutan las acciones1 y el control pasa a la sentencia inmediatamente siguiente a la SI.... ENTONCES....FINSI, es decir a la siguiente sentencia del programa.
3. Si la condición toma el valor Falso, el control pasa a la sentencia inmediatamente siguiente a la SI.... ENTONCES....FINSI, es decir a la siguiente sentencia del programa.

CONDICIONALES O SELECTIVAS DOBLES

SI condición **ENTONCES**
 acciones1
SINO
 acciones2
FIN_SI

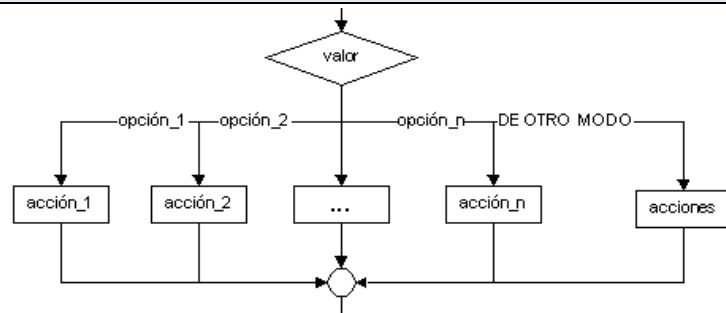


Funciona de la siguiente forma:

4. Se evalúa la expresión lógica *condición*.
5. Si la condición toma el valor Verdadero, se ejecutan las acciones 1 y el control pasa a la sentencia inmediatamente siguiente a la SI.... ENTONCES....FINSI, es decir a la siguiente sentencia del programa.
6. Si la condición toma el valor Falso, se ejecutan las acciones 2 y el control pasa a la sentencia inmediatamente siguiente a la SINO.... ENTONCES....FINSI, es decir a la siguiente sentencia del programa.

CONDICIONALES O SELECTIVAS MÚLTIPLES

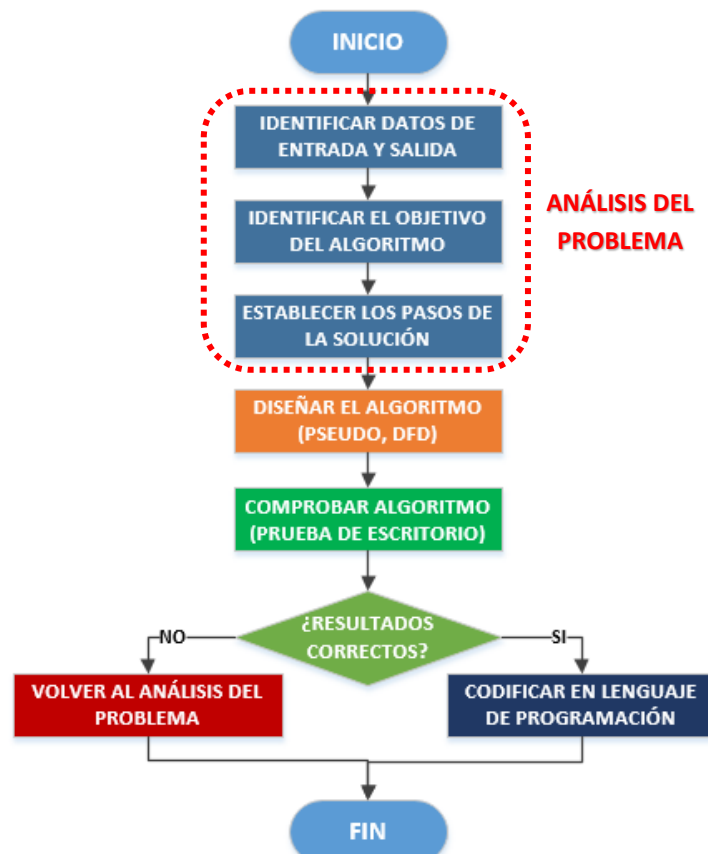
SEGÚN opción **HACER**
 op1: acciones_1
 op2: acciones_2
 ...
 opn: acciones_n
DE OTRO MODO
 acciones
FIN_SEGUN



Funcionamiento

1. La expresión opción se evalúa si es igual a op1 se ejecutarán acciones_1, si es igual a op2 se ejecutarán acciones_2.... si es igual a opn se ejecutarán acciones_n. Se pueden agregar tantos casos como se necesiten.
2. Si el valor de opción no coincide con op1...opn, ejecutará las acciones que se indican en la opción DE OTRO MODO, que representa a todos los otros casos que no fueron indicados explícitamente.

METODOLOGÍA DE TRABAJO



EJERCICIOS RESUELTOS

Ejemplo 1: Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que calcule la superficie de un rectángulo de base b y altura h .

Análisis del problema

El análisis del problema debe permitirnos responder a las siguientes preguntas:



¿Cuál es el problema que debo resolver?

Calcular el área de un rectángulo 👍

¿Qué datos necesito para resolver el problema?

Base y altura del rectángulo 👍

¿Cuál es el resultado que debo obtener?

Área de un rectángulo 👍

¿Cuáles son los pasos que debo realizar para obtener la solución?

- 1) Obtener los datos de base y altura 👍
- 2) Aplicar la fórmula de superficie de rectángulos, utilizando los datos de base y altura 👍
- 3) Mostrar el resultado calculado 👍



El algoritmo tiene por objetivo **calcular el área de un rectángulo**, para ello, será necesario utilizar 3 variables reales: **b (base)**, **h (altura)** y **a (área)**; donde **b** y **h** son variables de **entrada del problema** y **a** es variable de **salida**. Los valores de **b** y **h** se obtienen mediante operaciones de lectura o entrada, luego se calcula el área (que se guarda en la variable **a**) y se muestra el resultado final.

ESTRUCTURA DE UN PROGRAMA

```
{CABECERA DE PROGRAMA}
PROGRAMA nombre_programa
{DECLARACIÓN DE VARIABLES}
VARIABLES
Nombre variable: tipo
{DECLARACIÓN DE PROCEDIMIENTOS Y FUNCIONES}
.....
{PROGRAMA PRINCIPAL}
INICIO
Acción 1
Acción 2
....
Acción n
FIN
```

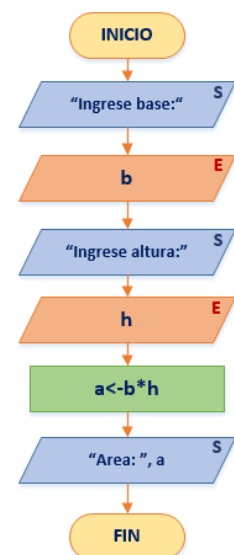
PROGRAMA PRINCIPAL

```
PROGRAMA rectángulo
VARIABLES
  b, h, a: REAL
INICIO
  ESCRIBIR "ingrese base"
  LEER b
  ESCRIBIR "ingrese altura"
  LEER h
   $a \leftarrow b * h$ 
  ESCRIBIR "Area:" a
FIN
```

LEER: operación de entrada que permite capturar valores y asignarlos a variables específicas. Esta operación se conoce como Lectura.

ESCRIBIR: operación de salida que envía a un dispositivo de salida un mensaje o valores de variables. Esta operación se conoce como Escritura.

DIAGRAMA DE FLUJO



Ejemplo 2: Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que calcule la suma de dos números enteros, ingresados por el usuario y luego muestre el resultado por pantalla.

Análisis del problema



¿Cuál es el problema que debo resolver?

Calcular suma de 2 números 👍

¿Qué datos necesito para resolver el problema?

Dos valores numéricos enteros 👍

¿Cuál es el resultado que debo obtener?

Un número entero 👍

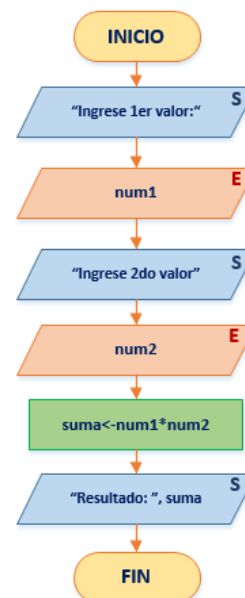
¿Cuáles son los pasos que debo realizar para obtener la solución?

- 1) Obtener los valores a sumar 👍
- 2) Aplicar el operador de suma (+) a los valores de entrada 👍
- 3) Mostrar el resultado calculado 👍



El algoritmo tiene por objetivo **calcular la suma de dos números enteros** ingresados por el usuario, para ello, se usan 3 variables enteras: **num1**, **num2** y **suma**; donde **num1** y **num2** son las variables de **entrada del problema** y **suma** es la variable de **salida**. Los valores de **num1** y **num2** se obtienen mediante operaciones de lectura o entrada. Obtenidos éstos, se aplicará el operador de suma (+) y se almacenará el resultado en la variable **suma**, cuyo contenido se mostrará mediante una operación de escritura o salida.

```
PROGRAMA suma_numeros
VARIABLES
    num1, num2, suma: ENTERO
INICIO
    ESCRIBIR "Ingrese el 1er valor:"
    LEER num1
    ESCRIBIR "Ingrese el 2do valor:"
    LEER num2
    suma ← num1 + num2
    ESCRIBIR "Resultado: ", suma
FIN
```



Ejemplo 3: Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que calcule la longitud de una cadena ingresada por el usuario. Si la cadena estuviese vacía, debe mostrarse notificarse al usuario.

Análisis del problema



¿Cuál es el problema que debo resolver?

¿Qué datos necesito para resolver el problema?

¿Cuál es el resultado que debo obtener?

¿Cuáles son los pasos que debo realizar para obtener la solución?

Contar los caracteres de una cadena 👍

Una cadena de caracteres 👍

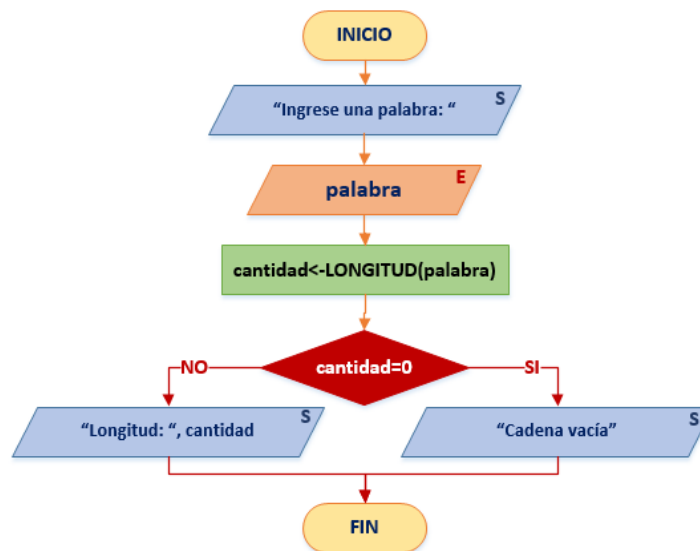
Cantidad de caracteres de la cadena o un mensaje en caso de una cadena vacía. 👍

- 1) Obtener la cadena de caracteres 👍
- 2) Aplicar la operación que cuenta los caracteres de una cadena 👍
- 3) Mostrar la longitud de la cadena o el mensaje de cadena vacía. 👍



El algoritmo tiene por objetivo **contar los caracteres de una cadena** ingresada por el usuario, usando para ello 2 variables: **palabra** de tipo cadena y **cantidad** de tipo entero. La variable **palabra** (**dato de entrada**) se cargará mediante una operación de lectura (entrada por teclado), mientras que la variable **cantidad** (**resultado**) almacenará la cantidad de caracteres de la cadena aplicando la operación LONGITUD. Si la cadena estuviese vacía, se deberá presentar un mensaje que notifique al usuario de tal situación.

```
PROGRAMA ejemplo3
VARIABLES
    palabra: CADENA
    cantidad: ENTERO
INICIO
    ESCRIBIR "Ingrese una palabra"
    LEER palabra
    cantidad ← LONGITUD(palabra)
    SI cantidad=0 ENTONCES
        ESCRIBIR "CADENA VACIA"
    SINO
        ESCRIBIR "Longitud: ", cantidad
    FIN_SI
FIN
```



El algoritmo tiene por objetivo **determinar la cantidad de caracteres de una cadena** ingresada por el usuario. Para ello, se utilizarán 2 variables: **palabra** y **cantidad**. La variable **palabra** (**dato de entrada**) almacenará la cadena (secuencia de caracteres) introducida por teclado (operación de lectura). La variable **cantidad** (**dato de salida**) se empleará para guardar la cuenta de caracteres que se obtendrá mediante la operación **LONGITUD**. Esta función permite contar los caracteres de una cadena especificada. El resultado de esta función se almacenará en la variable **cantidad** mediante el operador de **ASIGNACIÓN**. A partir del resultado obtenido se podrá mostrar la longitud de la cadena **palabra** o bien el **mensaje "Cadena Vacía"**. Para evaluar el resultado obtenido y elegir la acción a realizar, se utilizará una **estructura selectiva** (SI/ENTONCES/SINO/FIN_SI). Esta estructura evalúa una expresión lógica (**cantidad=0**) y determina si se ejecutarán las acciones indicadas por el "camino" VERDADERO o por el "camino" FALSO.

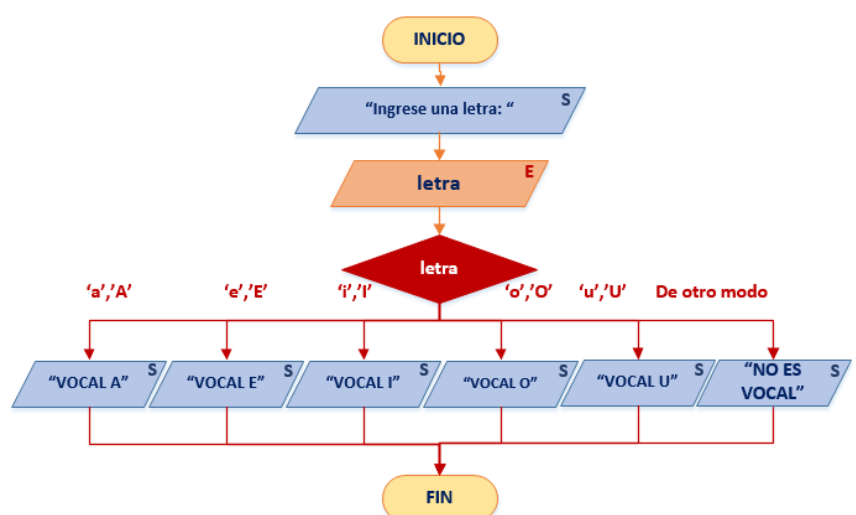
Ejemplo 4: Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que muestre el nombre de la vocal ingresada por el usuario.

Para ello, considere lo siguiente:

- Si la vocal es 'a' mostrar VOCAL A
- Si la vocal es 'e' mostrar VOCAL E
- Si la vocal es 'i' mostrar VOCAL I
- Si la vocal es 'o' mostrar VOCAL O
- Si la vocal es 'u' mostrar VOCAL U
- Para cualquier otra letra o símbolo, mostrar NO ES VOCAL

```

PROGRAMA letras
VARIABLES
letra: CARACTER
INICIO
  ESCRIBIR "Ingrese una letra"
  LEER letra
  SEGÚN letra HACER
    'a','A': ESCRIBIR "VOCAL A"
    'e','E': ESCRIBIR "VOCAL E"
    'i','I': ESCRIBIR "VOCAL I"
    'o','O': ESCRIBIR "VOCAL O"
    'u','U': ESCRIBIR "VOCAL U"
  DE OTRO MODO
    ESCRIBIR "NO ES VOCAL"
  FIN_SEGÚN
FIN
  
```

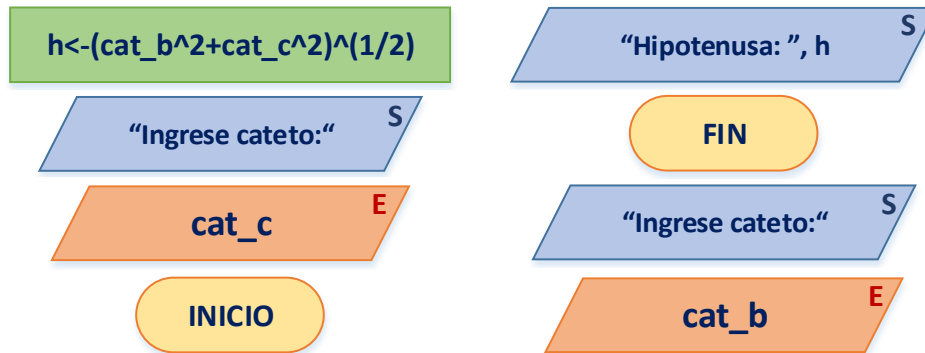


El algoritmo tiene por objetivo **mostrar un mensaje** indicando cuál fue la vocal ingresada por el usuario. Para ello se utilizará una variable de tipo carácter, denominada **letra** (dato de entrada), que almacenará el carácter introducido. En función de ese dato se mostrará el cartel "**VOCAL NOMBRE DE LA VOCAL**", salvo cuando el carácter ingresado no sea una vocal en cuyo caso se presentará el mensaje "**NO ES VOCAL**". Para elegir cuál será el mensaje que se presentará es posible utilizar una estructura selectiva múltiple.

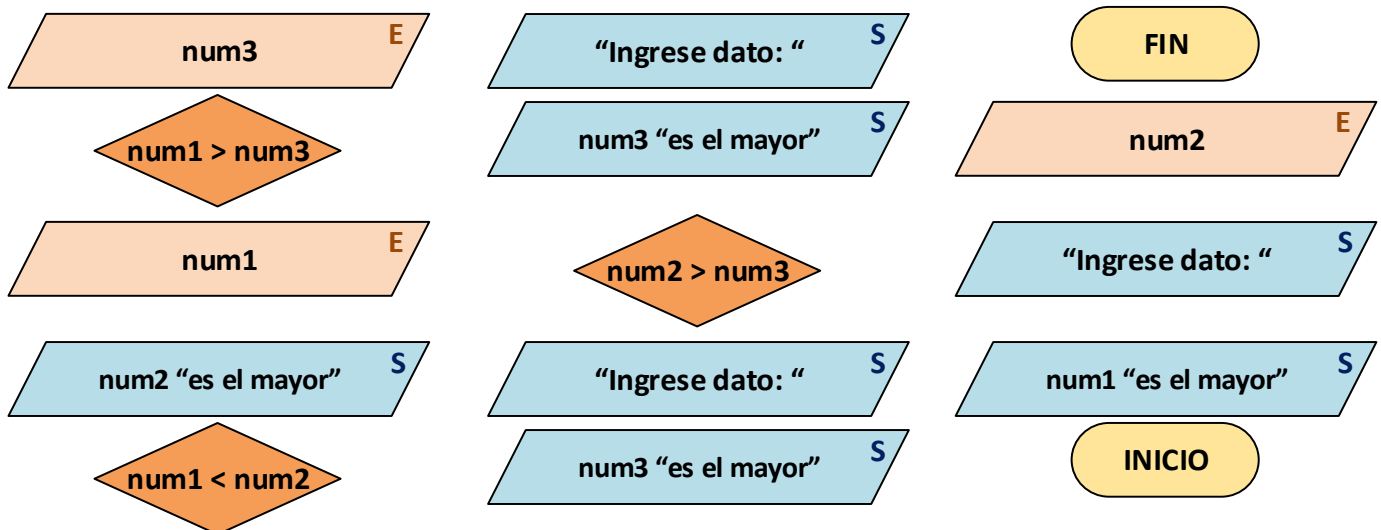
Esta estructura permite seleccionar uno de varios caminos en función de un valor simple (entero, carácter). Por ejemplo, si la variable *letra* contiene el dato 'e' entonces se mostrará el mensaje "VOCAL E", mientras que si letra vale '@' se presentará el cartel "NO ES VOCAL".

EJERCICIOS A RESOLVER

1. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que muestre el mensaje "Hola Mundo!!!, mi primer algoritmo".
2. Un programador diseñó un algoritmo para calcular la hipotenusa de un triángulo rectángulo, sin embargo, olvidó conectar los símbolos del diagrama de flujo que definen el orden de los pasos del algoritmo. ¿Podrías ordenar y conectar los símbolos de la solución propuesta por este programador?



3. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que calcule la superficie y el volumen de un cilindro de radio *r* y altura *h* (ingresadas por el usuario).
4. El jefe de programadores solicita agregar controles al algoritmo del ítem 2. Estos controles deben verificar que los datos ingresados sean válidos para el problema, indicando mediante mensajes las situaciones de error. ¿Podrías agregarlos?
5. Nuevamente el programador del ítem 1 olvidó conectar los símbolos del diagrama de flujo. Sabiendo que el objetivo del algoritmo es identificar el mayor de 3 valores, ¿podrías ordenar y conectar adecuadamente los símbolos del diagrama?



6. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que permita ingresar 2 cadenas de caracteres, compararlas y mostrar la longitud de la mayor.
7. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que permita ingresar 2 cadenas de caracteres, mostrar la longitud de cada una e indicar cuál tiene la mayor longitud. Si alguna de las cadenas es nula se presentará el mensaje "Ingreso incorrecto".
8. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que permita ingresar 2 valores (dividendo y divisor) y calcular el cociente y resto de éstos. Tenga en cuenta que la división por cero no está definida.
9. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que permita ingresar 3 valores e identificar cuál es el mayor y cuál el menor. En caso de valores iguales debe presentarse el mensaje "IGUALES".

10. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que determine si 3 datos ingresados por el usuario, correspondientes a horas, minutos y segundos, son válidos o no. Tenga en cuenta que las horas se encuentran en el rango de 0 a 23, mientras que los minutos y segundos pertenecen al intervalo 0 a 59.
11. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que determine la condición final de un alumno tras cursar una materia. Para ello, considere que la materia tiene 2 evaluaciones, cada una con su respectivo recuperatorio, y que éstas deben aprobarse con nota mayor o igual a 7. En el caso de los alumnos aprobados deberá mostrarse el promedio, mientras que para los desaprobados se presentará el mensaje "Alumno Libre".
12. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que muestre el nombre de un tipo de video juego, de acuerdo a una letra ingresada por el usuario. Considere la siguiente correspondencia: ('a','A') Acción, ('s','S') Shooter, ('e','E') Estrategia y ('d','D') Deportes. En caso de un ingreso no contemplado, muestre el mensaje "ERROR EN LA ENTRADA".
13. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que muestre una opción seleccionada en el menú de un cajero automático, de acuerdo a una letra ingresada por el usuario. Considere la siguiente correspondencia: ('d','D') Depósitos, ('e','E') Extracciones, ('c','C') Consultas y ('s','S') Salir. En caso de un ingreso no contemplado, muestre el mensaje "ERROR EN LA ENTRADA".
14. Dado el siguiente algoritmo:

```
PROGRAMA misterio
VARIABLES
  z, x: ENTERO
  band: LOGICO
INICIO
  ESCRIBIR "Ingrese un numero: "
  LEER x
  ESCRIBIR "Ingrese un numero: "
  LEER z
  band ← (x MOD z = 0)
  SI (band = VERDADERO) ENTONCES
    ESCRIBIR "El numero", x, "es ____ de", z
  FIN_SI
FIN
```

- a) Realice la prueba de escritorio para los valores:
x=6 y z=2; x=9 y z=4 .
- b) Determine el objetivo del algoritmo.
- c) ¿Con qué palabra completa el mensaje mostrado por pantalla?
- d) Realice el diagrama de flujo equivalente.

15. Dado el siguiente algoritmo:

```
PROGRAMA incognita
VARIABLES
  a, b: ENTERO
INICIO
  ESCRIBIR "Ingrese 1er numero: "
  LEER a
  ESCRIBIR "Ingrese 2do numero: "
  LEER b
  SI (a > b) ENTONCES
    a ← a-b
    b ← a+b
    a ← b-a
  SINO
    b ← b-a
    a ← a+b
    b ← a-b
  FIN_SI
  ESCRIBIR "Primer numero: ", a
  ESCRIBIR "Segundo numero: ", b
FIN
```

- a) Realice la prueba de escritorio para los valores:
a=3, b=4 y a=7, b=2.
- b) Determine el objetivo del algoritmo.
- c) Realice el diagrama de flujo equivalente.

