PROGRAMACIÓN ESTRUCTURADA

Trabajo Práctico N° 3 — 03

Estructuras Secuenciales y Selectivas

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:

a) ***Pseudocódigo*** es un lenguaje (escrito) de especificación de algoritmos

b) ***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:

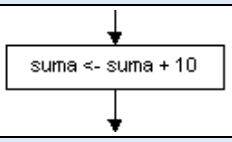
- ***tiene exactamente una entrada y una salida para control del programa,***
- ***existen caminos que se pueden seguir desde la entrada hasta la salida que conducen por cada parte del***

ESTRUCTURAS DE CONTROL SECUENCIALES Y SELECTIVAS

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

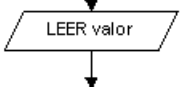
ASIGNACIÓN

```
suma ← suma + 10
```



LECTURA

```
LEER valor
```

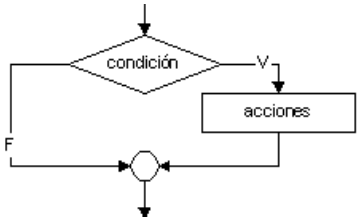


ESCRITURA

```
ESCRIBIR "hola mundo!!!"
```

CONDICIONALES O SELECTIVAS SIMPLES

```
SI condición ENTONCES
   acciones
FIN_SI
```

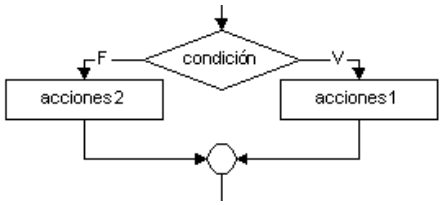


Funciona de la siguiente forma:

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
```

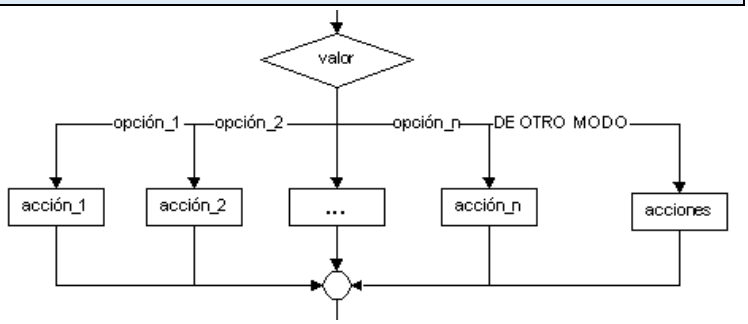


Funciona de la siguiente forma:

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

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.

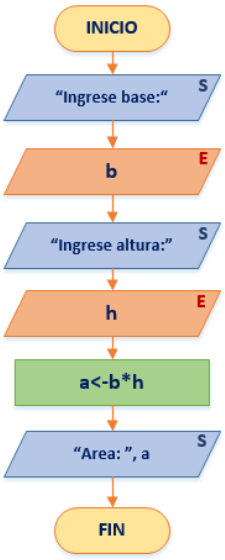
ESTRUCTURA DE UN PROGRAMA

{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←b*h
  ESCRIBIR "Area:" a
FIN
```

DIAGRAMA DE FLUJO



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*.

***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*

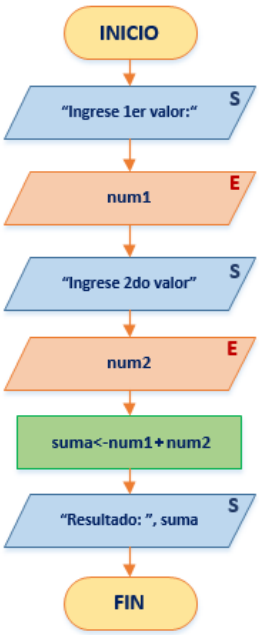
¿Qué datos necesito para resolver el problema? — *Dos valores numéricos enteros*

¿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?

Contar los caracteres de una cadena

¿Qué datos necesito para resolver el problema?

Una cadena de caracteres

¿Cuál es el resultado que debo obtener?

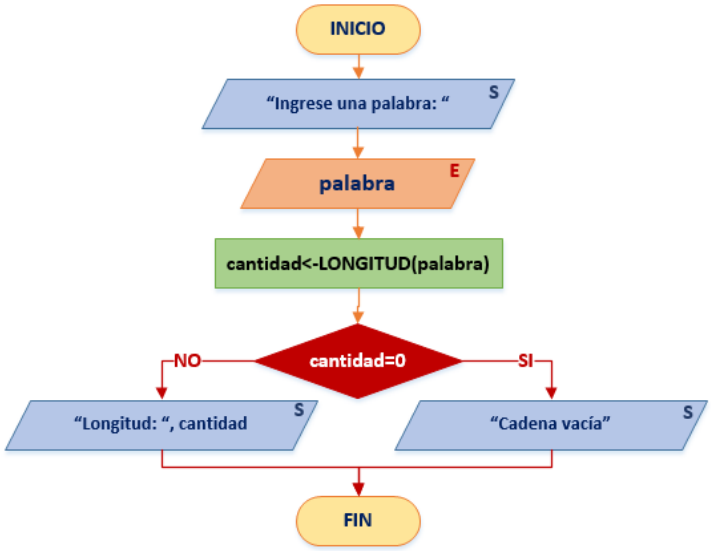
Cantidad de caracteres de la cadena o un mensaje en caso de una cadena vacía.

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

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
```

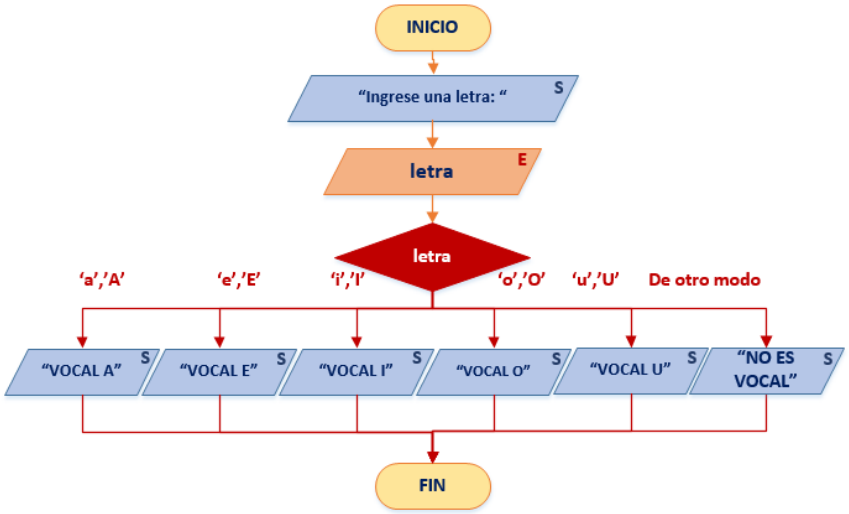


El algoritmo tiene por objetivo **determinar la cantidad de caracteres de una cadena** ingresada por el usuario. Para ello, se utilizarán 2 variables: ***palabra*** y ***cantidad***. La variable *palabra* (**dato de entrada**) almacenará la cadena (secuencia de caracteres) introducida por teclado (operación 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
```



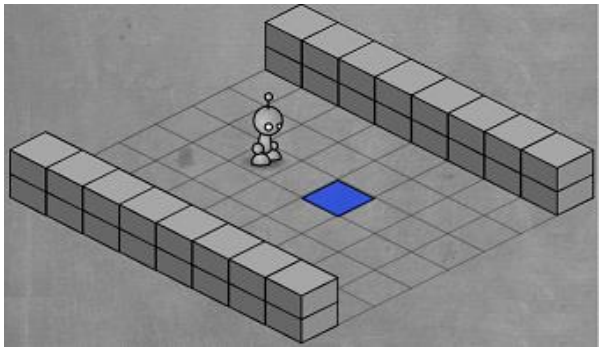
El algoritmo tiene por objetivo **mostrar un mensaje** indicando cuál fue la vocal ingresada por el usuario. Para ello se utilizará una variable de tipo carácter, denominada ***letra*** (dato de entrada), que almacenará el carácter introducido. En función de ese dato se mostrará el cartel "VOCAL *NOMBRE 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. Utilizando el juego online *LightBot* (https://www.cokitos.com/aprender-a-programar-un-robot/play/) resuelva los niveles 3, 4, 5 y 6. Para ello, tome como referencia las soluciones propuestas para los niveles 1 y 2. Consigne los pasos aplicados.

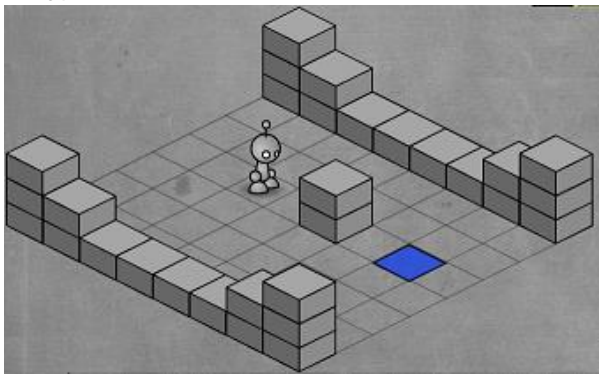
Nivel 1



Acciones

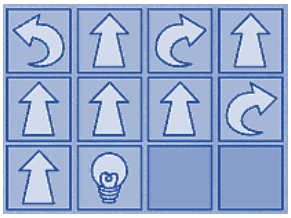
Pasos de solución

Nivel 2

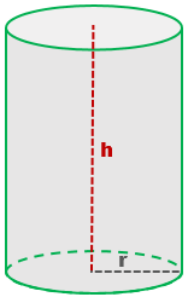


Acciones

Pasos de solución



2. Un programador diseñó un algoritmo para calcular la superficie y el volumen de un cilindro recto usando las siguientes fórmulas: 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? ¿Existe una única solución válida?



h: altura

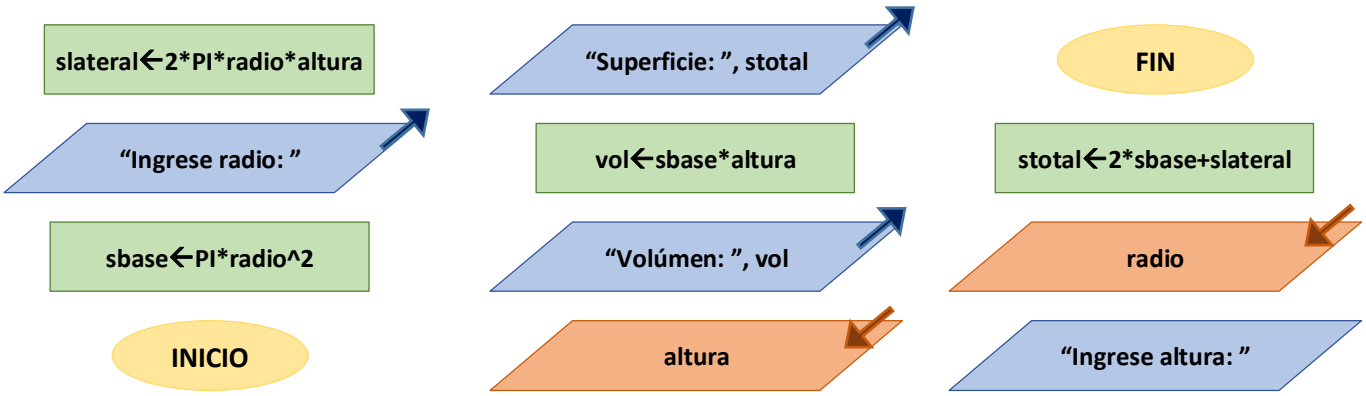
r: radio

$$Sup.base = \Pi \times radio^2$$

$$Sup.lateral = 2 \times \Pi \times radio \times altura$$

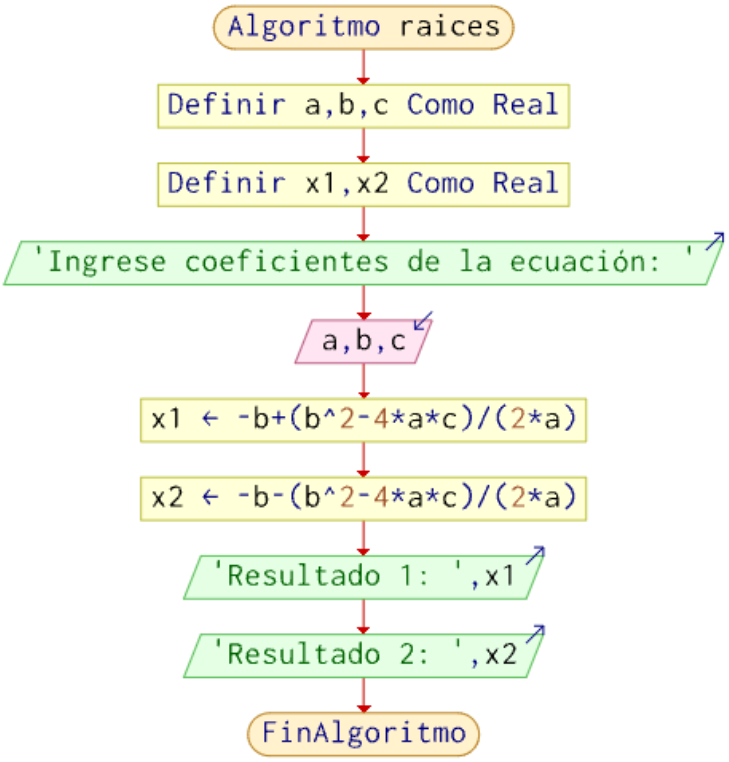
$$Superficie\ cilindro = 2 \times sup.base + sup.lateral$$

$$Volúmen\ cilindro = sup.base \times altura$$



3. Tomando como referencia el ejercicio 13 del TP2, diseñe un algoritmo que resuelva las operaciones planteadas. Incluya los controles para verificar que los datos ingresados sean válidos para el problema, y muestre mensajes para notificar al usuario de las situaciones de error.

4. Uno de tus compañeros del curso de programación diseño un algoritmo para calcular las raíces de una ecuación cuadrática. Al ser su primer algoritmo, este joven programador no está seguro de haberlo resuelto correctamente. Podrías revisar su diseño, corregir los errores que se le hayan escapado y agregar los controles necesarios para realizar las operaciones sin error. Como ayuda, la profe de la práctica te indicó los siguientes valores de prueba: a=4, b=7, c=3 cuyos resultados son x1= -0,75; x2=-1 y a=8, b=17, c=9 cuyos resultados son x1=-1, x2=-1,125



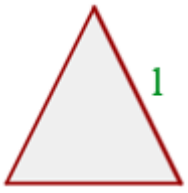
5. Un triángulo es un polígono de 3 lados que, en función del valor de sus lados, puede clasificarse en equilátero, isósceles o escaleno. En particular si el triángulo tiene un ángulo recto se denomina triángulo rectángulo y verifica el conocido teorema de Pitágoras. Teniendo en cuenta las propiedades que se comentan a continuación diseñe los algoritmos (diagrama de flujo y pseudocódigo) correspondientes.

   a) Cálculo del área de un triángulo conociendo su base y altura: Se aplica la muy conocida fórmula

$$area_{triángulo} = \frac{base \times altura}{2}$$

   b) Cálculo del área de un triángulo conociendo la longitud de sus lados (fórmula de Herón): Para calcular el área de un triángulo cuando sólo se conoce la longitud de sus lados puede aplicarse la fórmula de Herón.

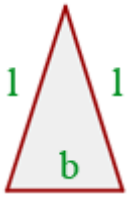
Triángulo Equilátero



Perímetro

$$p = 3 \times lado$$

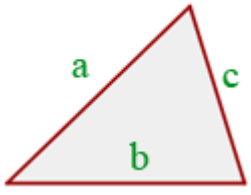
Triángulo Isósceles



Perímetro

$$p = 2 \times lado + b$$

Triángulo Escaleno



Perímetro

$$p = a + b + c$$

$$\text{Semiperímetro } sp = \frac{p}{2}$$

$$\text{Área del triángulo } a = \sqrt{sp \times (sp - lado_1) \times (sp - lado_2) \times (sp - lado_3)}$$
(fórmula de Herón)

c) Determinación de la existencia de un triángulo: Para que un triángulo exista (el valor de sus lados sea válido) debe verificarse que la suma de cualquier par de sus lados sea mayor que el lado restante.

d) Identificación de triángulos equilátero, isósceles y escaleno: De acuerdo al valor de sus lados puede determinarse si un triángulo corresponde a un equilátero, un isósceles o un escaleno.

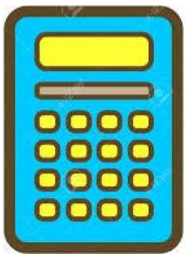
e) Identificación de triángulos rectángulos: Tomando como referencia el teorema de Pitágoras, y el valor de los lados de un triángulo, puede deducirse si un triángulo es rectángulo o no.

f) Calculo de la hipotenusa de un triángulo rectángulo: Aplicando el teorema de Pitágoras puede calcularse el valor de la hipotenusa de un triángulo rectángulo.

6. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que, dado un valor de 4 cifras ingresado por el usuario, determine si éste tiene todos sus dígitos distintos, es un valor capicúa, se compone sólo de dígitos impares o sólo tiene cifras pares. Tenga en cuenta que deben incluirse los controles para el correcto funcionamiento del algoritmo.
7. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que permita ingresar un nombre de usuario, un servidor de correo (gmail, outlook, yahoo, etc.) y un dominio (ar, br, cl, etc.), y con éstos generar una dirección de correo electrónico. Tenga en cuenta que si alguno de los datos es nulo se omitirá la salida, presentándose un mensaje de advertencia al usuario.

```
NOMBRE DE USUARIO: diego90
SERVIDOR: yahoo
DOMINIO: br
DIRECCIÓN: diego90@yahoo.com.br
```

8. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que permita ingresar una cadena de caracteres y determinar si ésta está compuesta únicamente por letras mayúsculas. Considere que el algoritmo analizará cadenas de hasta 5 caracteres, presentando un mensaje de advertencia al usuario para aquellas que sean nulas o mayores 5 caracteres.
9. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que permita ingresar un nombre de usuario y contraseña (4 caracteres) y validar que el nombre de usuario no inicie con un dígito y que la contraseña contenga al menos una letra mayúscula, una letra minúscula y un dígito. Si el nombre de usuario o la contraseña no son válidos deben presentarse los mensajes de error correspondientes.
10. Diseñe un algoritmo que simule el funcionamiento de una calculadora sencilla. Esta calculadora debería ser capaz de leer 2 valores de entrada y resolver la suma (+), la resta (-), el producto (*), la división real (/), la división entera (d), el resto (m), la potencia (^) o la raíz (~). Para ello, el usuario proporcionará los valores a operar y el símbolo que indique la operación a realizar. La solución debe incluir los controles necesarios para realizar correctamente las operaciones.



11. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que presente el siguiente menú de un cajero automático:

```
**** MENU ATM ****
E) Extracción
D) Depósito
T) Transferencia
C) Consulta
S) Salir
******************
Seleccione una opción:
```

Considere que el algoritmo, antes de mostrar el menú, debe solicitar al usuario los siguientes datos: nombre, apellido, tipo de cuenta (caja de ahorro, cuenta corriente) y saldo actual (un valor positivo). Al ejecutar las operaciones debe verificarse que la operación sea posible (saldo suficiente) y que se verifiquen las siguientes restricciones:

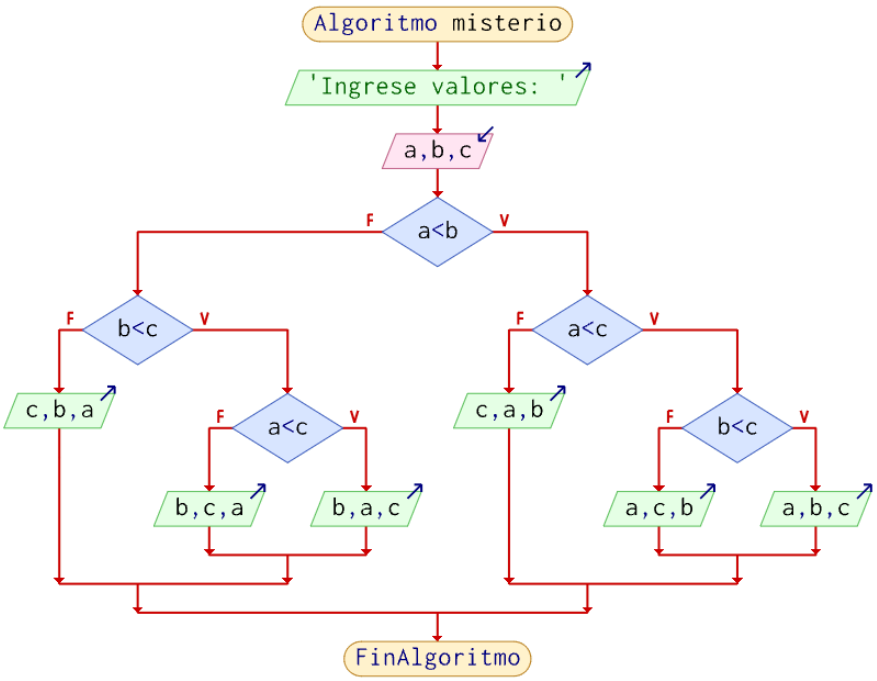
a) las extracciones tiene un límite diario: $70.000
b) los depósitos no pueden exceder los $100.000 debiendo notificarse al cliente que debe dirigirse a sector de cajas del banco para realizar la operación
c) las transferencias no pueden superar los $200.000
d) la operación de consulta debe mostrar el titular de la cuenta (apellido y nombre), tipo de cuenta, su saldo actual y el monto de préstamo disponible (se calcula como el monto del saldo actual + 50% de éste).

Además, ante opciones incorrectas debe presentarse un mensaje de error.

12. Dado el siguiente algoritmo:

```
PROGRAMA enigma
VARIABLES
      num, aux1, aux2, aux3, aux4: ENTERO
INICIO
   ESCRIBIR "Ingrese un valor: "
   LEER num
   SI (num-99>0) Y (num-1000<0) ENTONCES
      aux1←num MOD 10
      aux2←num MOD 100-aux1
      aux3←num DIV 100
      aux4←aux1*100+aux2+aux3
      SI aux4=num ENTONCES
            ESCRIBIR "?????"
      SINO
            ESCRIBIR "?????"
      FINSI
   SINO
      ESCRIBIR "No válido"
   FINSI
FIN
```

a) Realice la prueba de escritorio para *num=173* y *num=212*.
b) Analice el algoritmo, determine su propósito y complete los mensajes en pantalla.
c) Realice el diagrama de flujo equivalente.
d) Redacte el enunciado del problema.



a) Realice la prueba de escritorio para los valores *a=3*, *b=7*, *c=1* y *a=9*, *b=2*, *c=9*
b) Determine el objetivo del algoritmo.
c) Escriba el pseudocódigo equivalente.
d) Redacte el enunciado del problema.

Realizar los siguientes pasos:

1) **Solicitar 2 valores numéricos al usuario, guardándolos en 2 variables.**
2) **Verificar si la primera variable es mayor que la segunda y, en caso afirmativo, ejecutar el paso 3. En caso contrario, ejecutar el paso 4.**
3) **Restar a la primera variable la segunda, almacenando el resultado en la primera variable. Luego, sumar a la segunda variable la primera, guardando el resultado en la segunda. Por último, restar a la segunda variable la primera, almacenando el resultado en ésta última. Ejecutar el paso 5.**
4) **Restar a la segunda variable la primera, almacenando el resultado en la segunda variable. Luego, sumar a la primera variable la segunda, guardando el resultado en la primera. Por último, restar a la primera variable la segunda, almacenando el resultado en ésta última. Ejecutar el paso 5.**
5) **Mostrar el contenido de las variables.**

a) Dibuje el diagrama de flujo y escriba el pseudocódigo equivalente.
b) Realice la prueba de escritorio para los valores *num1=12, num2=25 y num1=37, num2=24.*
c) Determine el objetivo del algoritmo.
d) Redacte el enunciado del problema.