|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Claudia Rodriguez Espino |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 1102 |
| *No de Práctica(s):* | 4 |
| *Integrante(s):* | García León César André |
| *Semestre:* | Primero |
| *Fecha de entrega:* | Viernes 8 Septiembre 2017 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

Guía práctica de estudio 04: Diagramas de flujo

Objetivo:

Elaborar diagramas de flujo que representen soluciones algorítmicas vistas como una serie de acciones que comprendan un proceso.

Desarrollo:

En esta práctica estudiamos lo que son los diagramas de flujo los cuales son la representación gráfica de un algoritmo y son muy importantes ya que son la base para comenzar con la codificación de un programa. Se vieron los símbolos que se utilizan en los diagramas de flujo y sus usos correspondientes, que anteriormente fueron vistos en clase. Los símbolos que forman parte de un diagrama de flujo son:

Representa el inicio y el fin del algoritmo.

Son las líneas que indican la dirección del flujo del diagrama, conocidas como

líneas de flujo, las cuales deben ser rectas, verticales u horizontales.

Proceso: Es en donde se realizan las asignaciones u operaciones.

Se anotan los datos de entrada del algoritmo.

Es el símbolo de decisión el cual valida una condición y toma uno u otro

camino.

Imprime los resultados

Sirve como conexión dentro de la misma página.

Sirve como conexión entre diferentes páginas.

Es el módulo de un problema, realiza una llamada a otros módulos o

Funciones.

Símbolo de decisión múltiple. Se ocupa en problemas que tienen

relación con un menú.

Después de haber visto algunos de los símbolos que se utilizan se analizó la estructura que debe llevar un diagrama de flujo para que se realice de una buena manera.

*1.- El diagrama se realiza de arriba hacia abajo y de izquierda a derecha*

*2.- Las líneas de flujo deben estar conectadas a un símbolo, únicamente una línea a cada símbolo.*

*3.- Si el diagrama excede una página se debe utilizar y numerar los símbolos adecuados.*

*4.- Utilizar la notación de camello para nombrar a las variables o funciones. En esta notación la variable comienza con una letra mayúscula seguida de minúsculas. Ej: Área*

Por último, para terminar con la práctica se explicaron los tres tipos de estructuras de control, cada una de ellas con su respectivo ejemplo y condiciones que maneja, ya que posteriormente éstas condiciones se aplicarán en el desarrollo de la codificación de un programa.

**Estructura de control secuencial:** Las sentencias se realizan una seguida de otra en el orden en que están escritas.

**Estructuras de control condicionales:** Son las que evalúan una condición la cual puede ser verdadera o falsa y dependiendo de lo que se obtenga se realizan otra serie de instrucciones.

En estas estructuras se encuentra la condición conocida como ***if (si)*** que evalúa la condición, si es verdadera realiza instrucciones de lo contrario se sigue con el flujo del diagrama.

***(If- Else) (Si- De lo contrario):*** Evalúa la expresión, si es verdadera realiza una serie de instrucciones, si es falsa realiza otra serie de instrucciones.

***(If- Else) anidado:*** Evalúa más de dos valores

***(Switch case) Seleccionar caso:*** Valida el valor de la variable que se encuentra en el hexágono y comprueba si es igual al valor que está definido en cada caso o línea del hexágono.

**Estructuras de control iterativas o repetitivas:** Ejecutan una serie de instrucciones mientras se cumpla la expresión lógica.

***(While) Mientras:*** Primero valida la condición, si es verdadera realiza las instrucciones, de lo contrario se rompe el ciclo y se continúa el flujo del diagrama.

***(Do while) Hacer Mientras:*** Primero se ejecutan las instrucciones y al final valida la condición.

Conociendo toda la información y estructura que debe llevar un diagrama de flujo se anexarán a la práctica 3 ejercicios con algoritmos y su respectivo diagrama.

Ejercicios

Fórmula general

Algoritmo:

1.-Inicio

2.- Pedir un valor para a

3.- Si a=0, regresar al paso 2 ya que no habría solución

4.- Pedir un valor para b y c

5.- Realizar d= b^2-4ac

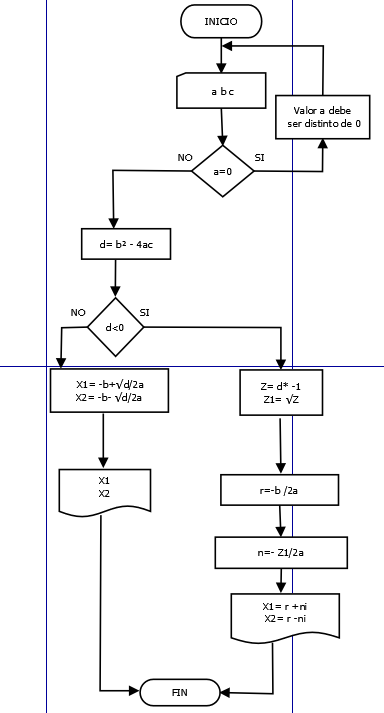
6.- Si d<0, entonces realizar y hacer la operación r=-b/2ª, n =z/2ª es imaginaria;

x1=r+ni x2=r-ni, en caso contrario ir al paso 7

7.- x= -b+/2a , -b =x2, imprimir x1 y x2

8.- Fin

Diagrama de flujo



Pedir 3 valores y determinar el triángulo formado

Algoritmo:

1.-Inicio

2.- Pedir valor para a

3.- Pedir valor para b

4.- Pedir valor para c

5.- Verificar que los valores a, b, c sean mayores que 0, de lo contrario regresar al paso 2 ya que no se formaría ningún tipo de triángulo

6.- Si a = b y a = c; mostrar en pantalla “el triángulo es equilátero”

7.- Si a ≠ b; a ≠ c y b ≠ c; mostrar en pantalla “el triángulo es escaleno”

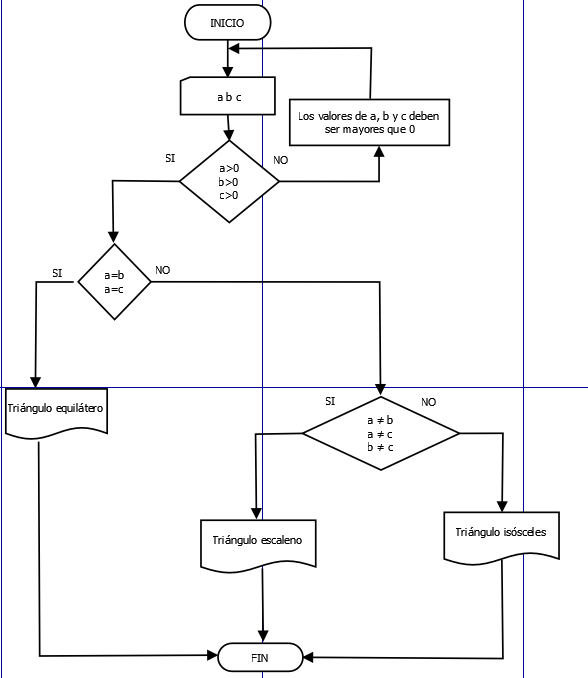
8.- Si a = b y a ≠ c; mostrar en pantalla “el triángulo es isósceles “

9.- Si a ≠ b y a = c; mostrar en pantalla “el triángulo es isósceles”

10.- Si a ≠ b y b = c; mostrar en pantalla “el triángulo es isósceles”

11.- Fin

Diagrama de flujo



Pedir 3 valores y decir si la suma de los 2 primeros números es igual al tercer número

Algoritmo:

1.-Inicio

2.- Pedir valor para a

3.- Pedir valor para b

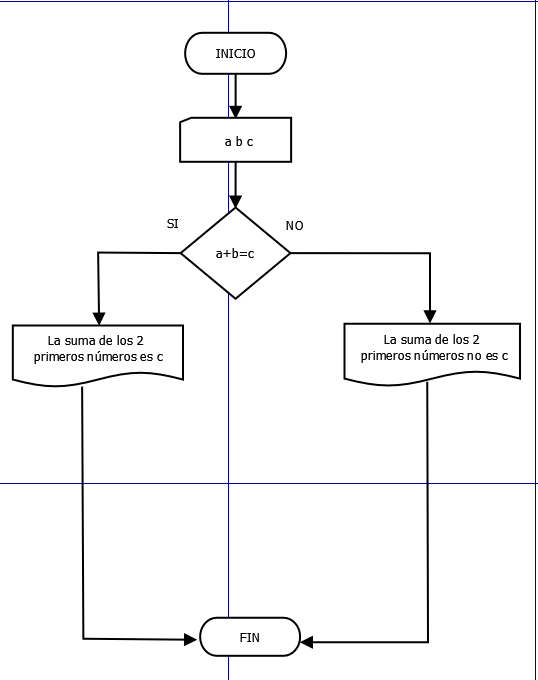
4.- Pedir valor para c

5.- Si a + b = c; mostrar en pantalla “la suma de los 2 primeros números es igual a c”

6.- Si a + b ≠ c; mostrar en pantalla “la suma de los 2 primeros números no es igual a c”

7.- Fin

Diagrama de flujo



Conclusiones: Esta práctica fue de gran utilidad ya que nos permitió conocer el concepto y la importancia que tienen los diagramas de flujo en el proceso del desarrollo de software, se aprendió a realizar y construir un diagrama de flujo adecuadamente utilizando los símbolos mencionados en la práctica que son fundamentales en el desarrollo de cualquier diagrama que se vaya a realizar. Además, considero que fue importante su estudio ya que nos permite analizar de una manera diferente la resolución de un problema y la realización de éstos diagramas nos introduce directamente a la codificación de un programa porque sabiendo realizar el algoritmo que resuelva el problema dado y poder comprobarlo con un diagrama de flujo realizado correctamente, prácticamente hace que sea posible la programación o codificación del problema en cualquier lenguaje de programación utilizado.