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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Alejandro Esteban Pimentel Alarcón |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 3 |
| *No de Práctica(s):* | 4 |
| *Integrante(s):* | Itzel Erendira Mojica Hernandez |
| *No. de Equipo de cómputo empleado:* | 54 |
| *No. de Lista o Brigada:* | 3850 |
| *Semestre:* | 2020-I |
| *Fecha de entrega:* | 10/09/19 |
| *Observaciones:* |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Diagramas de flujo**

Objetivo:

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

Introducción:

Un diagrama de flujo es la representación gráfica de un proceso, es decir, muestra gráficamente el flujo de acciones a seguir para cumplir con una tarea específica.

Dentro de las ciencias de la computación, un diagrama de flujo es la representación gráfica de un algoritmo. La correcta construcción de estos diagramas es fundamental para la etapa de codificación, ya que, a partir del diagrama de flujo es posible codificar un programa en algún lenguaje de programación.

Formas de los diagramas de flujo

Los diagramas de flujo poseen símbolos que permiten estructurar la solución de un problema de manera gráfica. A continuación se muestran los elementos que conforman este lenguaje gráfico.

1. Todo diagrama de flujo debe tener un inicio y un fin.

fin

Inicio

1. Las líneas utilizadas para indicar la dirección del flujo del diagrama deben ser rectas, verticales u horizontales, exclusivamente.
2. Todas las líneas utilizadas para indicar la dirección del flujo del diagrama deben estar conectadas a un símbolo.
3. 4. El diagrama debe ser construido de arriba hacia abajo (top-down) y de izquierda a derecha (left to right).
4. 5. La notación utilizada en el diagrama de flujo debe ser independiente del lenguaje de programación en el que se va a codificar la solución.
5. Se recomienda poner comentarios que expresen o ayuden a entender un bloque de símbolos.
6. Si la extensión de un diagrama de flujo ocupa más de una página, es necesario utilizar y numerar los símbolos adecuados.
7. A cada símbolo solo le puede llegar una línea de dirección de flujo.

Invalido valido

1. Notación de camello. Para nombrar variables y nombres de funciones se debe hacer uso de la notación de camello.

Los diagramas de flujo poseen símbolos que permiten estructurar la solución de un problema de manera gráfica. Por tanto es fundamental conocer los elementos que conforman este lenguaje gráfico.

Representa el inicio o el fin del diagrama de flujo.

Datos de entrada. Expresa lectura de datos.

Proceso. En su interior se expresan asignaciones u operaciones.

Decisión. Valida una condición y toma uno u otro camino.

si

no

Escritura. Impresión del o los resultado(s).

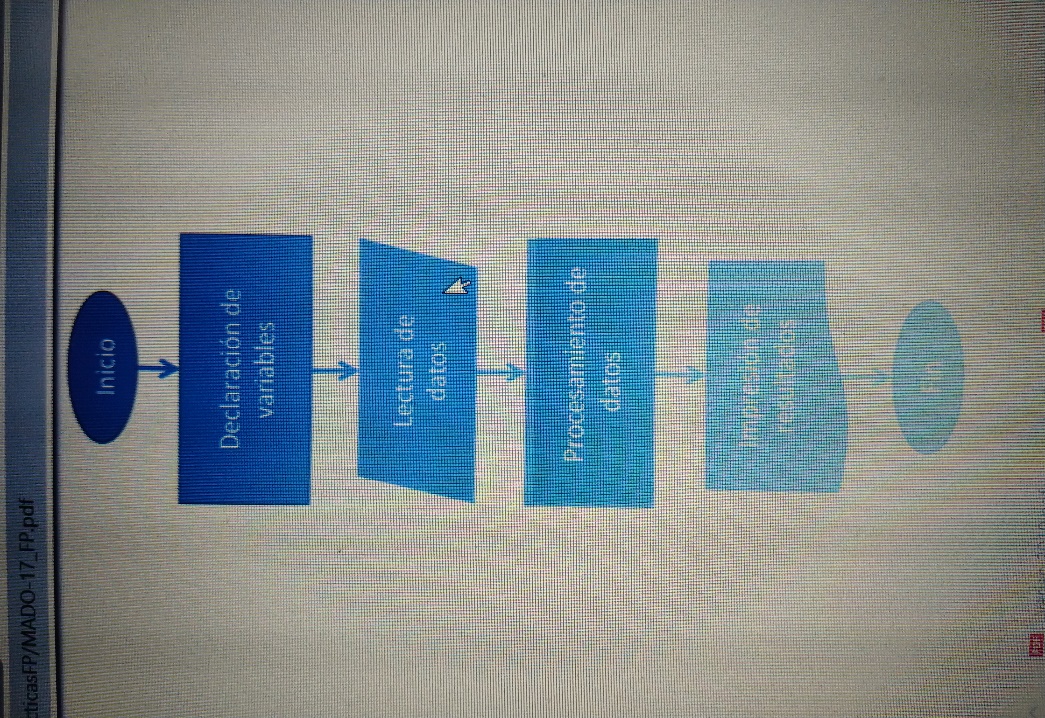
Conexión dentro de la misma página.

Conexión entre diferentes páginas.

Las estructuras de control de flujo.

Las estructuras de control de flujo permiten la ejecución condicional y la repetición de un conjunto de instrucciones. Existen 3 estructuras de control: secuencial, condicional y repetitivas o iterativas. Estructura de control secuencial Las estructuras de control secuenciales son las sentencias o declaraciones que se realizan una a continuación de otra en el orden en el que están escritas.

Estructuras de control condicionales (o selectivas).

 Las estructuras de control condicionales permiten evaluar una expresión lógica (condición que puede ser verdadera o falsa) y, dependiendo del resultado, se realiza uno u otro flujo de instrucciones. Estas estructuras son mutuamente excluyentes (o se ejecuta una acción o se ejecuta la otra). La estructura de control de flujo más simple es la estructura condicional.

Se evalúa la expresión lógica y si se cumple (si la condición es verdadera) se ejecutan las instrucciones del bloque [Acciones]. Si no se cumple la condición, se continúa con el flujo normal del programa.

La estructura condicional completa es SI-DE LO CONTRARIO (IF-ELSE):

Se evalúa la expresión lógica y si se cumple (si la condición es verdadera) se ejecutan las instrucciones del bloque Sí. Si no se cumple la condición se ejecutan las instrucciones del bloque No. Al final el programa sigue su flujo normal

Estructuras de control iterativas o repetitivas:

Las estructuras de controles de flujo iterativos o repetitivos (también llamadas cíclicas) permiten ejecutar una serie de instrucciones mientras se cumpla la expresión lógica.

Existen dos tipos de expresiones cíclicas MIENTRAS y HACER- MIENTRAS.

La estructura MIENTRAS primero valida la condición y si ésta es verdadera procede a ejecutar el bloque de instrucciones de la estructura, de lo contrario rompe el ciclo y continúa el flujo normal del programa

La estructura HACER-MIENTRAS primero ejecuta las instrucciones descritas en la estructura y al final valida la expresión lógica.

Si la condición se cumple vuelve a ejecutar las instrucciones de la estructura, de lo contrario rompe el ciclo y sigue el flujo del algoritmo. Esta estructura asegura que, por lo menos, se ejecuta una vez el bloque de la estructura, ya que primero ejecuta y después pregunta por la condición.

Funciones

Cuando la solución de un problema es muy compleja se suele ocupar el diseño descendente (divide y vencerás). Este diseño implica la división de un problema en varios subprocesos más sencillos que juntos forman la solución completa. A estos subprocesos se les llaman módulos o funciones.

Una función está constituida por un identificador de función (nombre), de cero a n parámetros de entrada y un valor de retorno.

nomFun es el nombre con el que llama a la función. Las funciones pueden o no recibir algún parámetro (tipo de dato) como entrada, si la función recibe alguno se debe incluir en el recuadro inicial (el que apunta al nombre de la función). Todas las funciones pueden regresar un valor al final de su ejecución (un resultado) para ello se debe definir el dominio del conjunto de salida (tipo de dato)

Descripción

La primera función que se ejecuta es 'principal', ahí se crean las variables (uno y dos) y, posteriormente, se manda llamar a la función 'sumar'. La función 'sumar' recibe como parámetros dos valores enteros y devuelve como resultado un valor de tipo entero, que es la suma de los valores que se enviaron como parámetro.

Para la función 'principal' los pasos que realiza la función 'sumar' son transparentes, es decir, solo manda a llamar a la función y espera el parámetro de retorno.

La siguiente figura permite analizar la función a través del tiempo. El algoritmo inicia con la función principal, dentro de esta función se hace una llamada a una función externa (sumar). Sumar realiza su proceso (ejecuta su algoritmo) y devuelve un valor a la función principal, la cual sigue su flujo hasta que su estructura secuencial llega a su fin.

Actividades:

* Diagrama de flujo que reciba un número del 1 al 7, y que indique a qué día de la semana corresponde.
* Diagrama de flujo que reciba tres números y verifique si son válidos como los ángulos de un triángulo.
* Diagrama de flujo que reciba tres números como los lados de un triángulo, que responda si se trata de un triángulo equilátero, isósceles, o escaleno.
* Diagrama de flujo que reciba tres números como los lados de un triángulo, y que responda si se puede formar un triángulo con lados de esa longitud, o no.
* Verificar las actividades anteriores con los datos:
* Números a días: 3,7,-2,0,9,"Lunes"
* Ángulos de triángulo:
* 30,30,120
* -90,90,180
* 0,30,150
* 270,60,30
* Tipos de triángulos:
* 45,50,80
* 20,20,20
* 10,100,10
* 0,4,20
* Triángulo aceptable:
* 20,40,20
* 60,100,200
* -3,6,12
* 4,5,9

Actividad 1:

* Diagrama de flujo que reciba un número del 1 al 7,

y que indique a qué día de la semana corresponde.

n

* Números a días: 3,7,-2,0,9,"Lunes"

no

si

n=1

si

no

n=2

“Lunes”

n=3

“Domingo”

si

no

“Martes”

no

n=4

si

n=5

si

no

“Miércoles””

n=6

“Jueves”

si

“Viernes”

n=7

si

“Sábado”

“este no es un día de la semana”

Actividad 2:

* Diagrama de flujo que reciba tres números y verifique si son válidos como los ángulos de un triángulo.
* Ángulos de triángulo:
* 30,30,120
* -90,90,180
* 0,30,150
* 270,60,30

Actividad 3:

* Diagrama de flujo que reciba tres números como los lados de un triángulo, que responda si se trata de un triángulo equilátero, isósceles, o escaleno.
* Tipos de triángulos:
* 45,50,80
* 20,20,20

a,b,c

* 10,100,10

a>b

* 0,4,20

a>c

si

no

b>c

S=a+b

S=C+b

s=a+c

s=a+b

Actividad 4:

* Diagrama de flujo que reciba tres números como los lados de un triángulo, y que responda si se puede formar un triángulo con lados de esa longitud, o no.
* Triángulo aceptable:
* 20,40,20
* 60,100,200
* -3,6,12
* 4,5,9