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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | RODRIGUES ESPINO CLAUDIA |
| *Asignatura:* | FUNDAMENTOS DE PROGRAMACION |
| *Grupo:* | 4 |
| *No de Práctica(s):* | PRACTICA 4: Diagrama de flujo |
| *Integrante(s):* | FLORES MENDOZA OLGA |
|  |  |
| *No. de Equipo de cómputo empleado* | 44 |
| *Semestre:* | 2019-2 |
| *Fecha de entrega:* | 9 DE MARZO DE 2019 |
| *Obervaciones:* |  |
|  |  |

**Diagrama de flujos.**

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**PRACTICA 4**

**OBJETIVOS:**

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

**ACTIVIDADES:**

* Elaborar un diagrama de flujo que represente la solución algorítmica de un problema, en el cual requiera el uso de la estructura de control condicional.
* Elaborar la representación gráfica de la solución de un problema, a través de un diagrama de flujo, en el cual requiera el uso de la estructura de control iterativa.

**INTRODUCCION:**

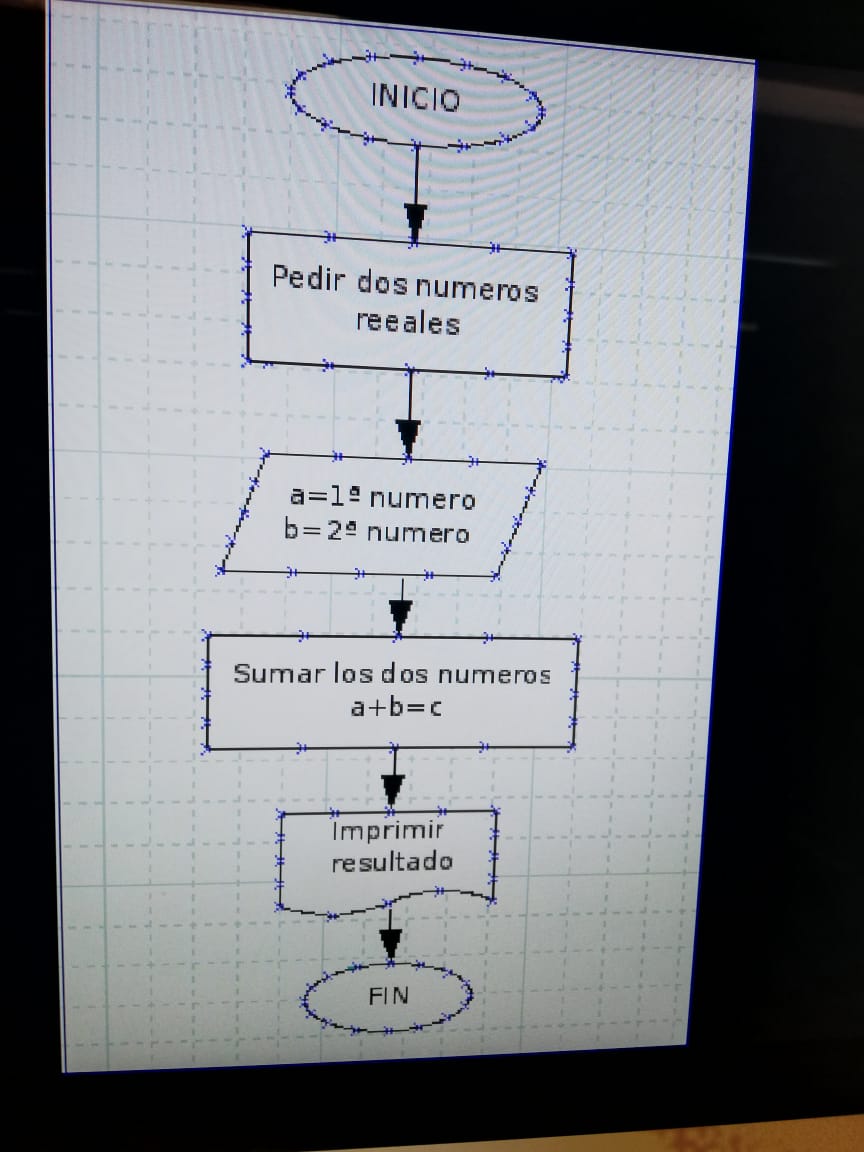
Un diagrama de flujo es la representación gráfica de un proceso (de un algoritmo). La correcta construcción de estos diagramas es fundamental para la etapa de codificación, ya que, a partir del diagrama de flujo es posible codificar un programa en algún lenguaje de programación. Se pueden construir de la siguiente manera:

1. Todo diagrama de flujo debe tener un inicio y un fin.
2. Se utilizan líneas para indicar la dirección del flujo del diagrama. Deben ser rectas, verticales u horizontales, exclusivamente y están conectadas a un símbolo.
3. El diagrama debe ser construido de arriba hacia abajo y de izquierda a derecha.
4. 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. Notación de camello. Para nombrar variables y nombres de funciones se debe hacer uso de la notación de camello.

|  |  |
| --- | --- |
| *Representa* | *SIMBOLO* |
| El inicio o el fin |  |
| 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. |  |
| Escritura. Impresión del o los resultado(s). |  |
| Conexión dentro de la misma página |  |
| Decisión múltiple. Almacena un selector que determina la rama por la que sigue el flujo. |  |

**DESARROLLO DE ACTIVIDADES:** A continuación se presentaran varios tipos de estructura y ejemplos de los mismos.

**Diagrama de flujos.**

* **Estructura de control secuencial:**

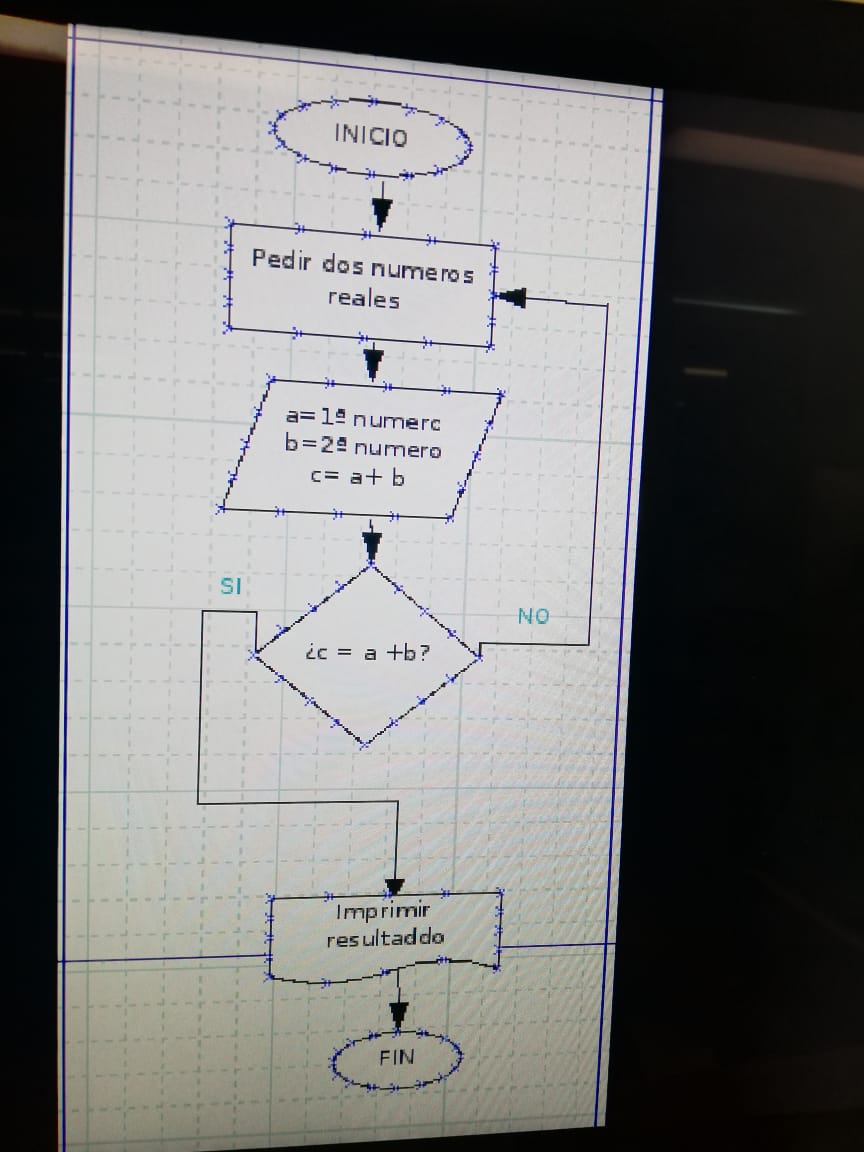
Las estructuras de control secuenciales son las sentencias o declaraciones que se realizan una a continuación de otra en el orden en el que están escritas.

**ALGORITMO:** La suma de dos números.

1. INICIO
2. Pedir un primer numero
3. Pedir un segundo número
4. Realizar la suma, el primer número más el segundo numero
5. Imprimir resultado
6. FIN

* **Estructuras de control iterativas o repetitivas:**

**Diagrama de flujos.**

Las estructuras de control de flujo iterativas o repetitivas (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.

**ALGORITMO:**

1. INICIO
2. Pedir un primer numero
3. Pedir un segundo número
4. Realizar la suma, el primer número más el segundo numero
   1. Si c = a + b, ir al paso 5
   2. Si c es diferente de a + b regresar al paso 3
5. Imprimir resultado.
6. FIN

**ALGORITMO:** Obtener el factorial de un numero.

1. INICIO
2. Pedir un numero, n
   1. Crear una variable llamada contador que inicie en 1 y una llamada factorial con inicio en 2.
   2. Si la variable del contador es menor o igual a numero dado se realiza lo siguiente:
   3. El valor de la variable del contador se multiplicara por el factorial.
   4. Guardar el resultado en la variable factorial.
   5. Se le incrementa un valor al contador.
   6. Ir al paso 3
3. Si la variable del contador no es menor o igual al número dado, se muestra el resultado guardado en la variable factorial.
4. Imprimir el resultado del factorial
5. FIN

INICIO

**DIAGRAMA DE FLUJOS.**

Pedir un número (n)

Contador= 1= C , Factorial=2

SI

NO

C=1<= n

C x Factorial =

Se guarda el resultado en el Factorial

C+1=

C no es <=n

Imprimir Factorial

FIN

**ALGORITMO:** Resolver la Formula General

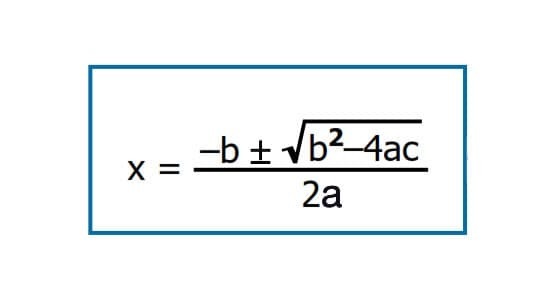
1. INICIO
2. Pedir tres números reales, diferente de cero.
   1. a= 1ª numero, b= 2ª numero, c= 3ª numero.
3. Elevar b al cuadrado y restarle cuatro veces el producto de a\*c, finalmente se le sacara raíz cuadrada. Y esto ser a igual al resultado= R
4. Si el resultado < 0
   1. No hay solución, se trata de raíces imaginarias (negativas).
5. Si el resultado es > 0
   1. Se hace la siguiente operación: - b + resultado, y también –b – resultado, dividido entre dos.
   2. Se obtienen dos resultados finales X1, X2.
6. Imprimir resultado de X1, X2^
7. FIN

INICIO

**DIAGRAMA DE FLUJO.**

Pedir números

a , b ,c

= R

SI

NO

¿R<0?

1. (-b + R)/2
2. (-b –R)/2

Raíces imaginarias

R + i, R - i

X1= a)

X2= b)

Imprimir resultado

FIN

* **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:

**ALGORITMO:**

1. INICIO
2. Pedir X, debe de ser:
   1. X > 2
   2. X <2
3. Si X >2, realizar la siguiente función, teniendo como enteros a 3, 3, -25: 3X2 + 3X - 25
4. Si X <2, realizar la siguiente funcione, teniendo como enteros a 2,-3,8: 2X2  - 3X + 8
5. Obtener los valores cuando, X<0, X>0.
6. FIN

INICIO

**DIAGRAMA DE FLUJOS.**

Pedir en valor de X diferente de 2

SI

NO

¿X > 2?

Realizar la suma de los productos:

2X2  - 3X + 8

Realizar la suma de lo productos:

3X2 + 3X - 25

Imprimir resultado

Imprimir resultado

FIN

**CONCLUSIONES:**

Después de realizar el algoritmo, viene la creación del diagrama de flujos, es cual es un poco mas explicito, ya que ver de manera grafica un procedimiento, a veces llega ser más factible. Con el uso de las flechas y símbolos seguir el procedimientos no es difícil. Pero para lograr un buen diagrama de flujos se debe ser cuidadoso en su creación, ya que al igual que hicimos con el algoritmo, debe de ser congruente y no caer en ambigüedades. Aprendí que hay ocasiones en las que la solución de un problema se vuelve complicada, para esto en recomendadle fragmentar el problema en pequeños procesos, y así la solución se vuelve más sencilla. Es muy importante conocer y hacer buen uso de la simbología ya que esta nos está mostrando que es lo que se debe de hacer.