



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

# Laboratorio de Computación

## Salas A y B



*Profesor:* García Morales Karina

*Asignatura:* Fundamentos de Programación

*Grupo:* 1121

*No de Práctica(s):* 4

*Integrante(s):* Mena Hernández Hebbby Renan

*No. de Equipo de  
cómputo empleado:* 4

*Semestre:* 2019-1

*Fecha de entrega:* 29-Agosto-2018

*Observaciones:*

CALIFICACIÓN: \_\_\_\_\_

## **Título de la práctica**

### 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 de la práctica**

Un diagrama de flujo es la representación gráfica del flujo o secuencia de rutinas simples. Tiene la ventaja de indicar la secuencia del proceso en cuestión, las unidades involucradas y los responsables de su ejecución, es decir, viene a ser la representación simbólica o pictórica de un procedimiento administrativo.

Luego, un diagrama de flujo es una representación gráfica que desglosa un proceso en cualquier tipo de actividad a desarrollarse tanto en empresas industriales o de servicios y en sus departamentos, secciones u áreas de su estructura organizativa.

Son de gran importancia ya que ayudan a designar cualquier representación gráfica de un procedimiento o parte de este. En la actualidad los diagramas de flujo son considerados en la mayoría de las empresas como uno de los principales instrumentos en la realización de cualquier método o sistema.

Ventajas.-

Podemos citar como ventajas que se pueden obtener con la utilización de los diagramas de flujo, las siguientes:

Ayudan a las personas que trabajan en el proceso a entender el mismo, con lo que facilitan su incorporación a la organización e incluso, su colaboración en la búsqueda de mejoras del proceso y sus deficiencias.




Al presentarse el proceso de una manera objetiva, se permite con mayor facilidad la identificación de forma clara de las mejoras a proponer.


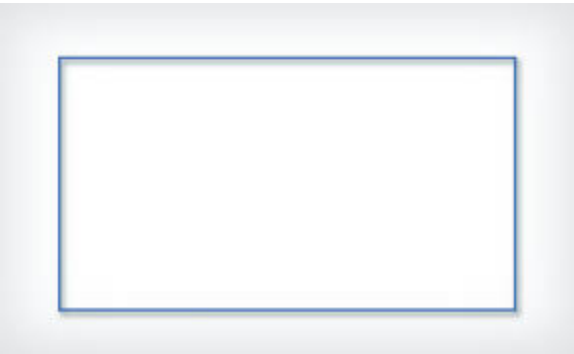
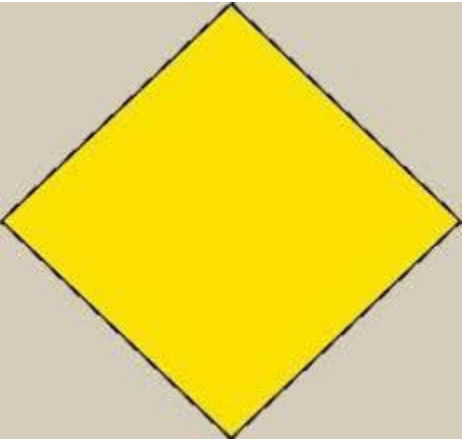

Permite que cada persona de la empresa se sitúe dentro del proceso, lo que conlleva a poder identificar perfectamente quien es su cliente y proveedor interno dentro del proceso y su cadena de relaciones, por lo que se mejora considerablemente la comunicación entre los departamentos y personas de la organización.

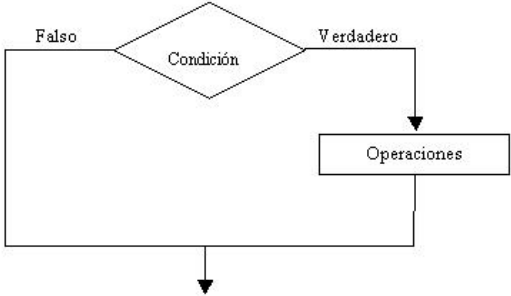
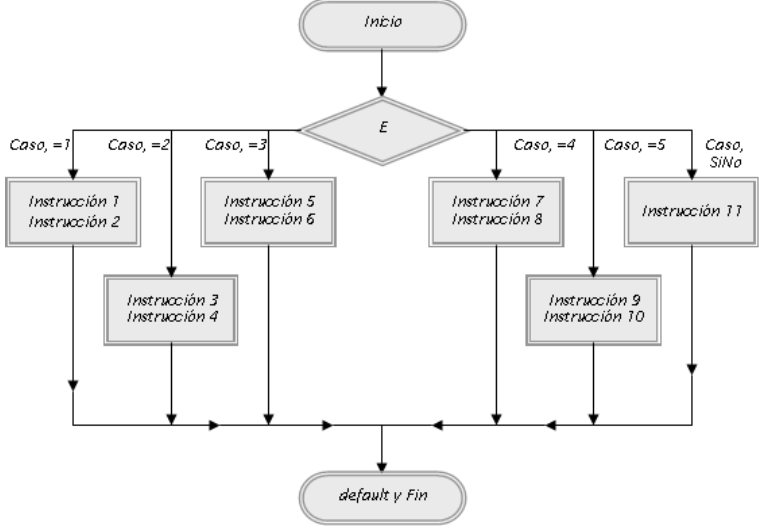
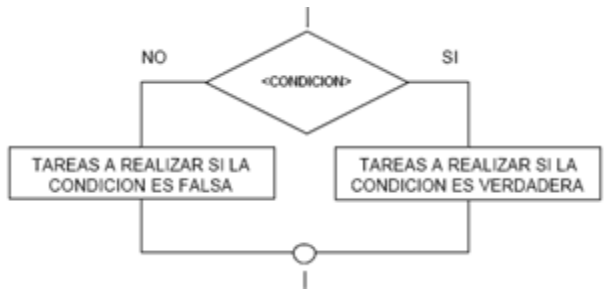
Normalmente sucede que las personas que participan en la elaboración del diagrama de flujo se suelen volver entusiastas partidarias del mismo, por lo que continuamente proponen ideas para mejorarlo.

Es obvio que los diagramas de flujo son herramientas muy valiosas para la formación y entrenamiento del nuevo personal que se incorpore a la empresa.

Lo mas reseñable es que realmente se consigue que todas las personas que están participando en el proceso lo entenderán de la misma manera, con lo que será mas fácil lograr motivarlas a conseguir procesos mas económicos en tiempo y costes y mejorar las relaciones internas entre los cliente-proveedor del proceso.

Figura	Para que se utiliza
 <pre> graph TD     Inicio([Inicio]) --&gt; Dots[...]     Dots --&gt; Fin([Fin])         </pre>	Inicio/ Fin
	Imprimir
	Lectura de datos, solo se pone la variable (sin letretros, comentarios, etc)

	<p>Conector de página con número</p>
	<p>Operaciones/ asignaciones</p>
	<p>Solo se ponen condiciones sin texto</p>
	<p>Conector de hoja</p>

	<p>Condicional Simple</p> <p>Después de la condición solo se tiene una opción.</p>
	<p>Condicional Múltiple</p> <p>Después de la condición se tienen 3 opciones con una de default. Es útil para la creación de menús</p>
	<p>Condicional Compuesta</p> <p>Después de la condición solo se tienen 2 opciones.</p>

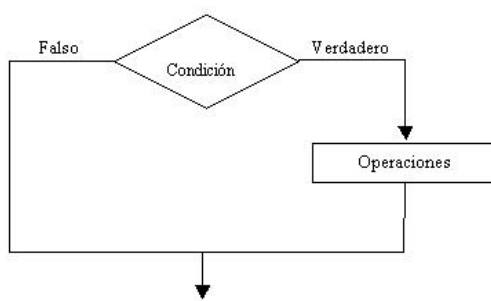
## Definición de las estructuras

Secuencial:

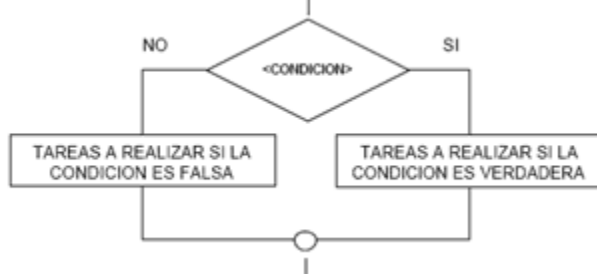
Son los pasos para la declaración de variables y la asignación de valores

Condicional o de Selección

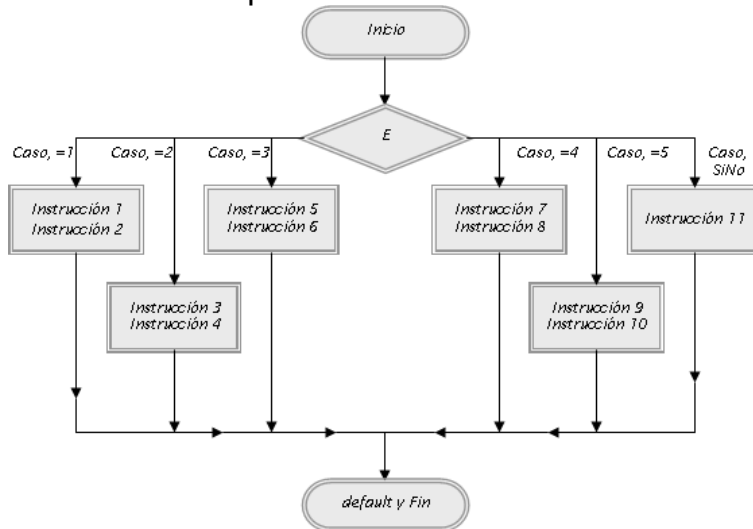
## Condicional Simple



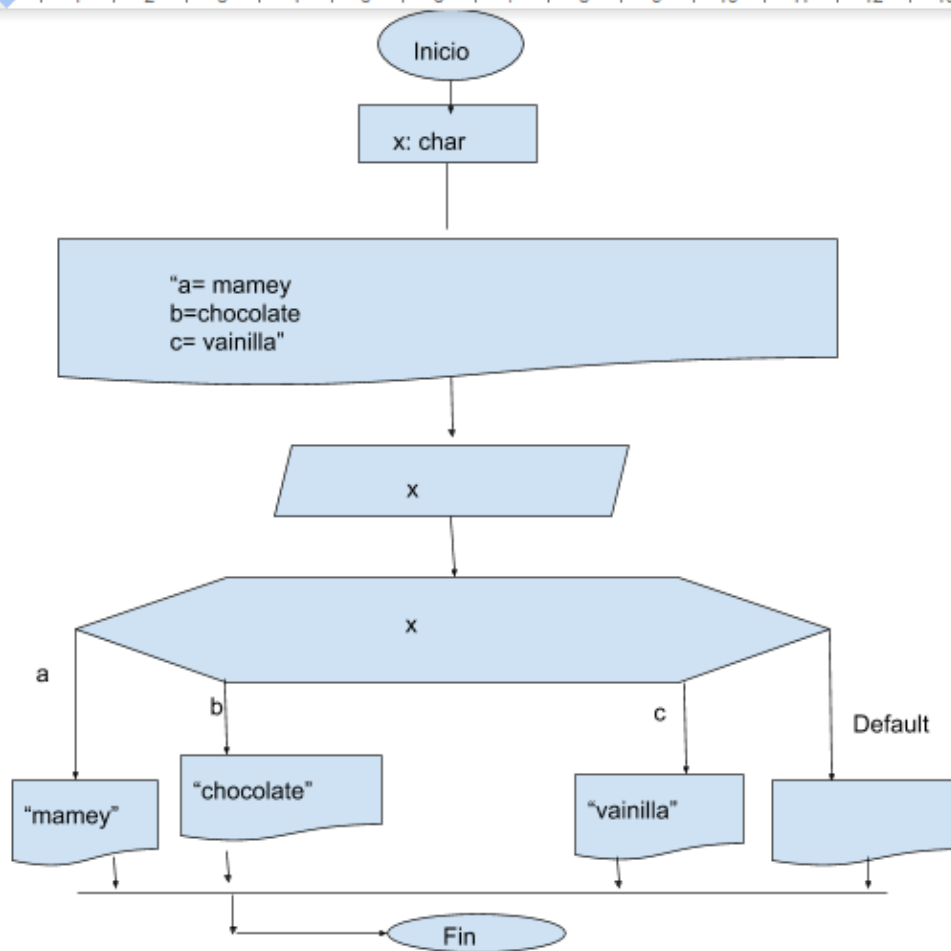
## Condicional Compuesto



## Condicional Múltiple



## Diagrama de flujo que da los sabores de helado

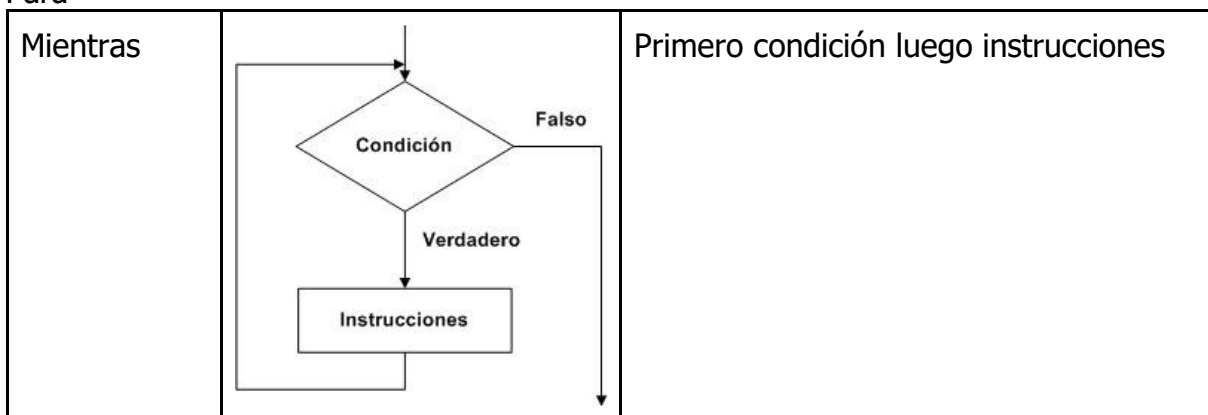


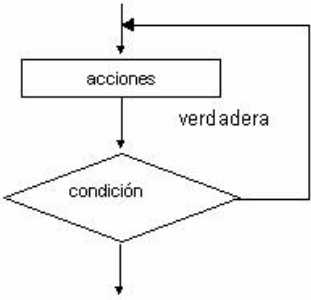

## Estructura Iterativa

Mientras

Hacer mientras

Para

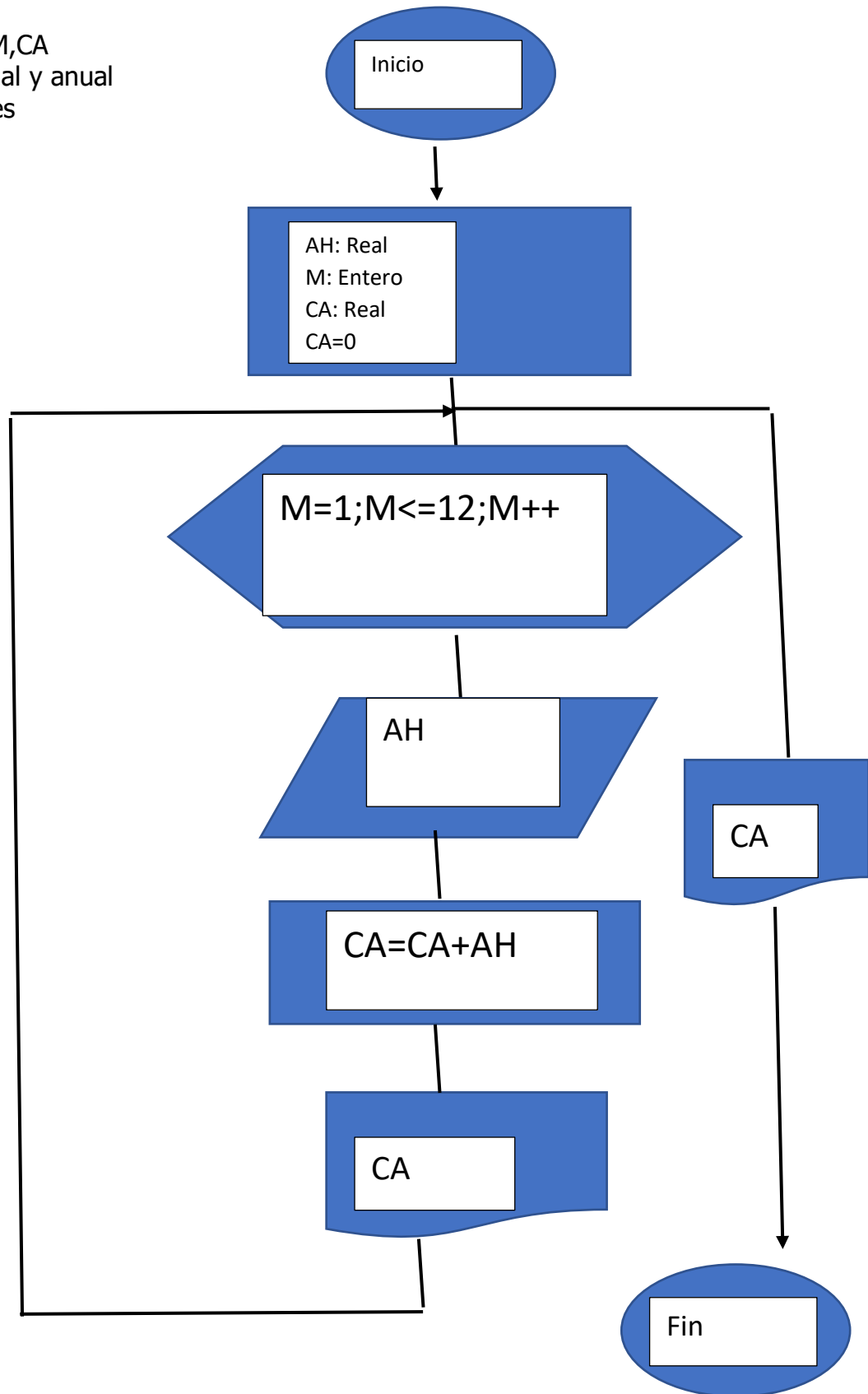


<p>Hacer mientras</p>	 <pre> graph TD     Entry(( )) --&gt; Acciones[acciones]     Acciones --&gt; Condicion{condición}     Condicion -- verdadera --&gt; Entry     Condicion --&gt; Exit(( )) </pre>	<p>Primero instrucciones luego operación</p>
<p>Para</p>	 <pre> graph TD     Entry(( )) --&gt; Hexagon{{&lt;variable&gt; ← &lt;exp1&gt; hasta&lt;br&gt;&lt;exp2&gt; paso &lt;exp3&gt;}}     Hexagon --&gt; Group[&lt;GRUPO DE INSTRUCCIONES&gt;]     Group --&gt; Circle(( ))     Circle --&gt; Entry </pre>	<p>Solo tiene 3 valores: Inicio, condición, incremento.</p>



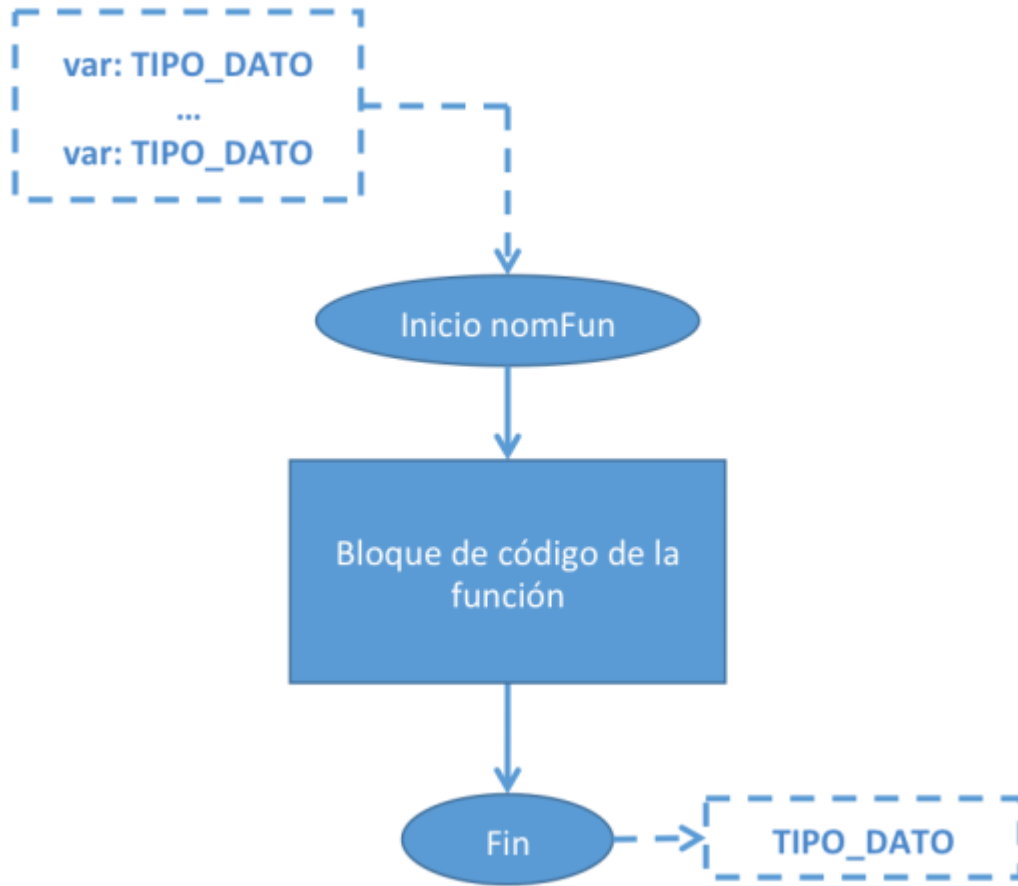
### Problema del examen

Análisis Entrada: AH,M,CA  
Salida Cantidad mensual y anual  
Restricciones 12 meses



## 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.



## Ejercicio de tarea

Análisis algoritmo diagrama de flujo pruebas de escritorio

1:Calculadora de dos números (/,\*,-, +)

Diagrama de funciones

y completo

Análisis:

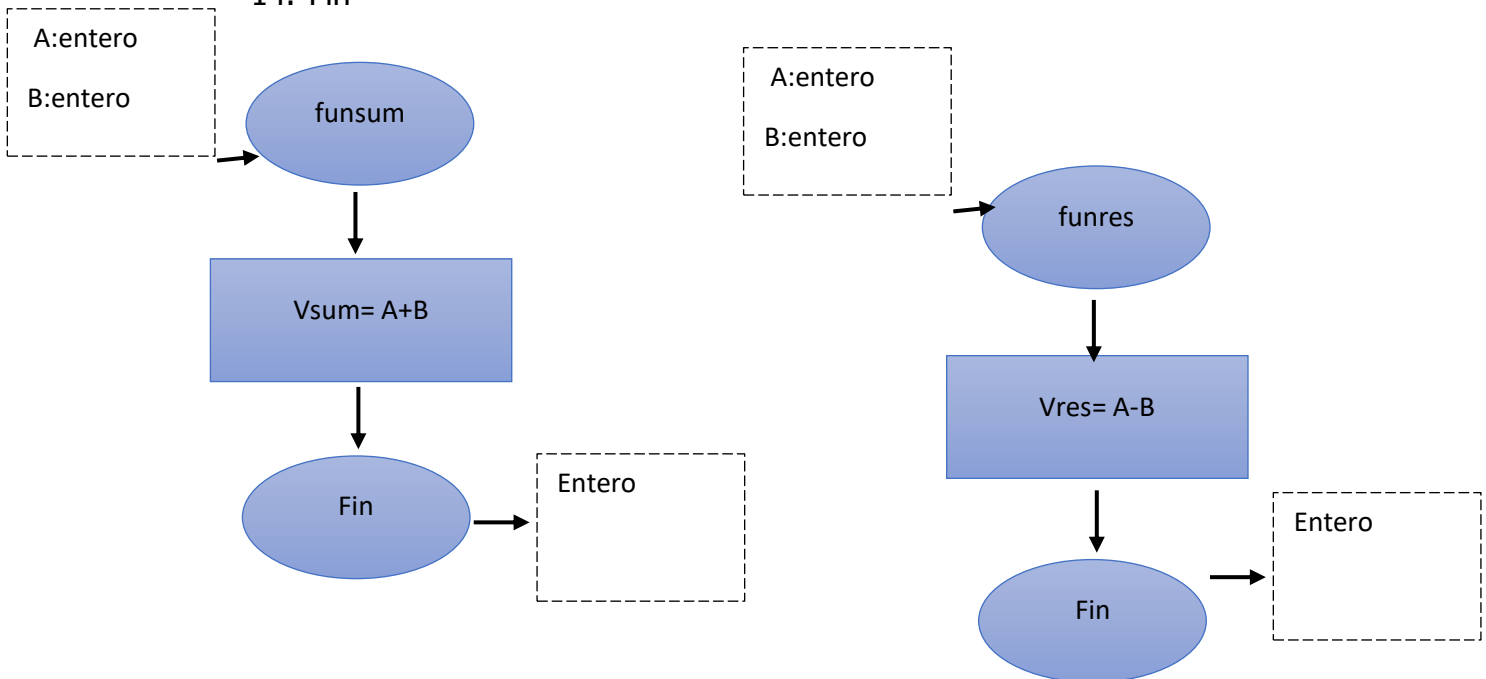
Entrada: 2 números (a,b) las funciones que se llamarán

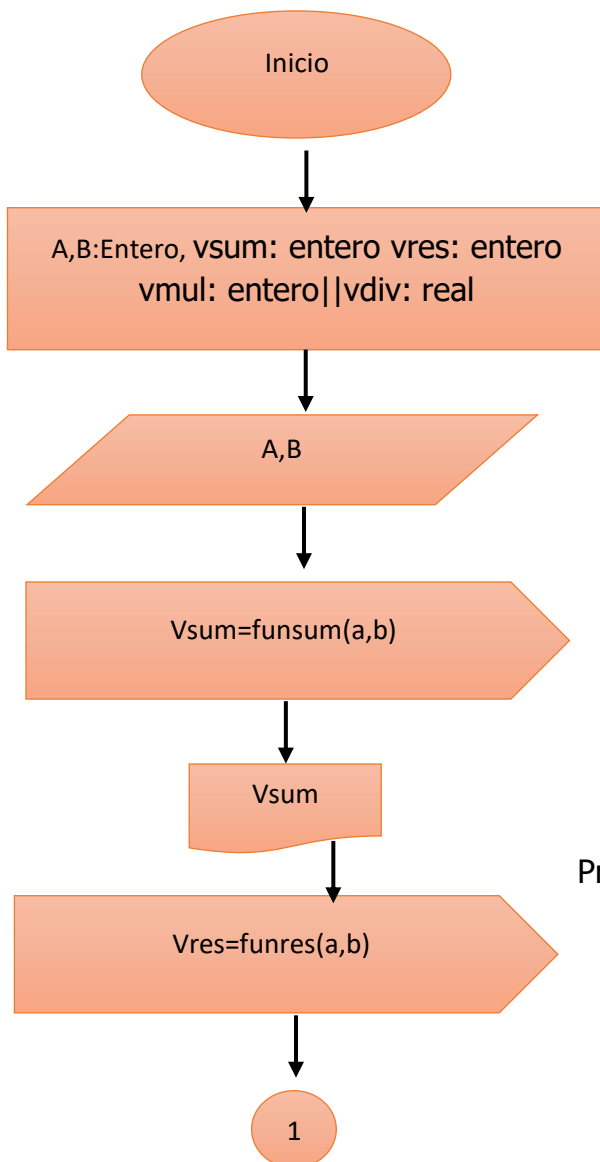
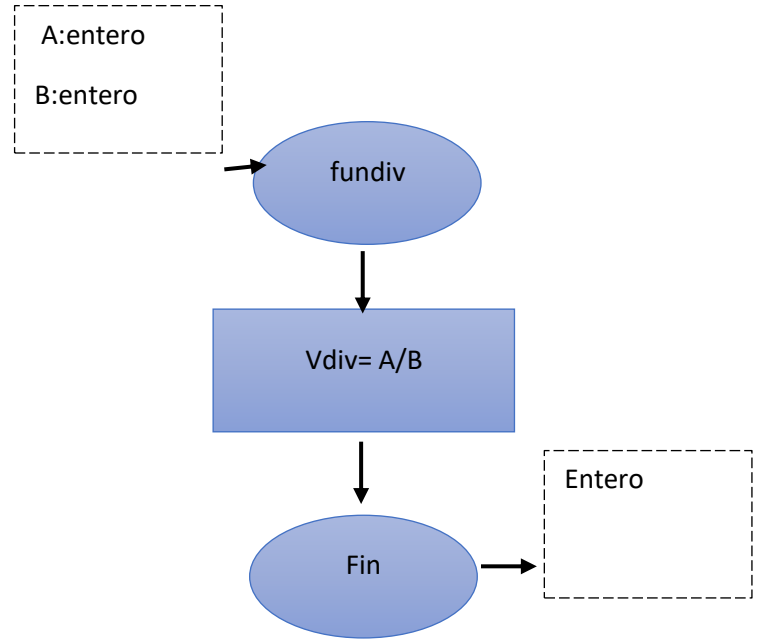
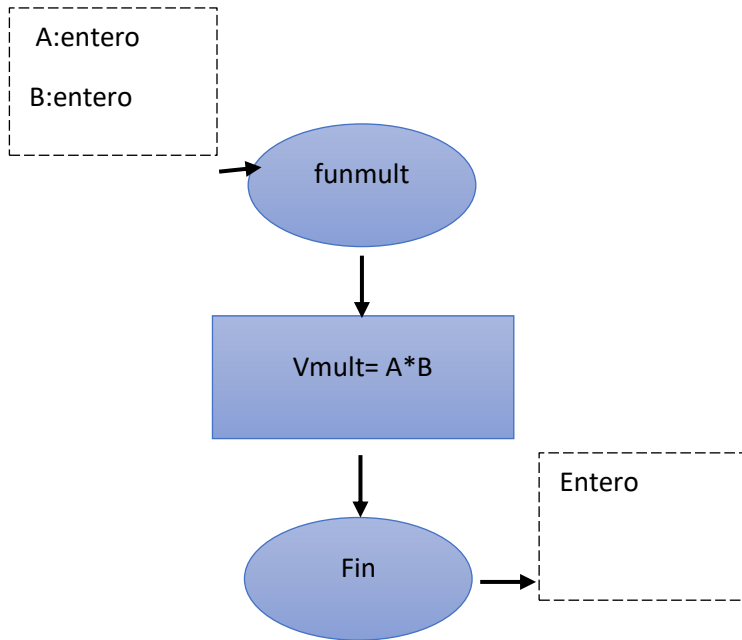
Restricciones solo se ingresan números enteros.

Salida: Se imprimirá después de cada operación el resultado.

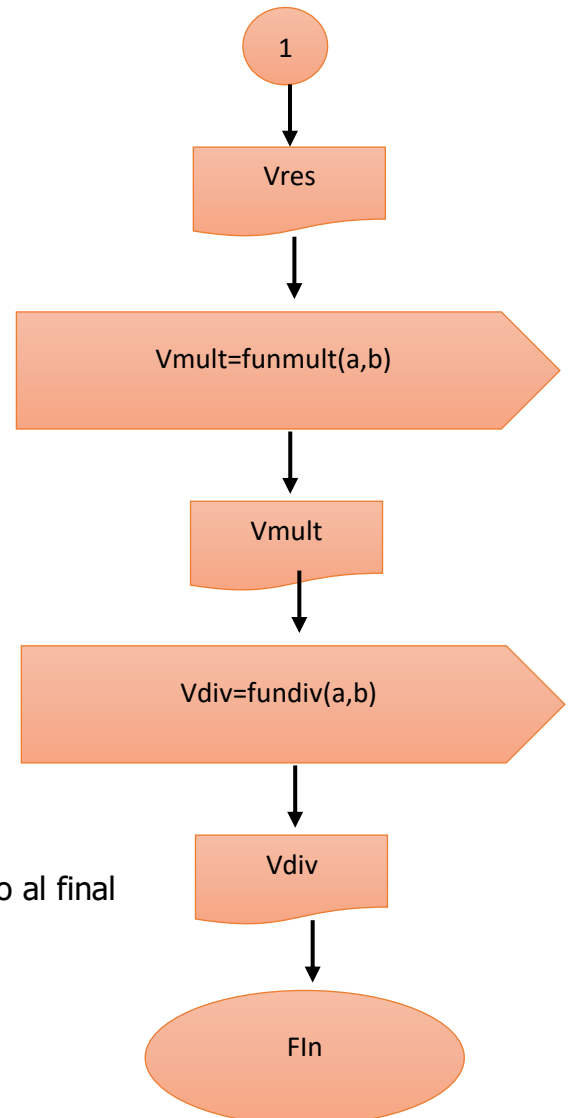
Algoritmo

1. Inicio
2. Declarar a y b : entero vsum: entero vres:entero vmul: entero||vdiv:real
3. "Ingresa valor de a y b
4. Leer el valor de a y b
5. Crear función para la suma, resta, multiplicación y división,
6. Llamar a la función suma y realizar operación.
7. Imprimir suma
8. Llamar a la función resta y realizar operación.
9. Imprimir resta
10. Llamar a la función multiplicación y realizar operación.
11. Imprimir la multiplicación
12. Llamar a la función división y realizar operación.
13. Imprimir la división
14. Fin





Prueba de escritorio al final



## 2 Menú de deportes

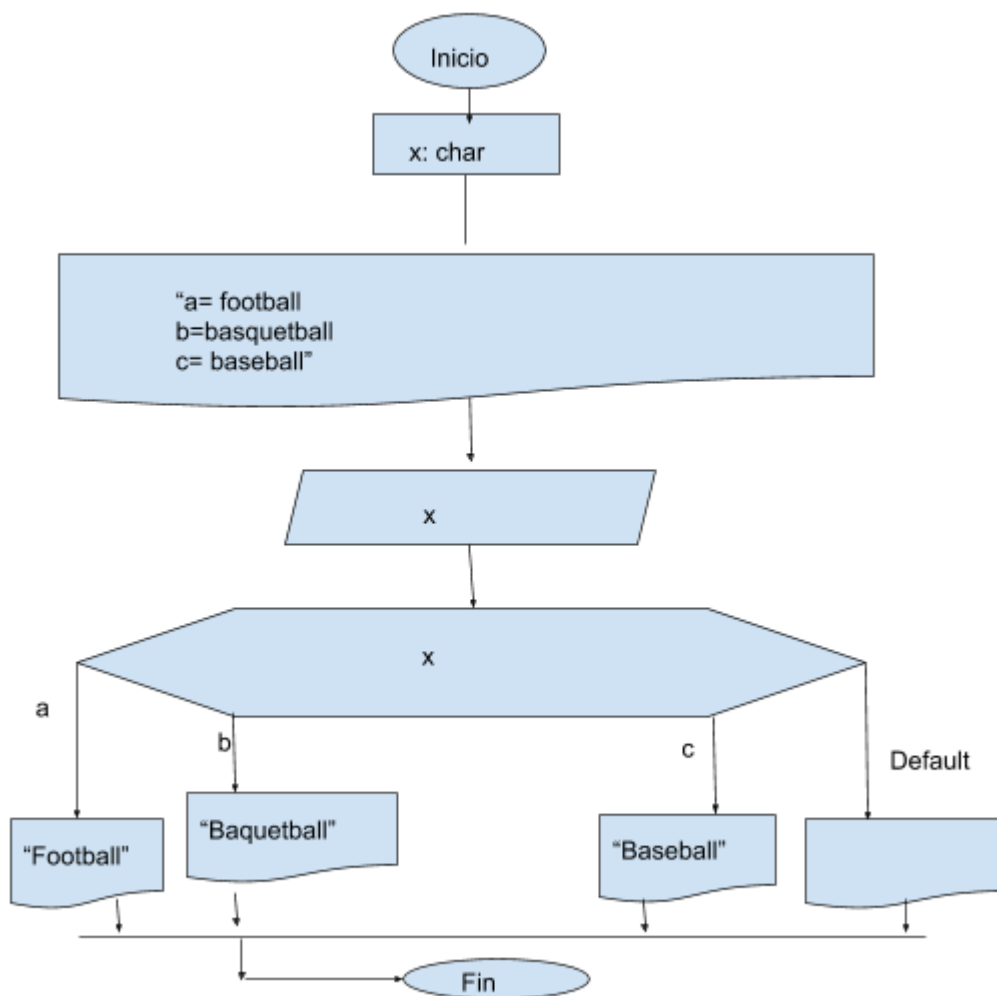
Análisis: Una variable donde el usuario ingresará su preferencia, las opciones del menú.

Restricciones: solo 3 opciones

Salida: La elección de deporte

Algoritmo:

1. Inicio
2. Dar el menú con a=football, b=basketball, c=baseball.
3. Leer x
4. Abrir condicional múltiple.
5. Desplegar las opciones
6. Fin



Prueba de escritorio al final

3 Tablas de multiplicar del 1-10, el usuario proporciona el valor a calcular

Análisis

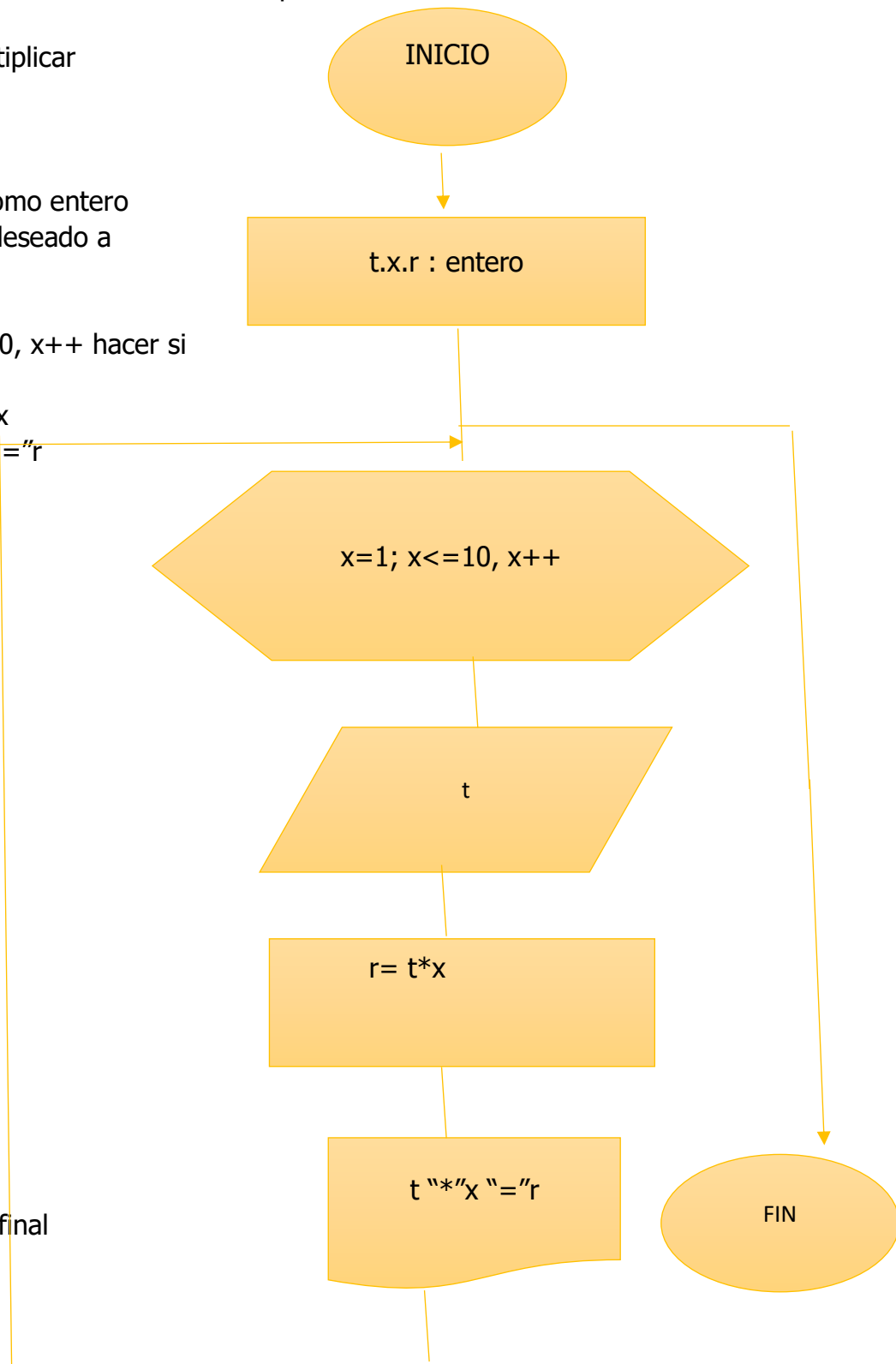
Entrada: Se dará el número de la tabla a multiplicar.

Restricciones: del 1-10

Salida: La tabla de multiplicar

Algoritmo

1. Inicio
2. Declarar t, x,r como entero
3. "Dame el valor deseado a multiplicar"
4. Leer t
5. Para x=1; x<=10, x++ hacer si no ir a fin
6. Multiplicar  $r = t * x$
7. Imprimir  $t * x = r$
8. Fin para
9. Fin proceso



Prueba de escritorio al final

## Pruebas de escritorio

### 1 Calculadora

Iteración	A	B	La Salida +	-	*	/
1	5	2	7	3	10	2.5
2	3	1	4	2	3	3
3	8	4	12	4	32	2

### 2 Menú

Iteración	X	Salida
1	A	Football
2	B	Basketball
3	C	Baseball

### 3 Tabla de multiplicar

Iteración	T	X	Salida R
1	2	1	2 "*" 1 "=" 2
2	3	3	3 "*" 3 "=" 9
3	4	5	4 "*" 5 "=" 20

## **Conclusiones**

La práctica fue de gran utilidad para la reafirmación de conceptos que pudieron haber quedado como lagunas en clase y que de manera práctica se ven más claras y detalladas además de que tenemos contacto directo con la computadora y nos vamos adaptando más.

El inicio de uso de funciones es clave, aunque se vieron de manera sencilla es un buen comienzo para que en un futuro sea más fácil ver la gran utilidad que tienen y el tiempo que nos pueden reducir si las aprendemos a usar correctamente.

El no crear el diagrama de flujo con una aplicación no me deja tan claro la solución si es que algo no hice bien así que esperaré en la próxima práctica utilizar un programa que me ayude a generar el diagrama de flujo.

## **Bibliografía**

Metodología de la programación. Osvaldo Cairó, tercera edición, México D.F., Alfaomega 2005.

Metodología de la programación a través de pseudocódigo. Miguel Ángel Rodríguez Almeida, primera edición, McGraw Hill