

**Universidad Autónoma del Estado de México
Unidad Académica Profesional Tlanguistenco
Licenciatura de Ingeniería en Producción Industrial**



**Manual de Practicas de Laboratorio:
Programación I**

Elaborado por:

M. en C. Rocío Elizabeth Pulido Alba

M. en C. Angélica Millán Díaz

Agosto 2017

Índice

	Pág.
I. Datos de identificación	4
II. Presentación del Manual de Prácticas para Programación	5
III. Ubicación de la unidad de aprendizaje en el mapa curricular	7
IV. Objetivos de la formación profesional	7
V. Objetivos de la unidad de aprendizaje	9
VI. Contenidos de la unidad de aprendizaje	10
VII Prácticas Resueltas	12
Práctica 1 El área de un círculo	12
Práctica 2 La suma de dos números enteros	15
Práctica 3 Ejercicios con Operadores	18
Práctica 4 El valor de x dada la función $f(x) = x^3 / 2(x^2)$	21
Práctica 5 El promedio de 5 calificaciones y saber si exenta o no con mínimo 8.0	25
Práctica 6 Transformación de temperaturas de acuerdo a la elección del usuario	32
Práctica 7 Operaciones Básicas	38
Práctica 8 Suma valores mientras el resultado sea menor a 150	45

	Pág.
Práctica 9 Tabla de Multiplicar que el usuario indique	50
Práctica 10 Suma de la secuencia $1^1+2^2+3^3+N^N$ leyendo el valor de N	55
Práctica 11 Números Arábigos a Romanos	60
Práctica 12 Arreglos en Matlab	71
Práctica 13 Llenado de un vector	72
Práctica 14 Decimal a Binario	78
VIII. Practicas Sugeridas y ejercicios	83
IX. Mapa curricular	91
X Bibliografía	93
XI Anexos	94
Anexo A Glosario de Términos	94
Anexo B Símbolos del Diagrama de Flujo	97
Anexo C Diagramas de Flujo Sentencias Selectivas y de Repetición	98
Anexo D El área de un círculo	102
Anexo E Obtener la suma de dos números enteros	103
Anexo F Formato de Reporte de Práctica	104
Notas	106



II. Presentación del Manual de Prácticas

Programación I es una unidad de aprendizaje que pertenece al área curricular de Ciencias de la Ingeniería, en el programa educativo de la Licenciatura en Ingeniería en Producción Industrial, que se imparte en la Unidad Académica Profesional Tianguistenco y en la Unidad Académica Profesional de Acolman.

Tiene como objetivo que, el estudiante comprenda la metodología de solución para problemas en el campo de la ingeniería, desarrollando programas basados en el paradigma de programación imperativa, proponiendo soluciones de calidad que permitan el manejo automatizado de estructuras de datos unidimensionales, así como el uso de condicionales y ciclos, como herramientas de solución.

Por tal motivo, se plantea un esquema metodológico con el cual se resolverán los ejercicios propuestos para prácticas, tal como se indica en la siguiente lista:

- 1) **Planteamiento del problema.** Descripción del contexto del problema, indicado a partir de elementos textuales, fórmulas, demostraciones en notación específica derivado del mismo, etc., los cuales indican las condiciones específicas que caracterizan al problema.
- 2) **Análisis del problema.** Desarrollado en tres partes: Entrada, Proceso y Salida. La **Entrada** especifica la cantidad de valores iniciales (asignados por medio de teclado o inicialización por defecto), el tipo de dato con el cual se puede relacionar (entero, real, caracter) para almacenar una determinada información, así como el identificador (nombre de la variable) con la cual se le denominará, desde la implementación del algoritmo hasta la codificación. El **Proceso**, consiste en determinar las operaciones de asignación de valores iniciales, asignación mediante teclado, asignación a partir de operaciones aritméticas, control de toma de decisiones dentro del programa a partir de sentencias condicionales, o de sentencias iterativas, las cuales ayudan a modelar cada uno de los procesos del problema, descritos con anterioridad en el planteamiento. La Salida como tal indicará las variables (cantidad, tipo e identificador) que permitirán desplegar los valores que indican la solución al problema, modelando el formato en el que se desplegará en pantalla para el usuario.
- 3) **Definición del algoritmo.** A través de las operaciones de proceso definidas en el análisis, se propone una estructura algorítmica en forma de pseudocódigo, el cual se lleva a cabo en el programa Pseint, para posteriormente realizar la validación del mismo a través de una prueba de escritorio.
- 4) **Prueba de escritorio.** Consiste en verificar que el algoritmo realice las operaciones definidas para solucionar el problema, distribuyendo en columnas los nombres de las variables, así como de incluir sentencias condicionales, para observar cómo van cambiando los valores de dichas variables a lo largo de la ejecución del algoritmo, así como de verificar que el algoritmo tenga una secuencia lógica de pasos. Durante este proceso es posible identificar errores,



los cuales deben corregirse hasta que el algoritmo cumpla con las especificaciones del problema.

- 5) **Codificación.** Una vez que se ha comprobado que las operaciones del algoritmo se realizan bajo las condiciones especificadas en el problema, se da paso a la traducción del mismo a un lenguaje de programación, para lo cual se utiliza el programa Matlab para la codificación del algoritmo. Una vez codificada la solución, se muestran los resultados de la ejecución con los valores de la prueba de escritorio, con la finalidad de validar que sean los mismos que en dicha prueba.

Para las primeras 2 unidades del programa de estudios, las prácticas se diseñan a nivel de Algoritmo y Pseudocódigo, y Diagrama de Flujo formalmente a partir de la unidad 3 se lleva a cabo el proceso de realizar la codificación en Matlab.

Al término de las prácticas resueltas, se presenta un conjunto de ejercicios propuestos, así mismo como información adicional, incluida en anexos, que permitirán complementar el conocimiento adquirido.



III. Ubicación de la unidad de aprendizaje en el mapa curricular

Núcleo de formación:	Básico
Área Curricular:	Ciencias de la Ingeniería
Carácter de la UA:	Obligatoria

IV. Objetivos de la formación profesional.

Objetivos del programa educativo:

Son objetivos de la Licenciatura de Ingeniería en Producción Industrial, formar profesionistas que contribuyan al progreso social, económico y cultural del país, y desarrollar en los alumnos los aprendizajes y competencias para:

Generales

- Ejercer el diálogo y el respeto como principios de la convivencia con sus semejantes, y de apertura al mundo.
- Reconocer la diversidad cultural y disfrutar de sus bienes y valores.
- Convivir con las reglas de comportamiento socialmente aceptables, y contribuir en su evolución.
- Adquirir los valores de cooperación y solidaridad.
- Cuidar su salud y desarrollar armoniosamente su cuerpo; ejercer responsablemente y de manera creativa el tiempo libre.
- Ampliar su universo cultural para mejorar la comprensión del mundo y del entorno en que vive, para cuidar de la naturaleza y potenciar sus expectativas.
- Participar activamente en su desarrollo académico para acrecentar su capacidad de aprendizaje y evolucionar como profesional con autonomía.
- Asumir los principios y valores universitarios, y actuar en consecuencia.
- Emplear habilidades lingüístico-comunicativas del inglés.



- Evaluar el progreso, integración e incertidumbre de las ciencias, ante la creciente complejidad de las profesiones.

Particulares

- Incorporar estrategias para el análisis de datos e información, comprender su significado, procesarla y convertirla en conocimiento para elevar la eficiencia del proceso de producción industrial.
- Desarrollar la sensibilidad y el arte como base de la creatividad para el diseño de procesos, herramientas, equipos y maquinaria de la producción industrial de bienes.
- Aprender los modelos matemáticos, teorías y ciencias que explican el proceso de producción industrial de bienes
- Tomar decisiones y formular soluciones racionales, éticas y estéticas para el mejoramiento de procesos de producción industrial.
- Desarrollar su forma de expresarse, su creatividad, iniciativa y espíritu emprendedor hacia el desarrollo de proyectos de producción industrial de bienes.
- Aplicar las metodologías de diseño, automatización, optimización y administración de procesos de producción industrial de bienes y de herramienta, equipos y maquinaria para la intervención profesional en empresas manufactureras.
- Aplicar los métodos de diseño, fabricación, medición y programación; técnicas de dibujo, simulación y control; e instrumentos de cómputo, máquinas herramienta, materiales a mecanizar, empleados en la intervención profesional.
- Emplear las habilidades técnicas y tecnológicas para evolucionar en el desarrollo industrial, cuidando el medio ambiente y satisfaciendo las necesidades sociales.
- Desarrollar un juicio profesional basado en la responsabilidad, objetividad, credibilidad y la justicia para evaluar la creación o diseño de procesos de producción industrial de bienes y de maquinaria, equipos y herramientas, fortaleciendo el desarrollo industrial y satisfaciendo las necesidades de la sociedad.



Objetivos del núcleo de formación:

Núcleo básico: Promoverá en el alumno el aprendizaje de las bases contextuales, teóricas y filosóficas de sus estudios, la adquisición de una cultura universitaria en las ciencias y las humanidades, y el desarrollo de las capacidades intelectuales indispensables para la preparación y ejercicio profesional, o para diversas situaciones de la vida personal y social.

Objetivos del área curricular o disciplinaria:

Examinar la aplicación de las ciencias básicas y las matemáticas para la solución de problemas en los procesos de producción industrial de bienes; aprovechando y transformando los recursos naturales, materiales y energéticos de manera eficiente, con la menor producción de residuos y contaminantes.

Objetivos del área curricular o disciplinaria:

Examinar la aplicación de las ciencias básicas y las matemáticas para la solución de problemas en los procesos de producción industrial de bienes; aprovechando y transformando los recursos naturales, materiales y energéticos de manera eficiente, con la menor producción de residuos y contaminantes.

V. Objetivos de la unidad de aprendizaje.

Proponer sistemas mecánicos, utilizando el puerto serie de la PC y circuitos eléctricos ya diseñados.



VI. Contenidos de la unidad de aprendizaje

Unidad 1. Introducción a la lógica algorítmica.

Objetivo: Relacionar los conceptos sobre los cuales se sustenta la programación imperativa a través de algoritmos, para incentivar el pensamiento lógico y ordenado.

Contenidos:

- 1.1 Fases de la metodología de la programación
- 1.2 Análisis de algoritmos
- 1.3 Representación del Algoritmo
 - 1.3.1 Pseudocódigo
 - 1.3.2 Diagrama de flujo
 - 1.3.3 Prueba del algoritmo

Unidad 2. Introducción a la programación

Objetivo: Realizar algoritmos con los diferentes tipos de datos soportados por el lenguaje de programación, a través de los operadores aritméticos lógicos y relacionales, con la finalidad de comprobar la prioridad y los resultados obtenidos en las operaciones mediante la computadora.

Contenidos:

- 2.1. Formato de datos (tipos de datos)
- 2.2. Operadores
 - 2.2.1. Aritméticos
 - 2.2.2. Lógico
 - 2.2.3. Relacionales
 - 2.2.4. Prioridad en los operadores
- 2.3. Aplicaciones



Unidad 3. Programación estructurada selectiva

Objetivo: Aplicar las bases de la programación estructurada hacia la solución de problemas de ingeniería, utilizando las sentencias selectivas.

Contenidos:

- 3.1. Sentencias: if, if-else y compuesta
- 3.2. Sentencia switch y compuesta
- 3.3. Aplicaciones

Unidad 4. Programación estructurada repetitiva

Objetivo: Aplicar las bases de la programación estructurada hacia la solución de problemas de ingeniería, utilizando las sentencias repetitivas.

Contenidos:

- 4.1. Sentencia repetitiva for
- 4.2. Sentencia repetitiva while
- 4.3. Aplicaciones

Unidad 5. Arreglos unidimensionales

Objetivo: Analizar algoritmos de ingeniería, programando una solución a través de la creación y recorrido de arreglos unidimensionales

Contenidos:

- 5.1. Acceso
- 5.2. Recorrido
- 5.3. Operaciones
- 5.4. Aplicaciones

VII. Prácticas Resueltas

Ficha Técnica			
Práctica No.	1	Unidad No.	1
Nombre de Práctica	<i>El área de un círculo</i>		

Definición de Problema
Diseñar un algoritmo en forma de pseudocódigo y diagrama de flujo para obtener el área de un círculo, solicitando por teclado el valor del radio
Objetivo
Utilizar los operadores adecuados para transformar una operación en expresiones a ser evaluadas de acuerdo a prioridad de operadores.
Objetivos Específicos
Para obtener el área de un Círculo emplear Algorítmica Pseudocódigo Prueba de Escritorio Diagrama de Flujo Codificación en Matlab (ver anexo D)

Análisis de Problema		
En este apartado identifica las variables, genera el algoritmo de solución		
Algoritmo de Solución		
1. Inicio 2. Solicitar al usuario el valor del radio 3. Realizar la Operación de $A = \pi * (\text{radio} * \text{radio})$ 4- Mostrar el Resultado 5.Fin		
Datos de Entrada	Proceso	Salida
radio	$A = \pi * \text{radio}^2$	A
Pi		
A		

Resultados
En este apartado se realiza la codificación del algoritmo anterior
Pseudocódigo Solución
Algoritmo areaCirculo Definir radio, A como Real; Escribir "Obtener el Área de un Círculo";



Escribir "Ingrese el valor del radio";
 Leer radio;
 $A \leftarrow \pi * (\text{radio} * \text{radio});$
 Escribir "El área del Circulo es ", A;
 FinAlgoritmo

Prueba de Escritorio

En este apartado realiza la prueba de escritorio de al menos tres ejemplos con el algoritmo Pseudocódigo que desarrollo

Variables identificadas			Visualiza el usuario		
A	Pi	radio			
	3.14159				Obtener el Área de un Círculo Ingrese el valor del radio
		3.4			
36.3168					El área del Circulo es 36.3168
	3.14159				Obtener el Área de un Círculo Ingrese el valor del radio
		2.5			
19.634					El área del Circulo es 19.634
	3.14159				Obtener el Área de un Círculo Ingrese el valor del radio
		-2.1			
13.854					El área del Circulo es 13.854

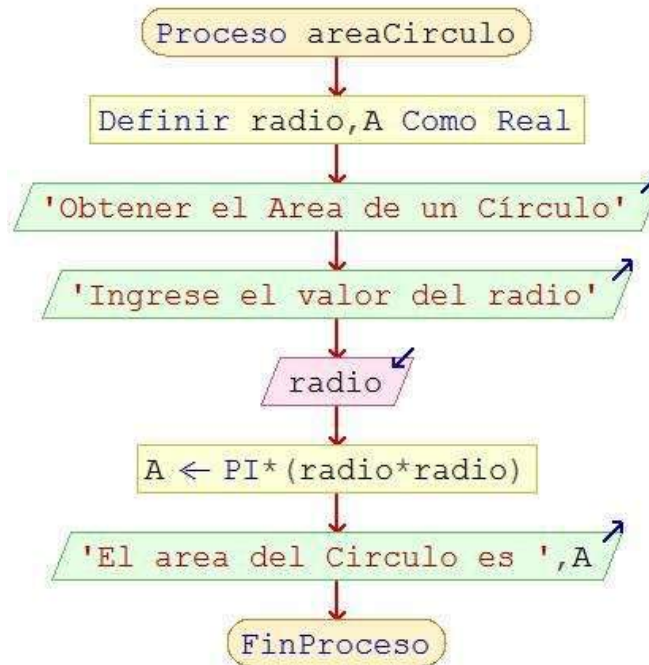
```

*** Ejecución Iniciada. ***
Obtener el Area de un Círculo
Ingrese el valor del radio
> -2.1
El area del Circulo es 13.8544236023
*** Ejecución Finalizada. ***
  
```



Diagrama de Flujo

En este apartado realiza el Diagrama de Flujo





Ficha Técnica			
Práctica No.	2	Unidad No.	1
Nombre de Práctica		La suma de dos números enteros	

Definición de Problema
Diseñar un algoritmo en forma de pseudocódigo y diagrama de flujo para obtener la suma de dos números enteros, solicitando por teclado los valores respectivos
Objetivo
Utilizar los operadores adecuados para transformar una operación en expresiones a ser evaluadas de acuerdo a prioridad de operadores.
Objetivos Específicos
Para obtener la suma de dos números emplear Algorítmica Pseudocódigo Prueba de Escritorio Diagrama de Flujo Codificación en Matlab (ver anexo E)

Análisis de Problema		
En este apartado identifica las variables, genera el algoritmo de solución		
Algoritmo de Solución		
1.- Inicio 2.- Solicitar al usuario el valor de A 3.- Solicitar al usuario el valor de B 4.- Sumar el valor de A + B y signarlo a R 5.- Mostrar R 5.-Fin		
Datos de Entrada	Proceso	Salida
A B R	R= A+B	R

Resultados
En este apartado se realiza la codificación del algoritmo anterior
Pseudocódigo Solución
Algoritmo suma2numeros Definir A, B, R como Real; Escribir "Obtener La suma de dos números"; Escribir "Ingrese el valor de A"; Leer A; Escribir "Ingrese el valor de B"; Leer B; R<- A + B; Escribir "La suma es ", R; FinAlgoritmo



Prueba de Escritorio

En este apartado realiza la prueba de escritorio de al menos tres ejemplos con el algoritmo Pseudocódigo que desarrollo

Variables identificadas						Visualiza el usuario
A	B	R				Obtener La suma de dos números
						Ingrese el valor de A
5						
						Ingrese el valor de B
	7					
		12				La suma es 12
						Obtener La suma de dos números
						Ingrese el valor de A
-8						
						Ingrese el valor de B
	-3					
		-11				La suma es -11
						Obtener La suma de dos números
						Ingrese el valor de A
-16						
						Ingrese el valor de B
	3					
		-13				La suma es -13

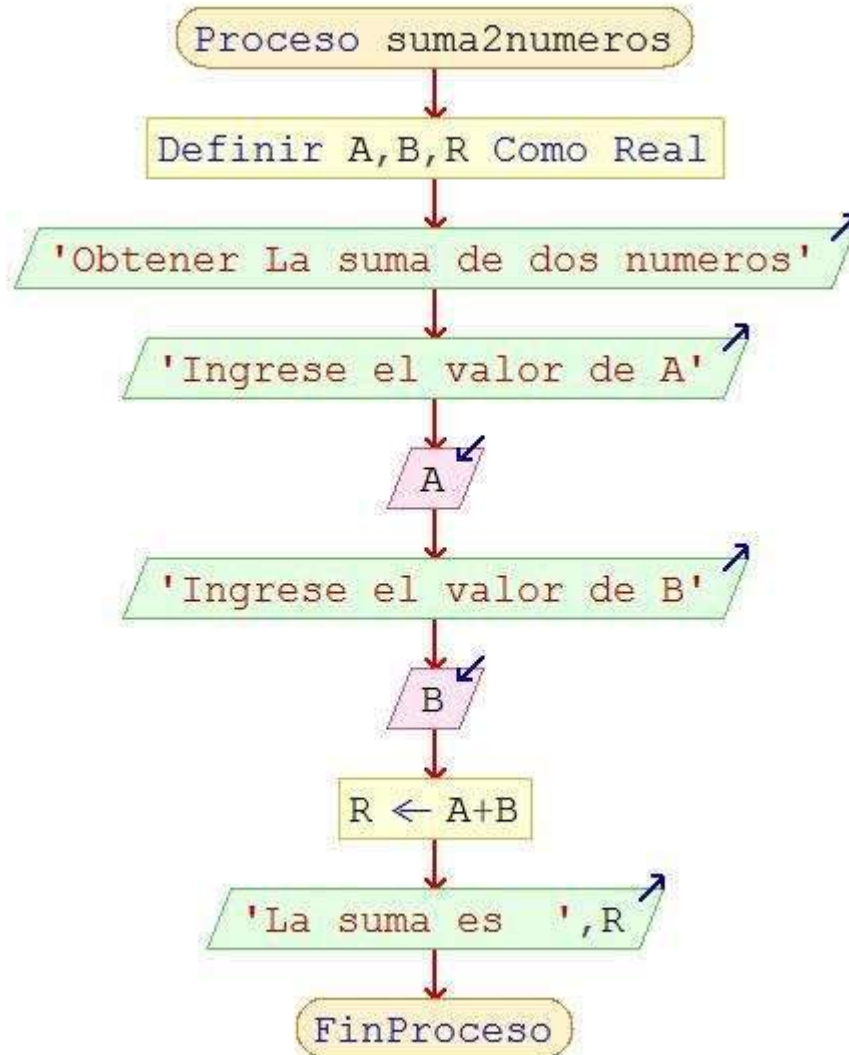
```

PSeInt - Ejecutando proceso SUMA2NUMEROS
*** Ejecución Iniciada. ***
Obtener La suma de dos numeros
Ingrese el valor de A
> -16
Ingrese el valor de B
> 3
La suma es -13
*** Ejecución Finalizada. ***
  
```



Diagrama de Flujo

En este apartado realiza el Diagrama de Flujo



Ficha Técnica			
Práctica No.	3	Unidad No.	2
Nombre de Práctica		Ejercicios Con Operadores	

Operadores Aritméticos

Operador	Descripción	Ejemplo
*	Multiplicación	(a*b)
/	División	(a/b)
+	Suma	(a+b)
-	Resta	(a-b)
MOD	División Modulo (resto)	a mod b
DIV	División Entera (cociente)	a div b

Prioridad de Operadores

()

^

* / mod div

+ -

Una operación se realiza de izquierda a derecha considerando prioridades, primero los paréntesis y si por alguna razón existiese una misma prioridad se considera la primera que se encuentra de izquierda a derecha.



Operadores Relacionales

Operador	Descripción	Ejemplo
<	Menor que	$a < b$
<=	Menor igual que	$a <= b$
>	Mayor que	$a > b$
>=	Mayor igual que	$a >= b$
==	Igual	$a == b$
<>	Diferente	$a <> b$

Operadores Lógicos

Operador	Descripción
No	Negación de una expresión
Y	Ambos lados de la expresión se cumple
O	Alguno de los lados de la expresión se cumple

Y Lógico		
V	V	Verdadero
V	F	Falso
F	V	Falso
F	F	Falso

O Lógico		
V	V	Verdadero
V	F	Verdadero
F	V	Verdadero
F	F	Falso

No Lógico	
V	Falso
F	Verdadero

Ficha Técnica

Práctica No.	4	Unidad No.	2
Nombre de Práctica	<i>El valor de x dada la función $f(x) = x^3 / 2(x^2)$</i>		

Definición de Problema

Diseñar la solución para la función $f(x) = x^3 / 2(x^2)$ solicitando por teclado el valor de x

Objetivo

Utilizar los operadores adecuados para transformar una operación en expresiones a ser evaluadas de acuerdo a prioridad de operadores

Objetivos Específicos

Para obtener el valor de x dada la función $f(x) = x^3 / 2(x^2)$

Algorítmica

Pseudocódigo

Prueba de Escritorio

Diagrama de Flujo

Codificación en Matlab (ver anexo B)

Análisis de Problema

En este apartado identifica las variables, genera el algoritmo de solución

Algoritmo de Solución

- 1.- Inicio
- 2.- Solicitar al usuario el valor de x
- 3.- Realizar la operación $fx = x^3 / 2(x^2)$
- 4.- Mostrar fx
- 5.-Fin

Datos de Entrada	Proceso	Salida
x	$fx = x^3 / 2(x^2)$	fx

Resultados

En este apartado se realiza la codificación del algoritmo anterior

Pseudocódigo Solución

Algoritmo fxX

Definir fx, x como Real;

Escribir "Obtener La función $x^3 / 2(x^2)$ ";

Escribir "Ingrese el valor x";

Leer x;

$fx \leftarrow (x * x * x) / (2 * (x * x))$;

Escribir "el valor de fx es ", fx;

FinAlgoritmo



Prueba de Escritorio

En este apartado realiza la prueba de escritorio de al menos tres ejemplos con el algoritmo Pseudocódigo que desarrollo

Variables identificadas							Visualiza el usuario
x	Fx						Obtener La función $x^3 / 2(x^2)$ Ingrese el valor x
3	1.5						el valor de fx es 1.5
6	3						Obtener La función $x^3 / 2(x^2)$ Ingrese el valor x
-4.8	-2.4						el valor de fx es 2.4

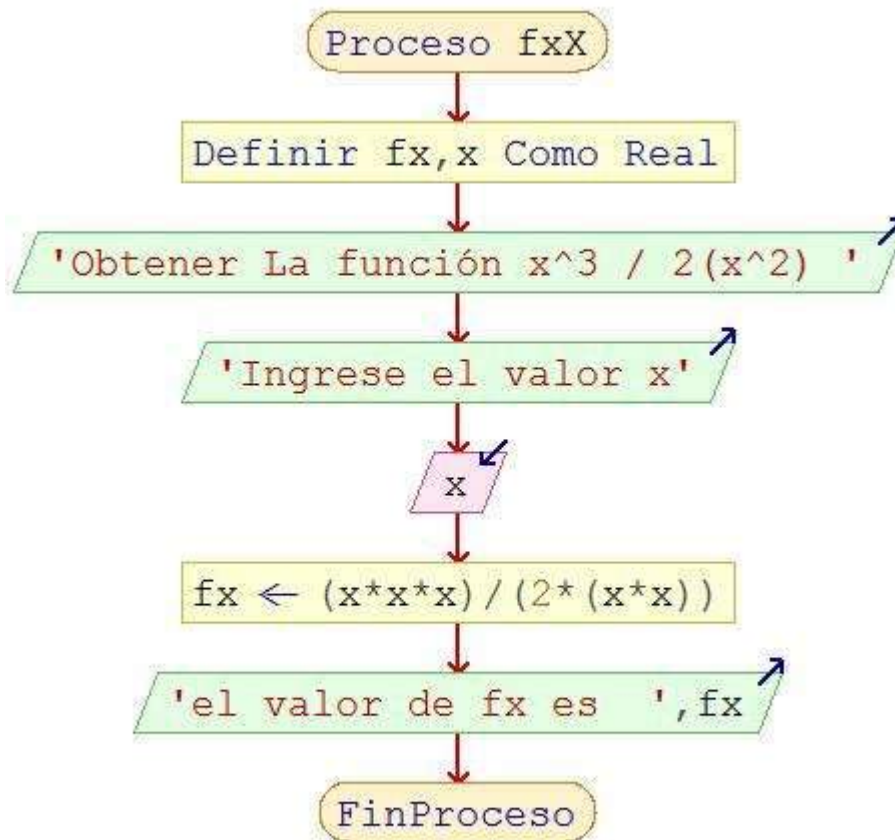
```

PSeInt - Ejecutando proceso FXX
*** Ejecución Iniciada. ***
Obtener La función  $x^3 / 2(x^2)$ 
Ingrese el valor x
> -4.8
el valor de fx es -2.4
*** Ejecución Finalizada. ***
  
```



Diagrama de Flujo

En este apartado realiza el Diagrama de Flujo





Codificación a Matlab

En este apartado realiza la Codificación a Código Matlab

```
disp('Obtener la función x^3 / 2(x^2)');  
x= input('Ingrese el valor x ');  
fx=(x*x*x) / (2*(x*x));  
fprintf('el valor de fx es %f \n', fx);
```

```
Command Window  
>> funcionFx  
Obtener la función x^3 / 2(x^2)  
Ingrese el valor x 3  
el valor de fx es 1.500000  
>> funcionFx  
Obtener la función x^3 / 2(x^2)  
Ingrese el valor x 6  
el valor de fx es 3.000000  
>> funcionFx  
Obtener la función x^3 / 2(x^2)  
Ingrese el valor x -4.8  
el valor de fx es -2.400000  
fx >>
```

El usuario ingresa el valor x e internamente se realiza la operación de la función para solo mostrar el resultado

Ficha Técnica			
Práctica No.	5	Unidad No.	3
Nombre de Práctica	<i>El promedio de 5 calificaciones y saber si exenta o no con mínimo 8.0</i>		

Definición de Problema
Diseñar la solución para obtener el promedio de 5 calificaciones solicitadas por teclado y definir si el alumno exenta o no
Objetivo
Emplear la sentencia condicional respectiva para dar solución al problema (if-else)
Objetivos Específicos
Para obtener el promedio de 5 calificaciones y saber si exenta o no con mínimo 80
Algorítmica
Pseudocódigo
Prueba de Escritorio
Diagrama de Flujo
Codificación en Matlab

Análisis de Problema		
En este apartado identifica las variables, genera el algoritmo de solución		
Algoritmo de Solución		
1.- Inicio		
2.- Solicitar al usuario el valor de las 5 calificaciones de una en una		
3.- Realizar la sumatoria y		
4.- Obtener el promedio dividiendo entre 5		
5.- Si el promedio es Mayor o igual a 8.0 exenta de no ser así no exenta		
6.-Fin		
Datos de Entrada	Proceso	Salida
C1		
C2		
C3		
C4		
C5		
	$P=(C1+C2+C3+C4+C5)/5$	
	Si $P \geq 8.0$	
	R=exenta	
	Sino R=NoExenta	
		R

Resultados

En este apartado se realiza la codificación del algoritmo anterior

Pseudocódigo Solución

Algoritmo exenta80

```

    Definir c1,c2,c3,c4,c5,p Como Real;
    Definir r Como Carácter;
    Escribir 'Obtener el promedio de 5 calificaciones ';
    Escribir 'Ingrese la calificación c1 ';
    Leer c1;
    Escribir 'Ingrese la calificación c2 ';
    Leer c2;
    Escribir 'Ingrese la calificación c3 ';
    Leer c3;
    Escribir 'Ingrese la calificación c4 ';
    Leer c4;
    Escribir 'Ingrese la calificación c5 ';
    Leer c5;
    p <- (c1+c2+c3+c4+c5)/5;
    Si (p>=8.0) Entonces
        r <- ' Si Exenta';
    Sino
        r <- ' No Exenta';
    FinSi
    Escribir 'El alumno con promedio ',p,r;
FinAlgoritmo
    
```

Prueba de Escritorio

En este apartado realiza la prueba de escritorio de al menos tres ejemplos con el algoritmo Pseudocódigo que desarrollo

Variables identificadas							Visualiza el usuario
c1	c2	c3	c4	c5	p	r	
							Obtener el promedio de 5 calificaciones
							Ingrese la calificación c1
7.5							
							Ingrese la calificación c2
	6.8						



						'Ingrese la calificación c3
		7.9				
						'Ingrese la calificación c4
			8.5			
						'Ingrese la calificación c5
				9.3		
					$p < - (c1+c2+c3+c4+c5)/5$	
						Si exenta
						'El alumno con promedio 8 Si exenta
						Obtener el promedio de 5 calificaciones
						Ingrese la calificación c1
5.6						
						Ingrese la calificación c2
		7.8				
						'Ingrese la calificación c3
			6.7			
						'Ingrese la calificación c4
				4.6		
						'Ingrese la calificación c5
					9.7	
					$p < - (c1+c2+c3+c4+c5)/5$	
						No exenta
						'El alumno con promedio 6.88 No exenta
						Obtener el promedio



						de 5 calificaciones
						Ingrese la calificación c1
8.1						
						Ingrese la calificación c2
	9.4					
						'Ingrese la calificación c3
		8.9				'Ingrese la calificación c4
			8.5			'Ingrese la calificación c5
				9.8		
					$p \leq (c1+c2+c3+c4+c5)/5$	
						Si exenta
						'El alumno con promedio 8.94 Si exenta



```
*** Ejecución Iniciada. ***
Obtener el promedio de 5 calificaciones
Ingrese la calificación c1
> 5.6
Ingrese la calificación c2
> 7.8
Ingrese la calificación c3
> 6.7
Ingrese la calificación c4
> 4.6
Ingrese la calificación c5
> 9.7
El alumno con promedio 6.88 No Exenta
*** Ejecución Finalizada. ***
```

No cerrar esta ventana Siempre visible Reiniciar

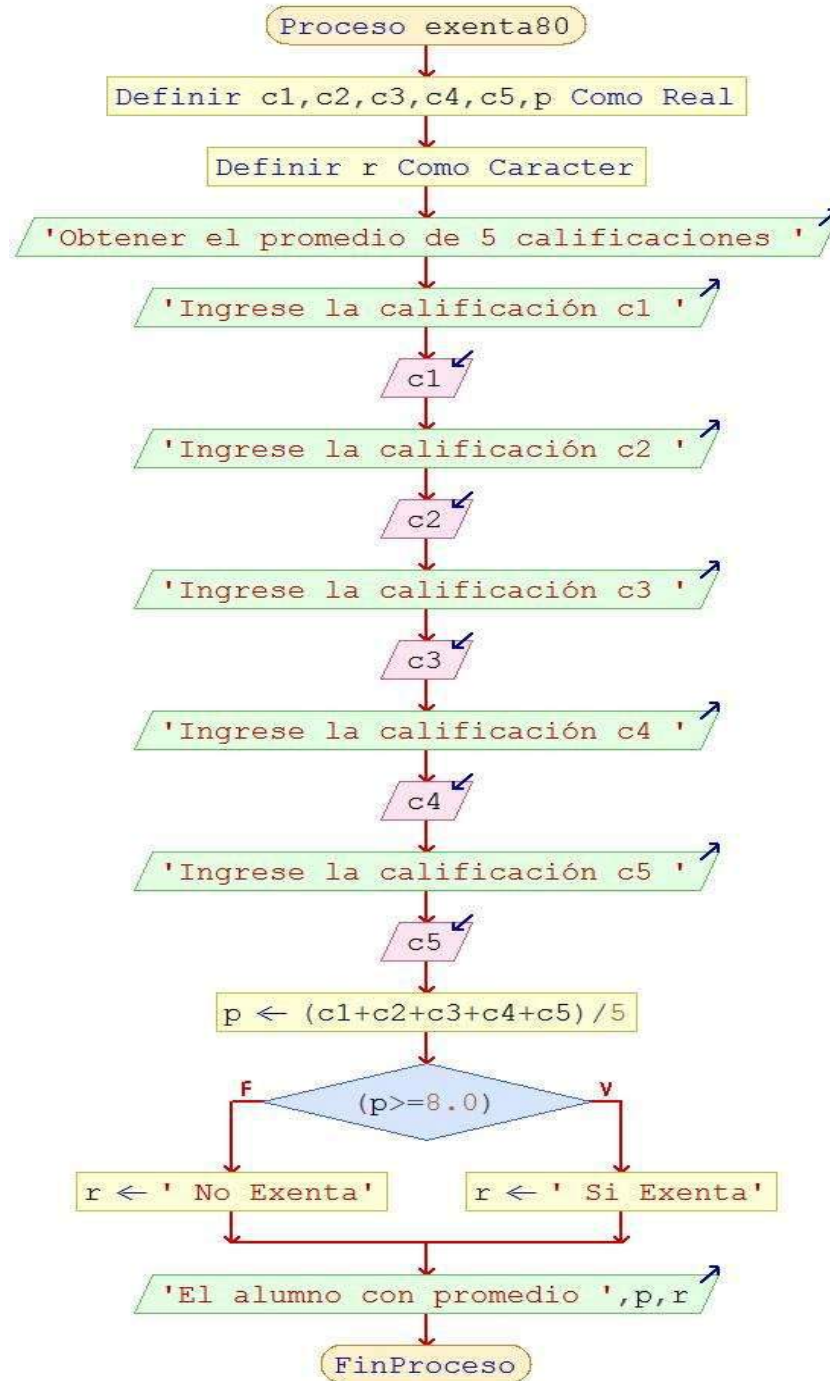
```
*** Ejecución Iniciada. ***
Obtener el promedio de 5 calificaciones
Ingrese la calificación c1
> 7.5
Ingrese la calificación c2
> 6.8
Ingrese la calificación c3
> 7.9
Ingrese la calificación c4
> 8.5
Ingrese la calificación c5
> 9.3
El alumno con promedio 8 Si Exenta
*** Ejecución Finalizada. ***
```

No cerrar esta ventana Siempre visible Reiniciar



Diagrama de Flujo

En este apartado realiza el Diagrama de Flujo





Codificación a Matlab

En este apartado realiza la Codificación a Código Matlab

```
disp('Obtener el promedio de 5 calificaciones ');  
c1=input('Ingrese la calificación c1 ');  
c2=input('Ingrese la calificación c2 ');  
c3=input('Ingrese la calificación c3 ');  
c4=input('Ingrese la calificación c4 ');  
c5=input('Ingrese la calificación c5 ');  
p = (c1+c2+c3+c4+c5)/5;  
if (p>=8.0)  
    r = ' Si Exenta';  
else  
    r = ' No Exenta';  
end  
fprintf('El alumno con promedio %f , %s \n',p,r);
```

```
>> promedio  
Obtener el promedio de 5 calificaciones  
Ingrese la calificación c1 7.5  
Ingrese la calificación c2 6.8  
Ingrese la calificación c3 7.9  
Ingrese la calificación c4 8.5  
Ingrese la calificación c5 9.3  
El alumno con promedio 8.000000 , Si Exenta  
>> promedio  
Obtener el promedio de 5 calificaciones  
Ingrese la calificación c1 5.6  
Ingrese la calificación c2 7.8  
Ingrese la calificación c3 6.7  
Ingrese la calificación c4 4.6  
Ingrese la calificación c5 9.7  
El alumno con promedio 6.880000 , No Exenta  
>> promedio  
Obtener el promedio de 5 calificaciones  
Ingrese la calificación c1 8.1  
Ingrese la calificación c2 9.4  
Ingrese la calificación c3 8.9  
Ingrese la calificación c4 8.5  
Ingrese la calificación c5 9.8  
El alumno con promedio 8.940000 , Si Exenta  
fx >>
```

El usuario ingresa cada una de las calificaciones y el programa da el promedio indicando si exenta o no.

Ficha Técnica			
Práctica No.	6	Unidad No.	3
Nombre de Práctica		<i>Transformación de temperaturas de acuerdo a la elección del usuario</i>	

Definición de Problema
Diseñar la solución para permitirle al usuario seleccionar entre diferentes acciones a realizar en la transformación de temperaturaas
Objetivo
Emplear la sentencia de selección según el caso que elija el usuario (switch)
Objetivos Específicos
Para obtener transformación de temperaturas de acuerdo a la elección de usuario
Algorítmica
Pseudocódigo
Prueba de Escritorio
Diagrama de Flujo
Codificación en Matlab

Análisis de Problema		
En este apartado identifica las variables, genera el algoritmo de solución		
Algoritmo de Solución		
1.- Inicio 2.- Mostrar las posibilidades de transformación de temperatura "Transformación de temperaturas"; "1 De Fahrenheit A Celsius"; "2 De Celsius A Fahrenheit"; "3 De Rankin A Kelvin"; "4 De Kelvin A Rankin"; 3.- Solicitar que transformación desea 4.- Según la elección del usuario Para cada transformación Solicitar temperatura a transformar Realizar la transformación Mostrar el resultado 7.-Fin		
Datos de Entrada	Proceso	Salida
num	Según la elección de usuario	r
r	Realizar la transformación	
opc		



Resultados

En este apartado se realiza la codificación del algoritmo anterior

Pseudocódigo Solución

Algoritmo temperaturas

Definir num, r como Real;

Definir opc como Entero;

Escribir "Transformación de temperaturas";

Escribir "1 De Fahrenheit A Celsius";

Escribir "2 De Celsius A Fahrenheit";

Escribir "3 De Rankin A Kelvin";

Escribir "4 De Kelvin A Rankin";

Escribir "5 Salida";

Escribir "Indique la equivalencia deseada ";

Leer opc;

Segun opc Hacer

1:

Escribir "Transformación de Fahrenheit a Celsius";

Escribir "Indique la temperatura en grados Fahrenheit ";

Leer num;

$r \leftarrow -(\text{num} * (-17.22))$;

Escribir "El valor de la temperatura es", r, "C";

2:

Escribir "Transformación de centígrados a Fahrenheit ";

Escribir "Indique la temperatura en grados centígrados";

Leer num;

$r \leftarrow -(\text{num} * 33.8)$;

Escribir "El valor de la temperatura es", r, "F";

3:

Escribir "Transformación de grados kelvin a rankin";

Escribir "Indique la temperatura en grados kelvin";

Leer num;

$r \leftarrow -(\text{num} * 1.8)$;

Escribir "El valor de la temperatura es", r, "K";

4:

Escribir "Transformación de grados rankin a kelvin";

Escribir "Indique la temperatura en grados rankin";

Leer num;

$r \leftarrow -(\text{num} * 0.5555)$;

Escribir "El valor de la temperatura es", r, "R";

5:

Escribir "Hasta Pronto";

De Otro Modo:



Escribir "El valor no está identificado";

FinSegun
FinAlgoritmo

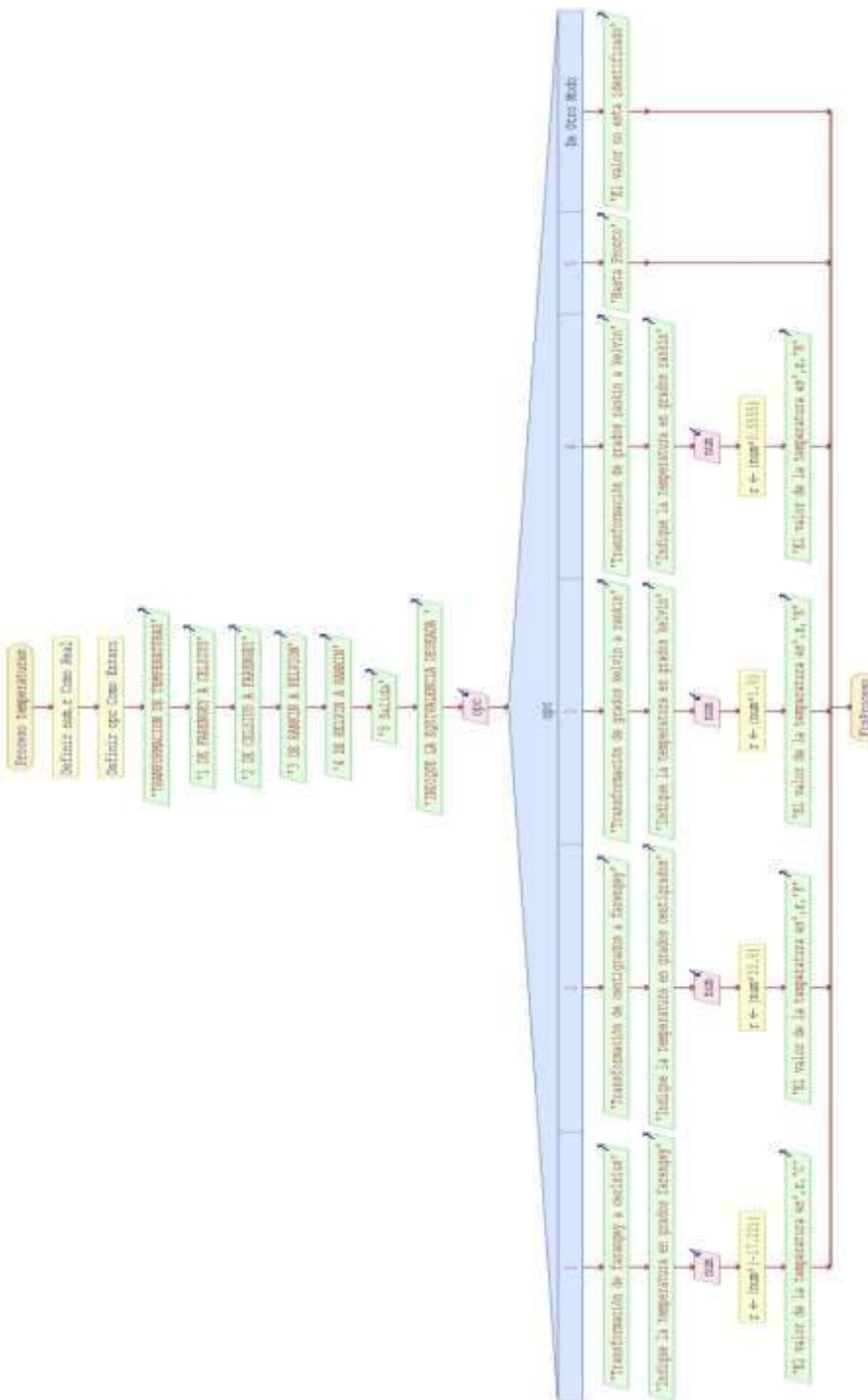
Prueba de Escritorio			
En este apartado realiza la prueba de escritorio de al menos tres ejemplos con el algoritmo Pseudocódigo que desarrollo			
Variables identificadas			Visualiza el usuario
opc	num	r	
			Transformación de temperaturas 1 De Fahrenheit A Celsius 2 De Celsius A Fahrenheit 3 De Rankin A Kelvin 4 De Kelvin A Rankin 5 Salida Indique la equivalencia deseada
5			Hasta Pronto
			Transformación de temperaturas 1 De Fahrenheit A Celsius 2 De Celsius A Fahrenheit 3 De Rankin A Kelvin 4 De Kelvin A Rankin 5 Salida Indique la equivalencia deseada
2			Transformación de centígrados a Fahrenheit Indique la temperatura en grados centígrados
	6.77		
		228.83	El valor de la temperatura es 228.83 F
			Transformación de temperaturas 1 De Fahrenheit A Celsius 2 De Celsius A Fahrenheit 3 De Rankin A Kelvin 4 De Kelvin A Rankin 5 Salida Indique la equivalencia deseada
4			Transformación de grados rankin a kelvin Indique la temperatura en grados rankin



1233.56		
		El valor de la temperatura es 685.24 R

Diagrama de Flujo

En este apartado realiza el Diagrama de Flujo





Codificación a Matlab

En este apartado realiza la Codificación a Código Matlab

```
disp('Transformación de temperaturas');
disp('1 De Fahrenheit A Celsius');
disp('2 De Celsius A Fahrenheit');
disp('3 De Rankin A Kelvin');
disp('4 De Kelvin A Rankin');
disp('5 Salida');
opc = input('Indique la equivalencia deseada ');
switch opc
    case 1
        disp('Transformación de Fahrenheit a Celsius');
        num=input('Indique la temperatura en grados Fahrenheit ');
        r=(num*(-17.22));
        fprintf('El valor de la temperatura es %3.2f C \n',r);
        break;
    case 2
        disp('Transformación de centígrados a Fahrenheit ');
        num = input('Indique la temperatura en grados centígrados ');
        r=(num*33.8);
        fprintf('El valor de la temperatura es %3.2f F \n',r);
        break;
    case 3
        disp('Transformación de grados kelvin a rankin');
        num = input('Indique la temperatura en grados kelvin ');
        r=(num*1.8);
        fprintf('El valor de la temperatura es %3.2f K \n',r);
        break;
    case 4
        disp('Transformación de grados rankin a kelvin');
        num = input('Indique la temperatura en grados rankin ');
        r=(num*0.5555);
        fprintf('El valor de la temperatura es %3.2f R \n',r);
        break;
    case 5
        disp(' Hasta Pronto');
        break;
    otherwise
        disp(' El valor no está identificado');
        break;
end
```



```
Command Window
>> temperaturas
Transformación de temperaturas
1 De Fahrenheit A Celsius
2 De Celsius A Fahrenheit
3 De Rankin A Kelvin
4 De Kelvin A Rankin
5 Salida
Indique la equivalencia deseada 5
Hasta Pronto
>> temperaturas
Transformación de temperaturas
1 De Fahrenheit A Celsius
2 De Celsius A Fahrenheit
3 De Rankin A Kelvin
4 De Kelvin A Rankin
5 Salida
Indique la equivalencia deseada 2
Transformación de centígrados a Fahrenheit
Indique la temperatura en grados centígrados 6.77
El valor de la temperatura es 228.83 F
>> temperaturas
Transformación de temperaturas
1 De Fahrenheit A Celsius
2 De Celsius A Fahrenheit
3 De Rankin A Kelvin
4 De Kelvin A Rankin
5 Salida
Indique la equivalencia deseada 4
Transformación de grados rankin a kelvin
Indique la temperatura en grados rankin 1233.56
El valor de la temperatura es 685.24 R
>> |
```

El usuario visualiza un menú de las actividades que puede realizar, ingresa su opción y el programa muestra a transformación de la temperatura indicada.

Ficha Técnica			
Práctica No.	7	Unidad No.	3
Nombre de Práctica		Operaciones Básicas	

Definición de Problema
Diseñar la solución para permitir al usuario realizar alguna operación básica, según la seleccione y si selecciona división verificar que el denominador sea menor al numerador para realizar la operación si alguno de ellos es cero indicar error
Objetivo
Hacer uso de sentencias de selección y condicional anidadas (switch y if-else)
Objetivos Específicos
Para obtener transformación de temperaturas de acuerdo a la elección de usuario Algorítmica Pseudocódigo Prueba de Escritorio Diagrama de Flujo Codificación en Matlab

Análisis de Problema		
En este apartado identifica las variables, genera el algoritmo de solución		
Algoritmo de Solución		
<ol style="list-style-type: none"> 1. Inicio 2. Mostrar al usuario las operaciones que puede realizar 3. Suma Resta Multiplicación y División 4. Solicitar la operación 5. Solicitar valor A 6. Solicitar Valor B 7. Realizar la operación según lo que selecciono el usuario 8. Si es división $A=0$ o $B=0$ mostrar mensaje de no se puede hacer la operación 9. Sino si $A>B$ dividir B en A 10. Sino A en B y mostrar el resultado 11. Fin 		
Datos de Entrada	Proceso	Salida
Opc, A B	Operación seleccionada	R

Prueba de Escritorio							
En este apartado realiza la prueba de escritorio de al menos tres ejemplos con el algoritmo que desarrollo							
Variables identificadas				Visualiza el usuario			
A	B	opc	c				Seleccione Operaciones Básicas 1 Suma 2 Resta 3 Multiplicación 4 División 5 Salir
		5					De acuerdo Hasta Pronto
							Seleccione Operaciones Básicas 1 Suma 2 Resta 3 Multiplicación 4 División 5 Salir
		3					Ingresa primer valor
13							Ingresa segundo valor
	6.9		$C=13*6.9$				Multiplicación es 89.7
							Seleccione Operaciones Básicas 1 Suma 2 Resta 3 Multiplicación 4 División 5 Salir
		4					Ingresa primer valor
34							Ingresa segundo valor
	7		$C=34/7$				División es 4.857142857



Resultados

En este apartado se realiza la codificación del algoritmo anterior

Pseudocódigo Solución

Algoritmo OperacionesB

```
Definir a,b,c Como Real;
Definir opc Como Entero;
Escribir 'Seleccione Operaciones Básicas';
Escribir '1 Suma';
Escribir '2 Resta';
Escribir '3 Multiplicación';
Escribir '4 División';
Escribir '5 Salir';
Leer opc;
Si opc==5 Entonces
    Escribir 'De acuerdo ';
Sino
    Escribir 'Ingresa Primer Valor';
    Leer a;
    Escribir 'Ingresa Segundo Valor';
    Leer b;
FinSi
Segun opc Hacer
1:
    Escribir 'La Suma';
    c <- a+b;
    Escribir 'es ',c;
2:
    Escribir 'La Resta';
    c <- a-b;
    Escribir 'es ',c;
3:
    Escribir 'La Multiplicación';
    c <- a*b;
    Escribir 'es ',c;
4:
    Escribir 'La División';
    Si a=0 O b=0 Entonces
        Escribir 'Operacion no Valida';
    Sino
        Si a>b Entonces
            c <- a/b;
        Sino
            c <- b/a;
    FinSi
```



```
        Escribir 'es ',c;
    FinSi
5:      Escribir 'Hasta Pronto';
De Otro Modo:
        Escribir 'Opcion no valida';
FinSegun
FinAlgoritmo
```

```
PSeInt - Ejecutando proceso OPERACIONESB
*** Ejecución Iniciada. ***
Seleccione Operaciones Básicas
1 Suma
2 Resta
3 Multiplicación
4 División
5 Salir
> 5
De acuerdo
Hasta Pronto
*** Ejecución Finalizada. ***

 No cerrar esta ventana  Siempre visible Reiniciar
```

```
PSeInt - Ejecutando proceso OPERACIONESB
*** Ejecución Iniciada. ***
Seleccione Operaciones Básicas
1 Suma
2 Resta
3 Multiplicación
4 División
5 Salir
> 4
Ingresa Primer Valor
> 34
Ingresa Segundo Valor
> 7
La División
es 4.8571428571
*** Ejecución Finalizada. ***

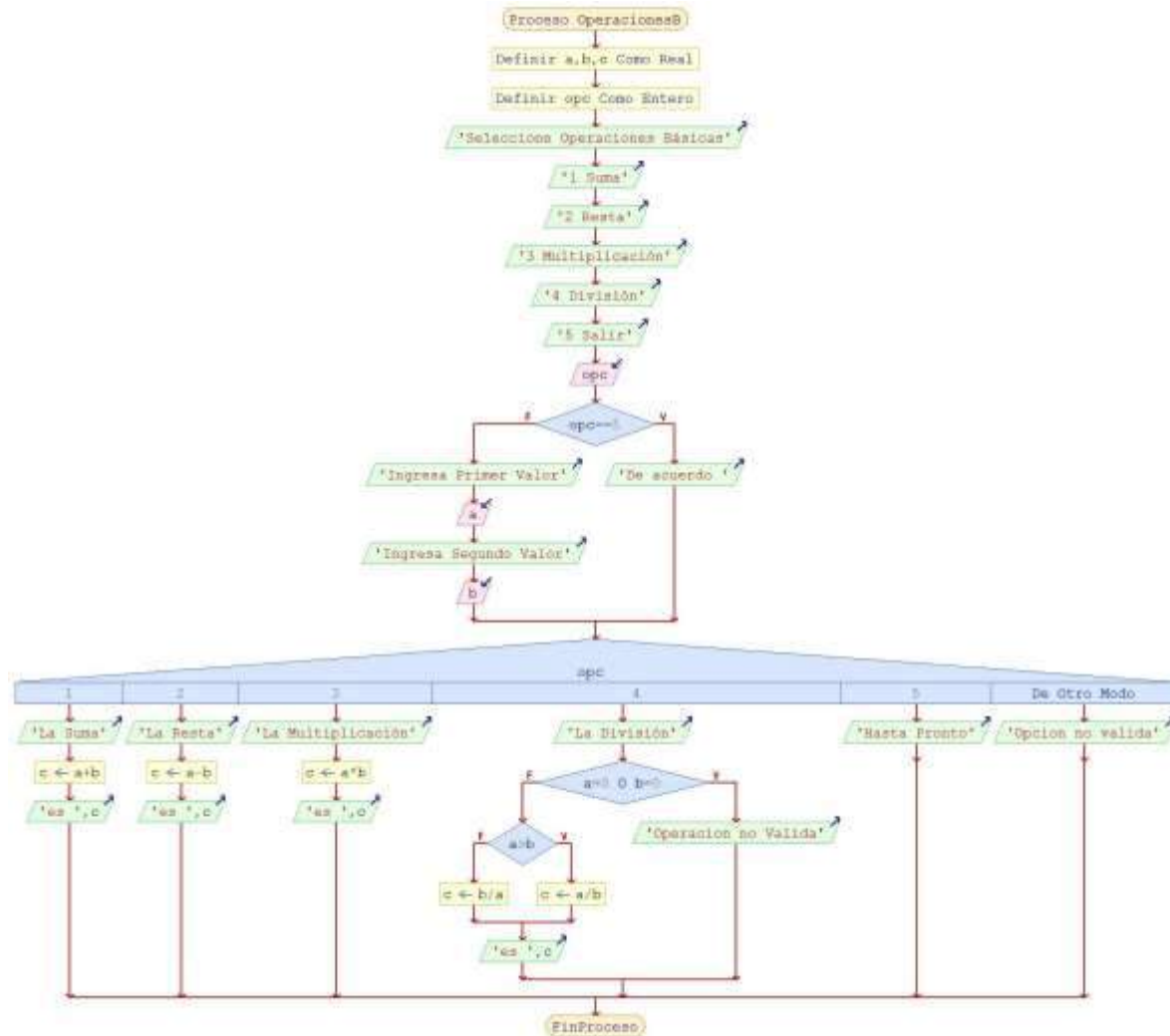
 No cerrar esta ventana  Siempre visible Reiniciar
```



Resultados

En este apartado realiza el diagrama de flujo correspondiente

Diagrama de Flujo





Resultados

En este apartado realiza la codificación del algoritmo planteado a Matlab

Codificación

```
disp('Seleccione Operaciones Básicas');
disp('1 Suma');
disp('2 Resta');
disp('3 Multiplicación');
disp('4 División');
disp('5 Salir');
opc = input('Ingrese su opción ');
if opc==5
    disp('De acuerdo ');
else
    a = input('Ingresa Primer Valor ');
    b = input('Ingresa Segundo Valor ');
end
switch opc
    case 1
        c = a+b;
        fprintf('La Suma es %f \n', c);
        break;
    case 2
        c = a-b;
        fprintf('La Resta es %f \n', c);
        break;
    case 3
        c = a*b;
        fprintf('La División es %f \n', c);
        break;
    case 4
        if a==0 || b==0
            disp('La División está en Operación no Valida');
        else
            if a>b
                c = a/b;
            else
                c = b/a;
            end
            fprintf('La División es %f \n', c);
        end
        break;
    case 5
```



```
disp('Hasta Pronto');  
break;  
otherwise  
disp('Opcion no valida');  
break;  
end
```

Resultados

En este apartado realiza la explicación de la ejecución en Matlab

Pantallas Resultado Explicadas

```
Command Window  
>> operaciones  
Seleccione Operaciones Básicas  
1 Suma  
2 Resta  
3 Multiplicación  
4 División  
5 Salir  
Ingrese su opción 3  
De acuerdo  
Hasta Pronto  
>> operaciones  
Seleccione Operaciones Básicas  
1 Suma  
2 Resta  
3 Multiplicación  
4 División  
5 Salir  
Ingrese su opción 3  
Ingrese Primer Valor 13  
Ingrese Segundo Valor 5.5  
La División es 89.700000  
>> operaciones  
Seleccione Operaciones Básicas  
1 Suma  
2 Resta  
3 Multiplicación  
4 División  
5 Salir  
Ingrese su opción 4  
Ingrese Primer Valor 34  
Ingrese Segundo Valor 7  
La División es 4.857143  
/s >> |
```

El usuario visualiza el menú de opciones e ingresa la operación a realizar así como los operadores y el programa muestra el resultado

Ficha Técnica			
Práctica No.	8	Unidad No.	4
Nombre de Práctica		Suma de valores mientras el resultado sea menor a 150	

Definición de Problema
Diseñar la solución para realizar una sumatoria mientras el resultado sea menor de 150, solicitando los valores por teclado
Objetivo
Hacer uso del ciclo de repetición mientras (while)
Objetivos Específicos
Para obtener transformación de temperaturas de acuerdo a la elección de usuario
Algorítmica
Pseudocódigo
Prueba de Escritorio
Diagrama de Flujo
Codificación en Matlab

Análisis de Problema		
En este apartado identifica las variables, genera el algoritmo de solución		
Algoritmo de Solución		
<ol style="list-style-type: none"> 1. Inicio 2. Mientras la suma sea menor a 150 seguir 3. Solicitar un número y sumarlo 4. Mostrar el resultado 5. Fin 		
Datos de Entrada	Proceso	Salida
Suma		
numero	Suma = suma + numero	suma

Prueba de Escritorio						
En este apartado realiza la prueba de escritorio de al menos tres ejemplos con el algoritmo que desarrollo						
Variables identificadas			Visualiza el usuario			
Suma	numero					
						Suma valores mientras el resultado sea menor a 150
						Ingrese valor a sumar
	12					
12						Suma =12
						Ingrese valor a sumar
	45					



57		Suma = 57
		Ingrese valor a sumar
	64	
121		Suma = 121
		Ingrese valor a sumar
	56	
177		Suma =177
		Fin Proceso
		Suma valores mientras el resultado sea menor a 150
		Ingrese valor a sumar
	11	
11		Suma =11
		Ingrese valor a sumar
	143	
154		Suma = 154
		Fin Proceso
		Suma valores mientras el resultado sea menor a 150
		Ingrese valor a sumar
	200	
200		Suma =200
		Fin Proceso

Resultados

En este apartado se realiza la codificación del algoritmo anterior

Pseudocódigo Solución

Algoritmo sumaMientras

Definir suma Como Real;

Definir numero Como Real;

Escribir 'Suma valores mientras el resultado sea menor a 150';

suma <- 0;

Mientras suma<=150 Hacer

 Escribir 'Ingrese valor a sumar ';

 Leer numero;

 suma <- suma+numero;

 Escribir 'suma = ',suma;

FinMientras

Escribir 'Fin proceso';

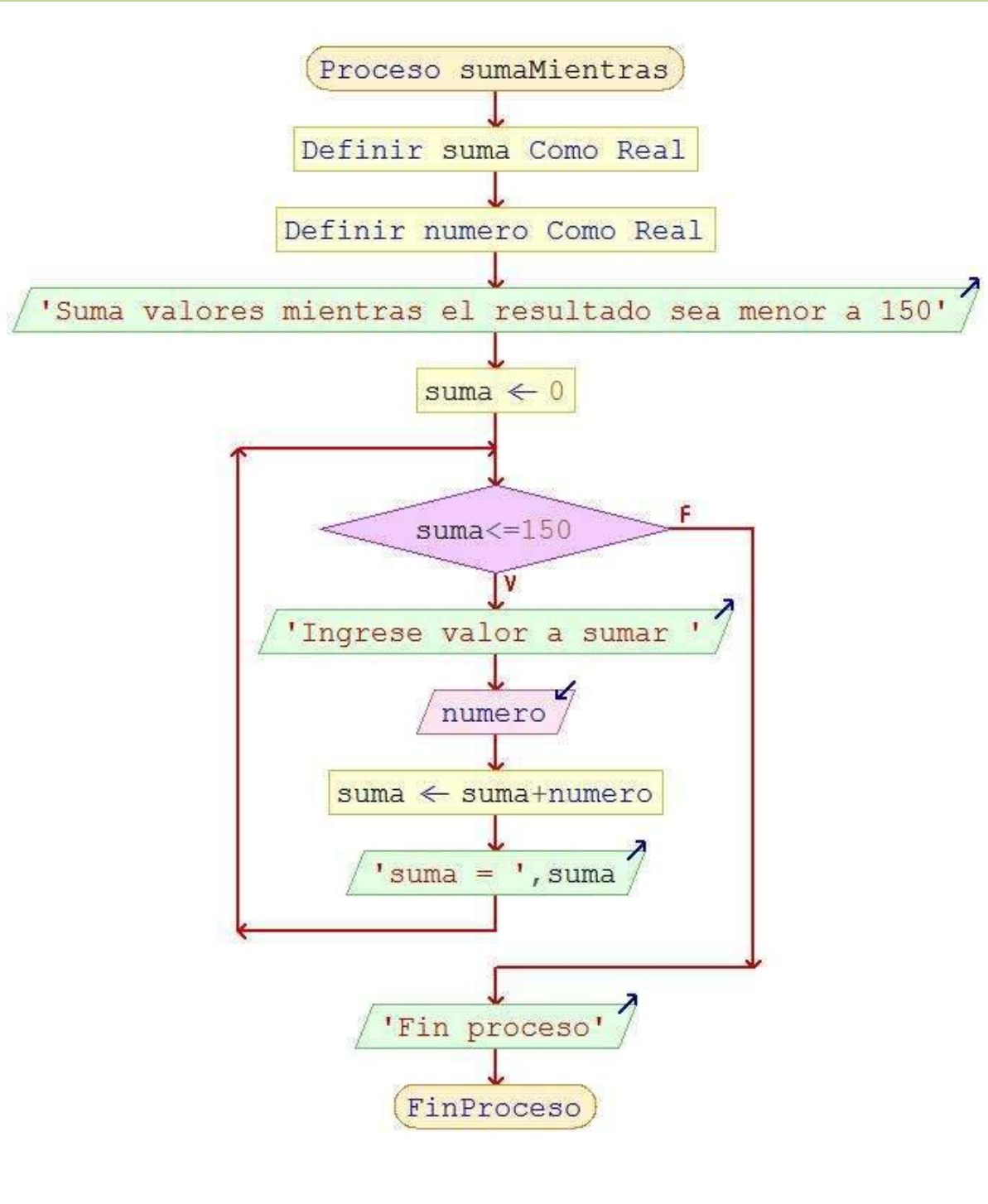
FinAlgoritmo



Resultados

En este apartado realiza el diagrama de flujo correspondiente

Diagrama de Flujo





```
Suma valores mientras el resultado sea menor a 150
Ingrese valor a sumar
> 12
suma = 12
Ingrese valor a sumar
> 45
suma = 57
Ingrese valor a sumar
> 64
suma = 121
Ingrese valor a sumar
> 56
suma = 177
Fin proceso
*** Ejecución Finalizada. ***
```

Resultados

En este apartado realiza la codificación del algoritmo planteado a Matlab

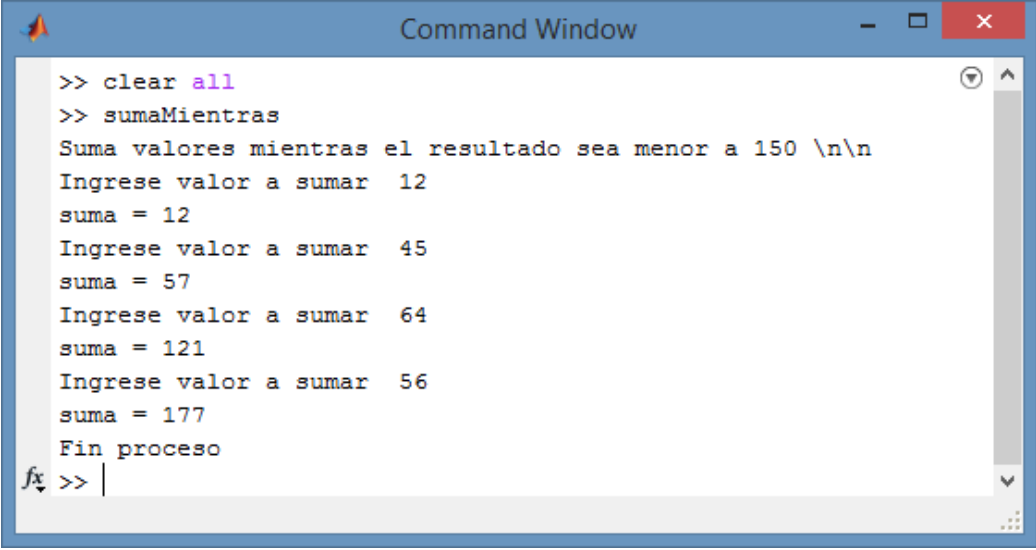
Codificación

```
disp('Suma valores mientras el resultado sea menor a 150 \n\n');
suma = 0;
while suma<=150
    numero=input('Ingrese valor a sumar ');
    suma = suma+numero;
    fprintf('suma = %d \n',suma);
end
disp('Fin proceso');
```

Resultados

En este apartado realiza la explicación de la ejecución en Matlab

Pantallas Resultado Explicadas



```
>> clear all
>> sumaMientras
Suma valores mientras el resultado sea menor a 150 \n\n
Ingrese valor a sumar 12
suma = 12
Ingrese valor a sumar 45
suma = 57
Ingrese valor a sumar 64
suma = 121
Ingrese valor a sumar 56
suma = 177
Fin proceso
fx >> |
```

El usuario ingresa números, mientras la sumatoria de esos números sea menor de 150 seguirá ingresando números para sumar, de no ser así simplemente termina el programa



Ficha Técnica			
Práctica No.	9	Unidad No.	4
Nombre de Práctica		Tabla de Multiplicar que el usuario indique	

Definición de Problema
Diseñar la solución para permitir que el usuario visualice la tabla de multiplicar deseada solicitando el numero por teclado
Objetivo
Mostrar la Tabla de multiplicar del uno al 10 del numero que indique el usuario haciendo uso del ciclo de repetición para (for)
Objetivos Específicos
Para obtener transformación de temperaturas de acuerdo a la elección de usuario Algorítmica Pseudocódigo Prueba de Escritorio Diagrama de Flujo Codificación en Matlab

Análisis de Problema		
En este apartado identifica las variables, genera el algoritmo de solución		
Algoritmo de Solución		
<ol style="list-style-type: none"> 1. Inicio 2. Solicitar el número de la tabla a mostrar 3. Realizar y mostrar el producto del número solicitado 4. Fin 		
Datos de Entrada	Proceso	Salida
Numero		
Valor	Numero * Valor	producto

Prueba de Escritorio							
En este apartado realiza la prueba de escritorio de al menos tres ejemplos con el algoritmo que desarrollo							
Variables identificadas						Visualiza el usuario	
numero	valor					Que tabla de multiplicar desea	
	7						
1						7 * 1= 7	
2						7 * 2= 14	
3						7 * 3= 21	
4						7 * 4= 28	
5						7 * 5= 35	
6						7 * 6= 42	



7							$7 * 7 = 49$
8							$7 * 8 = 56$
9							$7 * 9 = 63$
10							$7 * 10 = 70$
							Que tabla de multiplicar desea
	15						
1							$15 * 1 = 15$
2							$15 * 2 = 30$
3							$15 * 3 = 45$
4							$15 * 4 = 60$
5							$15 * 5 = 75$
6							$15 * 6 = 90$
7							$15 * 7 = 105$
8							$15 * 8 = 120$
9							$15 * 9 = 135$
10							$15 * 10 = 150$
							Que tabla de multiplicar desea
	123						
1							$123 * 1 = 123$
2							$123 * 2 = 246$
3							$123 * 3 = 369$
4							$123 * 4 = 492$
5							$123 * 5 = 615$
6							$123 * 6 = 738$
7							$123 * 7 = 861$
8							$123 * 8 = 984$
9							$123 * 9 = 1107$
10							$123 * 10 = 1230$

Resultados

En este apartado se realiza la codificación del algoritmo anterior

Pseudocódigo Solución

Algoritmo tablaMultiplicar

Definir numero como Entero;

Definir valor como Entero;

Escribir "Que tabla de multiplicar desea";

Leer valor;

Para numero <- 1 Hasta 10 Con Paso 1 Hacer

Escribir valor, " * ", numero, "=", valor * numero;

FinPara

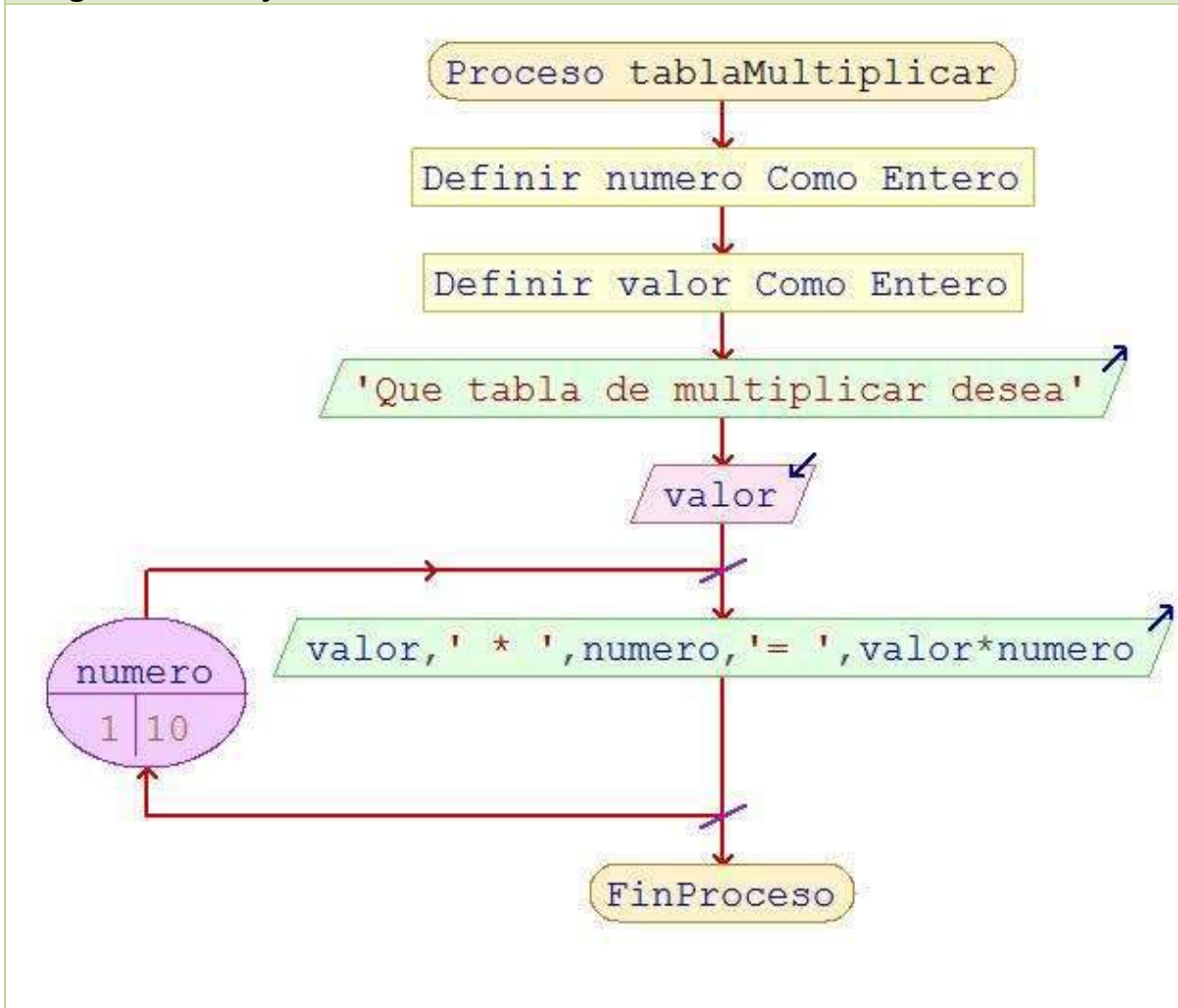
FinAlgoritmo



Resultados

En este apartado realiza el diagrama de flujo correspondiente

Diagrama de Flujo





```
PSelnt - Ejecutando proceso TABLAMULTIPLICAR - [ ] [X]
*** Ejecución Iniciada. ***
Que tabla de multiplicar desea
> 7
7 * 1= 7
7 * 2= 14
7 * 3= 21
7 * 4= 28
7 * 5= 35
7 * 6= 42
7 * 7= 49
7 * 8= 56
7 * 9= 63
7 * 10= 70
*** Ejecución Finalizada. ***
 No cerrar esta ventana  Siempre visible Reiniciar
```

```
PSelnt - Ejecutando proceso TABLAMULTIPLICAR - [ ] [X]
*** Ejecución Iniciada. ***
Que tabla de multiplicar desea
> 15
15 * 1= 15
15 * 2= 30
15 * 3= 45
15 * 4= 60
15 * 5= 75
15 * 6= 90
15 * 7= 105
15 * 8= 120
15 * 9= 135
15 * 10= 150
*** Ejecución Finalizada. ***
 No cerrar esta ventana  Siempre visible Reiniciar
```

Resultados

En este apartado realiza la codificación del algoritmo planteado a Matlab

Codificación

```
valor= input('Que tabla de multiplicar desea ');
for numero=1:10
    fprintf('%2.0f * %2.0f = %2.0f \n', valor, numero, valor*numero);
end
```



Resultados

En este apartado realiza la explicación de la ejecución en Matlab

Pantallas Resultado Explicadas

```
Command Window
>> tablasMult
Que tabla de multiplicar desea 7
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
7 * 10 = 70
>> tablasMult
Que tabla de multiplicar desea 15
15 * 1 = 15
15 * 2 = 30
15 * 3 = 45
15 * 4 = 60
15 * 5 = 75
15 * 6 = 90
15 * 7 = 105
15 * 8 = 120
15 * 9 = 135
15 * 10 = 150
>> tablasMult
Que tabla de multiplicar desea 123
123 * 1 = 123
123 * 2 = 246
123 * 3 = 369
123 * 4 = 492
123 * 5 = 615
123 * 6 = 738
123 * 7 = 861
123 * 8 = 984
123 * 9 = 1107
123 * 10 = 1230
>> |
```

El usuario ingresa el número de la tabla de multiplicar que desea ver, el programa le muestra la tabla de multiplicar indicada.



Ficha Técnica			
Práctica No.	10	Unidad No.	4
Nombre de Práctica		Suma de la secuencia $1^1+2^2+3^3+N^N$ leyendo el valor de N	

Definición de Problema
Diseña la solución para la secuencia $1^1+2^2+3^3+N^N$ solicitando el valor de N por teclado
Objetivo
Emplear los ciclos de repetición, permitiendo al usuario visualizar tantas veces solicite la secuencia (while - for)
Objetivos Específicos
Para obtener transformación de temperaturas de acuerdo a la elección de usuario Algorítmica Pseudocódigo Prueba de Escritorio Diagrama de Flujo Codificación en Matlab

Análisis de Problema		
En este apartado identifica las variables, genera el algoritmo de solución		
Algoritmo de Solución		
<ol style="list-style-type: none"> Inicio Solicitar el Valor de N Para cada número elevar a su potencia y Sumarlo Mostrar el resultado Fin 		
Datos de Entrada	Proceso	Salida
n	$s = i^i + s$	s

Prueba de Escritorio					
En este apartado realiza la prueba de escritorio de al menos tres ejemplos con el algoritmo que desarrollo					
Variables identificadas					Visualiza el usuario
R	i	valor	n	s	
1					
0					Ingresé el valor de N
			3		
				0	
	1	1^1			$1^1 +$
				$0+(1^1)$	



	2	2^2			$2^2 +$
			$0+(1^1)+(2^2)$		
	3	3^3			3^3
			$0+(1^1)+(2^2)+(3^3)$		
					32
					Desea ver otra sumatoria 1 Si 0 No
1					
1					
					Ingresé el valor de N
			5		
			0		
	1	1^1			$1^1 +$
			$0+(1^1)$		
	2	2^2			$2^2 +$
			$0+(1^1)+(2^2)$		
	3	3^3			$3^3 +$
			$0+(1^1)+(2^2)+(3^3)$		
	4	4^4			$4^4 +$
			$0+(1^1)+(2^2)+(3^3)+(4^4)$		
	5	5^5			$5^5 +$
			$0+(1^1)+(2^2)+(3^3)+(4^4) + (5^5)$		
					3413
					Desea ver otra sumatoria 1 Si 0 No
1					
					Ingresé el valor de N
			2		
			0		
	1	1^1			$1^1 +$
			$0+(1^1)$		
	2	2^2			$2^2 +$
			$0+(1^1)+(2^2)$		
					5
					Desea ver otra sumatoria 1 Si 0 No
0					



Resultados

En este apartado se realiza la codificación del algoritmo anterior

Pseudocódigo Solución

Algoritmo SumatoriaN

Definir r,i,valor,n,s Como Entero;

r <- 1;

Mientras r=1 Hacer

 r <- 0;

 Escribir 'Ingrese el valor de N ';

 Leer n;

 s <- 0;

 Para i<-1 Hasta n Hacer

 valor <- i^i ;

 Escribir i,'^',i,' +';

 s <- s+valor;

 FinPara

 Escribir s;

 Escribir 'Desea ver otra sumatoria 1 Si 0 No ';

 Leer r;

FinMientras

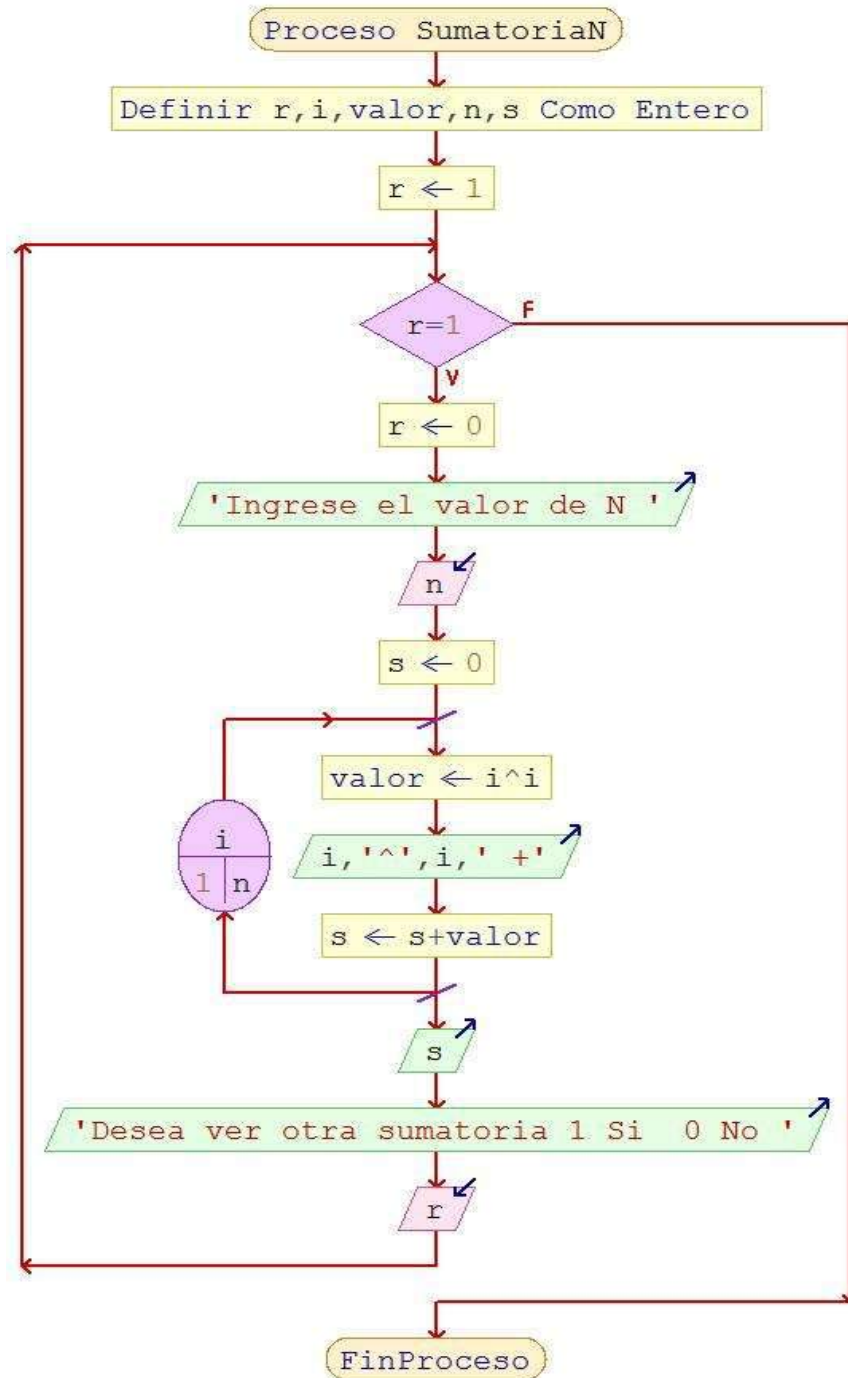
FinAlgoritmo



Resultados

En este apartado realiza el diagrama de flujo correspondiente

Diagrama de Flujo





```
PSelnt - Ejecutando proceso SUMATORIAN
*** Ejecución Iniciada. ***
Ingrese el valor de N
> 3
1^1 +
2^2 +
3^3 +
32
Desea ver otra sumatoria 1 Si 0 No
> 1
Ingrese el valor de N
> 5
1^1 +
2^2 +
3^3 +
4^4 +
5^5 +
3413
Desea ver otra sumatoria 1 Si 0 No
> 0
*** Ejecución Finalizada. ***
```

No cerrar esta ventana Siempre visible Reiniciar



Resultados

En este apartado realiza la codificación del algoritmo planteado a Matlab

Codificación

```
r=1;
while (r==1)
r=0;
n=input('Ingrese el valor de N ');
s=0;
for i=1: n
    valor=i^i;
    fprintf('%d ^ %d) + ', i,i);
    s=s+valor;
end
disp(s);
r=input('Desea ver otra sumatoria 1 Si 0 No ');
end
```

Resultados

En este apartado realiza la explicación de la ejecución en Matlab

Pantallas Resultado Explicadas

```
Command Window
Ingrese el valor de N 3
(1 ^ 1) + (2 ^ 2) + (3 ^ 3) +      32

Desea ver otra sumatoria 1 Si 0 No 1
Ingrese el valor de N 5
(1 ^ 1) + (2 ^ 2) + (3 ^ 3) + (4 ^ 4) + (5 ^ 5) +      3413

Desea ver otra sumatoria 1 Si 0 No 0
fx >>
```

El usuario ingresa el valor de N y el programa realiza la serie numérica mostrando el resultado



Ficha Técnica			
Práctica No.	11	Unidad No.	1,2,3,4
Nombre de Práctica		Números Arábigos a Romanos	

Definición de Problema
Diseñar la solución para que el usuario ingrese por teclado un numero arábigo y visualice uno romano, realiza el proceso tantas veces como lo desee
Objetivo
Hacer uso de todas las sentencias necesarias para dar solución al problema (sentencias if- else, switch, while y for)
Objetivos Específicos
Para obtener Transformación de números arábigos a romanos Algorítmica Pseudocódigo Prueba de Escritorio Diagrama de Flujo Codificación en Matlab

Análisis de Problema		
En este apartado identifica las variables, genera el algoritmo de solución		
Algoritmo de Solución		
<ol style="list-style-type: none"> 1. Inicio 2. Mientras el usuario lo solicite transformar arábigo a romano 3. Solicitar numero arábigo a transformar 4. Si está en rango transformar Sino indicar fuera de rango 5. Mostrar numero Romano 6. Preguntar si desea otra transformación 7. Fin 		
Datos de Entrada	Proceso	Salida
Numero arábigo	transformar	Numero romano



Prueba de Escritorio

En este apartado realiza la prueba de escritorio de al menos tres ejemplos con el algoritmo que desarrollo

Variables identificadas					Visualiza el usuario
num	i	r	residu o	Numerador	
		1			
0					Ingresar un número entre 1 y 3999
435					
6					Valor fuera de rango
					Deseas ver otro número 1 Si 2 No
		1			
0					Ingresar un número entre 1 y 3999
546					
			46	5	D
			6	4	XL
			6		VI
					Deseas ver otro número 1 Si 2 No
		1			
0					Ingresar un número entre 1 y 3999
389			899	3	
9					M
					M
					M
			99	8	D
					C
					C
					C
			90		XC
				9	IX
					Deseas ver otro número 1 Si 2 No
		2			



```
PSeInt - Ejecutando proceso ARABIGOROMANO
*** Ejecución Iniciada. ***
Ingresa un numero entre 1 y 3999
> 4356
Valor fuera de rango
Deseas ver otro numero 1 Si 2 No
> 1
Ingresa un numero entre 1 y 3999
> 546
D
XL
V
Deseas ver otro numero 1 Si 2 No
> 1
Ingresa un numero entre 1 y 3999
> 3899
M
M
M
D
C
C
C
XC
IX
Deseas ver otro numero 1 Si 2 No
> 2
*** Ejecución Finalizada. ***
 No cerrar esta ventana  Siempre visible Reiniciar
```



Resultados

En este apartado se realiza la codificación del algoritmo anterior

Pseudocódigo Solución

Algoritmo arabigoRomano

Definir num, i, r como Entero;

Definir residuo como Entero;

Definir numerador Como Entero;

r<-1;

Mientras(r=1) Hacer

num<-0;

Escribir "Ingresa un numero entre 1 y 3999";

Leer num;

Si num<=3999 y num>0 Entonces

residuo<-num MOD 1000;

numerador<-trunc (num/1000);

Si(residuo<>0)entonces

Para i<-1 Hasta numerador Con Paso 1 Hacer

Escribir "M";

FinPara

FinSi

num<-residuo;

residuo<-num MOD 100;

numerador<-trunc (num/100);

Segun numerador Hacer

9:

Escribir "CM";

4:

Escribir "CD";

5:

Escribir "D";

De Otro Modo:

Si numerador>5 Entonces

Escribir "D";

Para i<-1 Hasta numerador-5 Con Paso 1 Hacer

Escribir "C";

FinPara

Sino

Para i<-1 Hasta numerador Con Paso 1 Hacer

Escribir "C";

FinPara



```

    FinSi
FinSegun
num<-residuo;
residuo<-num MOD 10;
numerador<-trunc (num/10);
Segun numerador Hacer
    9:
        Escribir "XC";
    4:
        Escribir "XL";
    5:
        Escribir "L";
De Otro Modo:
    Si numerador>5 Entonces
        Escribir "L";
        Para i<-1 Hasta numerador-5 Con Paso 1 Hacer
            Escribir "X";
        FinPara
    Sino
        Para i<-1 Hasta numerador Con Paso 1 Hacer
            Escribir "X";
        FinPara
    FinSi
FinSegun
Segun residuo Hacer
    9:
        Escribir "IX";
    4:
        Escribir "IV";
    5:
        Escribir "V";
De Otro Modo:
    Si residuo>5 Entonces
        Escribir "V";
        Para i<-1 Hasta numerador-5 Con Paso 1 Hacer
            Escribir "I";
        FinPara
    Sino
        Para i<-1 Hasta residuo Con Paso 1 Hacer
            Escribir "I";
        FinPara
    FinSi
FinSegun
```



Sino

Escribir "Valor fuera de rango";

FinSi

Escribir "Deseas ver otro número 1 Si 2 No";

Leer r;

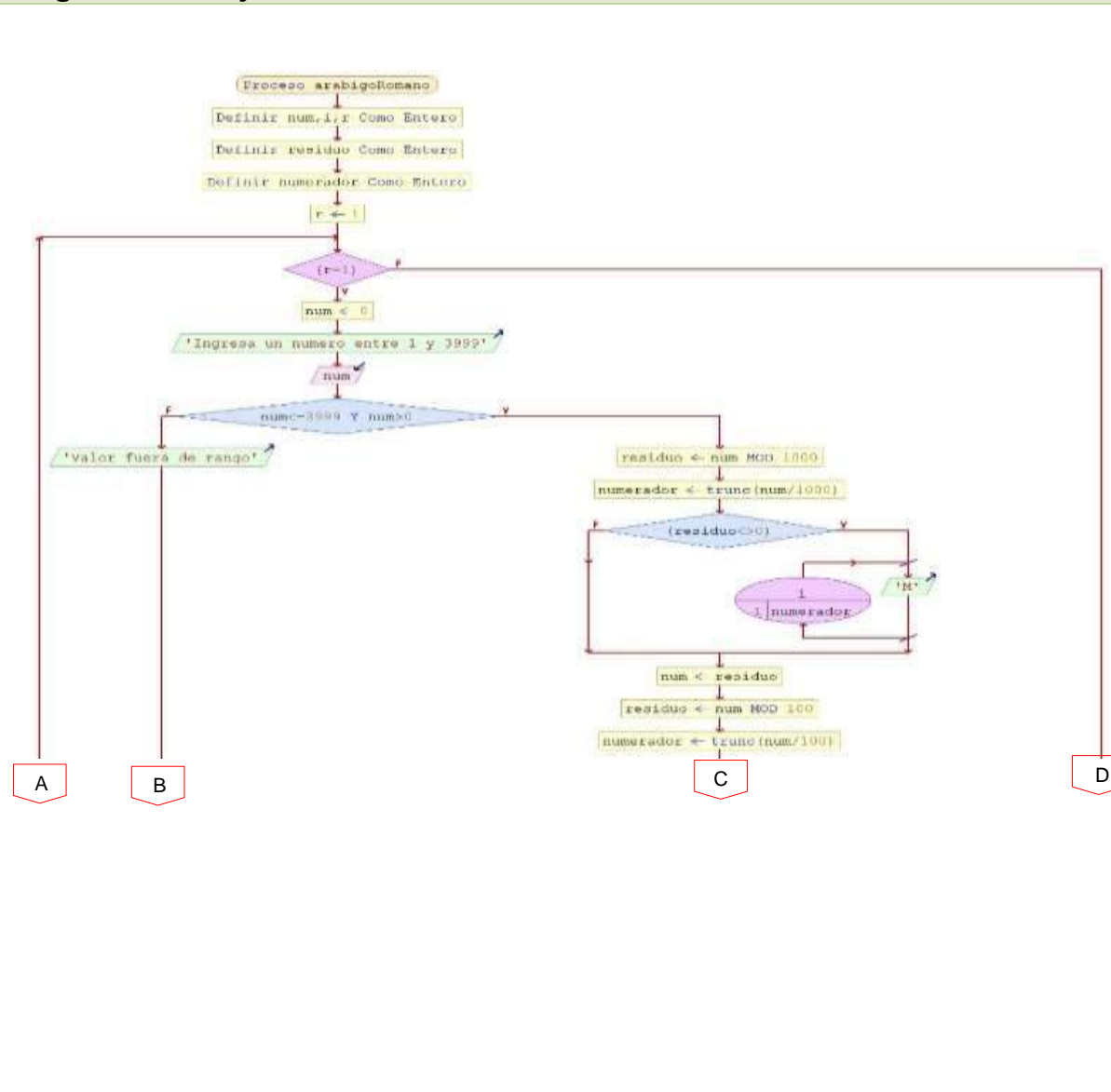
FinMientras

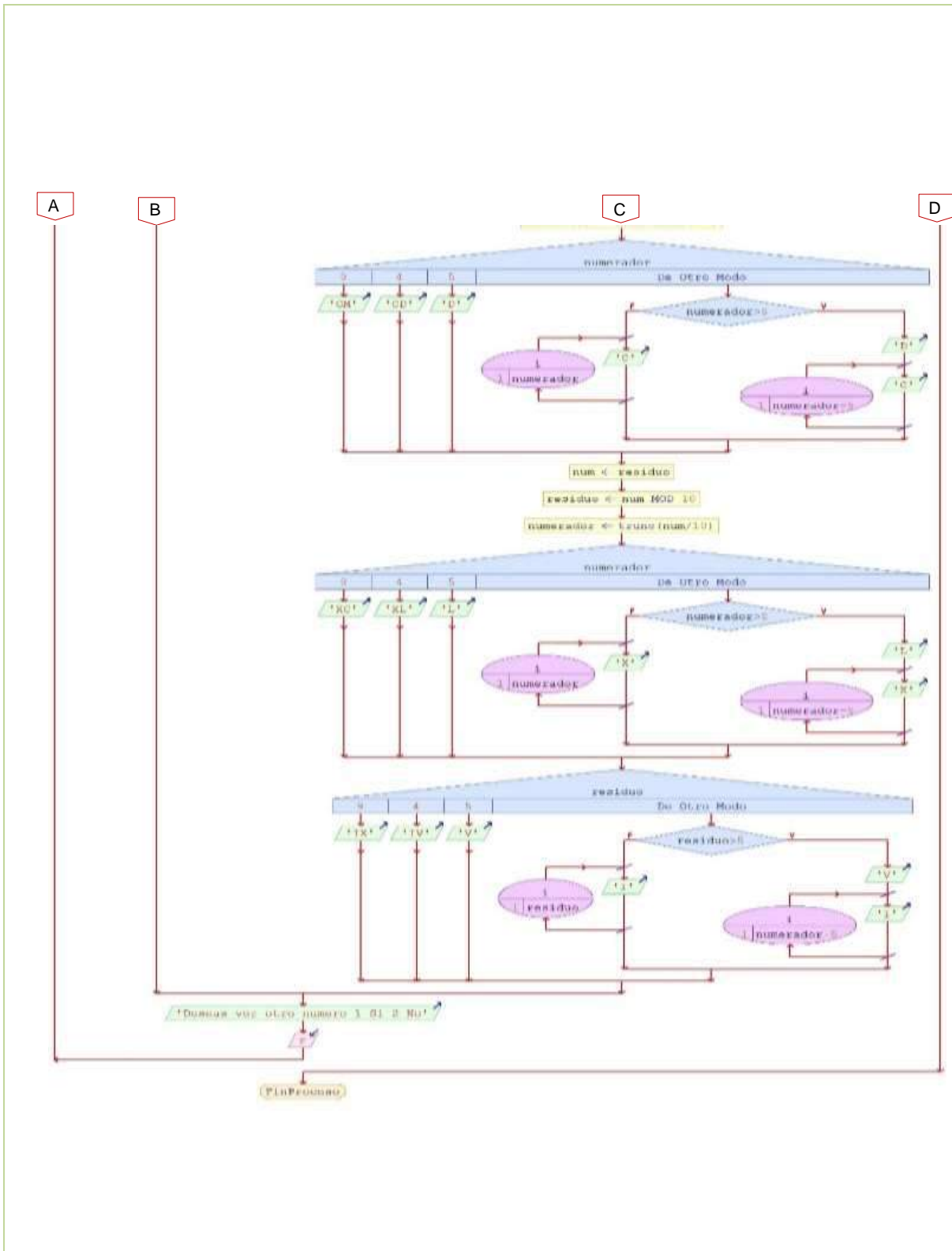
FinAlgoritmo

Resultados

En este apartado realiza el diagrama de flujo correspondiente

Diagrama de Flujo







Resultados

En este apartado realiza la codificación del algoritmo planteado a Matlab

Codificación

```
disp('Numero arábigo a Romano')
r = 1;
while (r==1)
    num=input('Ingresa un numero entre 1 y 3999 ');
    if (num<=3999 && num>0)
        residuo = mod(num,1000);
        numerador = fix(num/1000);
        if (residuo~=0)
            for i=1:numerador
                disp ('M');
            end
        end
        num = residuo;
        residuo = mod(num,100);
        numerador = fix(num/ 100);
        switch numerador
            case 9
                disp('CM');
            case 4
                disp('CD');
            case 5
                disp('D');
            otherwise
                if numerador>5
                    disp 'D';
                    for i=1:numerador-5
                        disp('C');
                    end
                else
                    for i=1:numerador
                        disp('C');
                    end
                end
            end
        end
        num = residuo;
        residuo = mod(num,10);
        numerador = fix(num/ 10);
        switch numerador
            case 9
                disp('XC');
            case 4
```



```
        disp('XL');
    case 5
        disp('L');
    otherwise
        if numerador>5
            disp('L');
            for i=1:numerador-5
                disp('X');
            end
        else
            for i=1:numerador
                disp('X');
            end
        end
    end
end
switch residuo
case 9
    disp('IX');
case 4
    disp('IV');
case 5
    disp('V');
otherwise
    if residuo>5
        disp('V');
        for i=1:numerador-5
            disp('I');
        end
    else
        for i=1:residuo
            disp('I');
        end
    end
end
end

else
    disp('Valor fuera de rango');
end
r=input('Deseas ver otro número 1 Si 2 No ');
end
```



Resultados

En este apartado realiza la explicación de la ejecución en Matlab

Pantallas Resultado Explicadas

```
Command Window
>> clear all
>> ArabigoRomano
Numero arábigo a Romano
Ingresa un numero entre 1 y 3999 4324
Valor fuera de rango
Deseas ver otro numero 1 Si 2 No 1
Ingresa un numero entre 1 y 3999 3245
M
M
M
C
C
XL
V
Deseas ver otro numero 1 Si 2 No 1
Ingresa un numero entre 1 y 3999 12
X
I
I
Deseas ver otro numero 1 Si 2 No 2
fx >>
```

El usuario ingresa el numero arábigo y el programa le regresa el numero romano
En este programa se hace uso de todas las sentencias de selección de condición y de iteración.



Ficha Técnica			
Práctica No.	12	Unidad No.	5
Nombre de Práctica		Arreglos en Matlab	

Definición de Arreglos en Matlab

En Matlab la creación de vectores unidimensionales es muy práctico y sencillo.

Definición de un Vector: nombre=[componentes]

Notación para los componentes de un vector

Notación Matemática: x_i

Notación Matlab: **x(i)**

Donde **x** es el nombre del vector e **i** es el número de la celda, numeradas en forma natural desde **1**

El siguiente listado ilustra algunos ejemplos:

Instrucción	Descripción
vector = [] ;	Declara un vector vacío
v = [5 7 9 1];	Crear vector fila de cuatro componentes
t = [4 ; 6; 8; 3]	Crear vector columna de cuatro componentes
v= [v, 3]	Agregar un nuevo elemento a la derecha del vector
v= [7, v]	Agregar un nuevo elemento a la izquierda del vector
v(3) =[]	Eliminar el tercer elemento del vector
v(4) = 2	Sustituir el cuarto elemento por 2
x= length(v)	La longitud de un vector almacenada en una variable
y= fix(rand*10);	Asignar un nuevo elemento aleatorio a la derecha del vector
v=[v, x]	
v(i)=3	Asignación de un elemento en una posición específica designada por la variable i

El alumno debe probar cada instrucción y analizar su funcionamiento, cabe destacar que existen mas sentencias en el uso de vectores.



Ficha Técnica			
Práctica No.	13	Unidad No.	5
Nombre de Práctica		Llenado de un vector	

Definición de Problema
Diseñar la solución para el llenado de un vector de 10 posiciones de forma aleatoria y mostrarlo
Objetivo
Haciendo uso de las sentencias adecuadas llenar un vector
Objetivos Específicos
Para obtener transformación de temperaturas de acuerdo a la elección de usuario Algorítmica Pseudocódigo Prueba de Escritorio Diagrama de Flujo Codificación en Matlab

Análisis de Problema		
En este apartado identifica las variables, genera el algoritmo de solución		
Algoritmo de Solución		
<ol style="list-style-type: none"> 1. Inicio 2. Llenar de forma aleatoria un arreglo de diez posiciones 3. Mostrar el vector en el orden en que se fue llenando 4. Mostrar el vector en orden invertido a como se fue llenando 5. Fin 		
Datos de Entrada	Proceso	Salida
A[10]	Llenar de forma aleatoria	
	Recorrer para mostrar	Mostrar vector
	Recorrer a la inversa para mostrar	Mostrar Vector en forma inversa

Prueba de Escritorio	
En este apartado realiza la prueba de escritorio de al menos tres ejemplos con el algoritmo que desarrollo	
Variables identificadas	Visualiza el usuario
A[]	1
	1
25	
	2
86	
	3



19							
	4						
80							
	5						
91							
	6						
81							
	7						
93							
	8						
19							
	9						
13							
	10						
10							
							Los elementos del arreglo son
	1						Posición 1: 25
	2						Posición 2: 86
	3						Posición 3: 19
	4						Posición 4: 80
	5						Posición 5: 91
	6						Posición 6: 81
	7						Posición 7: 93
	8						Posición 8: 19
	9						Posición 9: 13
	10						Posición 10: 10
							En orden inverso
	10						Posición 10: 10
	9						Posición 9: 13
	8						Posición 8: 19
	7						Posición 7: 93
	6						Posición 6: 81
	5						Posición 5: 91
	4						Posición 4: 80
	3						Posición 3: 19
	2						Posición 2: 86
	1						Posición 1: 25



Resultados

En este apartado se realiza la codificación del algoritmo anterior

Pseudocódigo Solución

Algoritmo llenaVector

Definir A,i Como Entero;

// declara un arreglo de 10 elementos

Dimension A[11];

// recorre los 10 elementos y va asignandoles enteros aleatorios

Para i<-1 Hasta 10 Hacer

 // elemento toma el contenido de cada posicion del arreglo

 // y si se modifica elemento se modifica el arreglo

 A[i] <- azar(100);

FinPara

Escribir 'Los elementos del arreglo son:';

// recorre los 10 elementos utilizando subindices y los muestra en pantalla

Para i<-1 Hasta 10 Hacer

 Escribir 'Posición ',i,': ',A[i];

FinPara

Escribir "; // deja una linea en blanco

Escribir 'En orden inverso:';

// recorre los 10 elementos en orden inverso y los muestra en una misma linea

Para i<-10 Hasta 1 Con Paso -1 Hacer

 Escribir ' sin bajar ',A[i], ' ';

FinPara

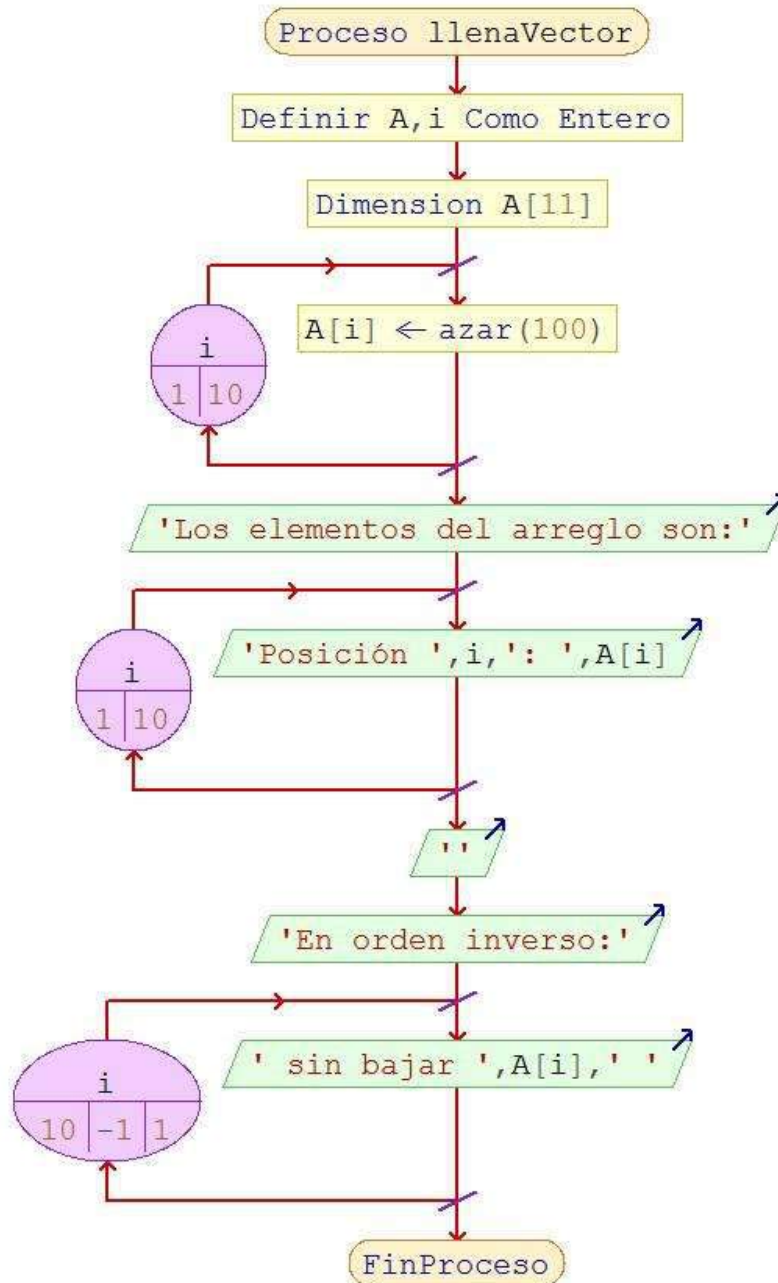
FinAlgoritmo



Resultados

En este apartado realiza el diagrama de flujo correspondiente

Diagrama de Flujo





```
PSelnt - Ejecutando proceso LLENAVECTOR
*** Ejecución Iniciada. ***
Los elementos del arreglo son:
Posición 1: 25
Posición 2: 86
Posición 3: 19
Posición 4: 80
Posición 5: 91
Posición 6: 81
Posición 7: 93
Posición 8: 19
Posición 9: 13
Posición 10: 18

En orden inverso:
sin bajar 18
sin bajar 13
sin bajar 19
sin bajar 93
sin bajar 81
sin bajar 91
sin bajar 80
sin bajar 19
sin bajar 86
sin bajar 25
*** Ejecución Finalizada. ***
```

No cerrar esta ventana Siempre visible Reiniciar



Resultados

En este apartado realiza la codificación del algoritmo planteado a Matlab

Codificación

```
%% declara un arreglo de 10 elementos
A=[];
%% recorre los 10 elementos y va asignandoles enteros aleatorios
for i=1:10 %%si no indica los incrementos por definición es uno
    %% elemento toma el contenido de cada posición del arreglo
    %% y si se modifica elemento se modifica el arreglo
    A = [A, fix(rand*100)];
end
disp('Los elementos del arreglo son:');
%% recorre los 10 elementos utilizando subíndices y los muestra en pantalla
for i=1:10
    fprintf('Posición A[%d]= %d \n',i,A(i));
end
disp(""); %% deja una línea en blanco
disp('En orden inverso:');
%% recorre los 10 elementos en orden inverso y los muestra en una misma
línea
for i=10:-1:1
    fprintf('Posición A[%d]= %d \n',i,A(i));
end
```

Resultados

En este apartado realiza la explicación de la ejecución en Matlab

Pantallas Resultado Explicadas

```
Command Window
>> llenaVector
Los elementos del arreglo son:
Posición A[1]= 84
Posición A[2]= 25
Posición A[3]= 51
Posición A[4]= 24
Posición A[5]= 92
Posición A[6]= 34
Posición A[7]= 19
Posición A[8]= 25
Posición A[9]= 61
Posición A[10]= 47
En orden inverso:
Posición A[10]= 47
Posición A[9]= 61
Posición A[8]= 25
Posición A[7]= 19
Posición A[6]= 34
Posición A[5]= 92
Posición A[4]= 24
Posición A[3]= 51
Posición A[2]= 25
Posición A[1]= 84
/>> |
```

Se llena el vector de 10 posiciones con números generados aleatoriamente los muestra y después los muestra en forma inversa a como se fueron ingresando al arreglo

Ficha Técnica			
Práctica No.	14	Unidad No.	5
Nombre de Práctica	Decimal a Binario		

Definición de Problema
Diseñar la solución para la Transformación de numero decimal a binario, solicitando por teclado el ingreso del numero decimal
Objetivo
Haciendo uso de las sentencias adecuadas transformar de decimal a binario usando un vector
Objetivos Específicos
Para obtener transformación de temperaturas de acuerdo a la elección de usuario
Algorítmica
Pseudocódigo
Prueba de Escritorio
Diagrama de Flujo
Codificación en Matlab

Análisis de Problema		
En este apartado identifica las variables, genera el algoritmo de solución		
Algoritmo de Solución		
<ol style="list-style-type: none"> 1. Inicio 2. Solicitar el numero decimal 3. Transformarlo a binario 4. Mostrar el resultado 5. Fin 		
Datos de Entrada	Proceso	Salida
x, b, d, i, j	Transformar decimal a binario	Mostrar binario

Prueba de Escritorio						
En este apartado realiza la prueba de escritorio de al menos tres ejemplos con el algoritmo que desarrollo						
Variables identificadas					Visualiza el usuario	
b	x	l	d	j		
						Numero Decimal a binario
	345					ingrese un número
				1		
			1			
	172					
1				2		
			0			
	86					
10						
				3		



		0			
	43				
100			4		
		1			
	21				
1001			5		
		1			
	10				
10011			6		
		0			
	5				
100110			7		
		1			
	2				
1001101			8		
		0			
	1				
10011010			9		
		1			
	0				
100110101					
					Binario
		9			1
		8			0
		7			1
		6			0
		5			1
		4			1
		3			0
		2			0
		1			1



Resultados

En este apartado se realiza la codificación del algoritmo anterior

Pseudocódigo Solución

Algoritmo decBinario

```
Definir b, x, i, d, j Como Entero;
Dimension b[33];
Escribir "Numero Decimal a binario ";
Escribir "ingrese un número ";
Leer x;
j<-1;
Mientras x>0 Hacer
    d <- x MOD 2;
    x<-trunc(x/2);
    b[j]<-d;
    j<-j+1;
FinMientras
Escribir "Binario ";
Para i<-j-1 Hasta 1 Con Paso -1 Hacer
    Escribir "",b[i];
FinPara
FinAlgoritmo
```

```
PSelnt - Ejecutando proceso DECBINARIO
ingrese un número
> 345
Binario
1
0
1
0
1
1
0
0
1
*** Ejecución Finalizada.
```

```
PSelnt - Ejecutando proceso DECBINARIO
Numero Decimal a binario
ingrese un número
> 45
Binario
1
0
1
1
0
1
*** Ejecución Finalizada. ***
Reiniciar
```




Resultados

En este apartado realiza la codificación del algoritmo planteado a Matlab

Codificación

```
%%mejorando el Pseudocódigo y haciendo algunas validaciones
clc
clear all
disp('Numero Decimal a binario ')
x= input('ingrese un numero ')
n=x
if(x>0)
    b= [ ]
    while x>0
        d= mod(x,2)
        x= fix(x/2)
        b=[d,b]
    end
    clc
    disp('Numero decimal')
    disp(x)
    disp ('Numero binario')
    disp(b)
else
    disp('Debe ser un numero positivo')
end
```

Resultados

En este apartado realiza la explicación de la ejecución en Matlab

Pantallas Resultado Explicadas

```
Command Window
Numero decimal
    345

Numero binario
     1     0     1     0     1     1     0     0     1

fx >>
```

El usuario ingresa el numero decimal, y el programa lo transforma en binario, almacenándolo en un arreglo y posteriormente lo muestra



VIII Practicas sugeridas y ejercicios

Prácticas

1. Realiza el intercambio de tres valores ingresados por el usuario
2. Proporciona el Salario de un empleado solicitando horas trabajadas y pago por hora
3. Clasifica número mayor medio y menor solicitando tres números al usuario, considere que deben ser diferentes
4. De solución a la formula General
5. De acuerdo con la Ley de Ohm, permita al usuario obtener Resistencia, Impedancia o Voltaje.
6. Permita al usuario realizar la transformación de pies a yarda, centímetro, metro o pulgada según elija el usuario
7. Mostrar un mensaje con incrementos de 5
8. Calcular el total que una persona debe pagar en un almacén de llantas. El precio de cada llanta es de \$80 si se compran menos de 5 llantas y de \$70 se compran 5 o más.
9. Suponga que un local vende llantas según cantidad y precio, es decir, menos de 4 no tienen descuento, 4 o 5 tienen un 5% de descuento, y mas de 5 un 10% de descuento.
10. Dado un entero entre 1 y 5 mostrar el día de la semana, valide que el valor este dentro del rango.
11. Dado un número entero, liste las raíces cuadradas de los enteros impares entre 1 y el dato dado



Ejercicios de programación con las instrucciones básicas

- 1.- Dados el radio y altura de un cilindro calcule el área total y el volumen
- 2.- Se tiene un recipiente cilíndrico con capacidad de x litros. Su altura es h metros. Determine el diámetro de la base
- 3.- Calcule el promedio de las tres calificaciones de las lecciones que obtuvo un estudiante
- 4.- Dadas las tres dimensiones de un recipiente rectangular, calcule su volumen

La siguiente fórmula proporciona el enésimo término u de una progresión aritmética:

$u = a + (n - 1)r$ en donde a es el primer término, n es el la cantidad de términos y r es la razón entre dos términos consecutivos

Calcular el valor de r dados u, a, n

En el ejercicio anterior, calcular el valor de: n dados u, a, r

7.- Un modelo de crecimiento poblacional está dado por: $n = 5t + 2e^{0.1t}$, en donde n es el número de habitantes, t es tiempo en años.

Calcule el número de habitantes que habrían en tres años consecutivos

8- Un ingeniero desea tener una cantidad de dólares acumulada en su cuenta de ahorros para su retiro luego de una cantidad de años de trabajo. Para este objetivo planea depositar un valor mensualmente. Suponga que el banco acumula el capital mensualmente mediante la siguiente fórmula:

$$A = P \frac{(1+x)^n - 1}{x}, \text{ en donde}$$

- A: Valor acumulado
- P: Valor de cada depósito mensual
- n: Cantidad de depósitos mensuales
- x: Tasa de interés mensual

Calcule el valor acumulado ingresando como datos valores para P, n, x

9.- Una empresa produce fertilizantes. Cada mes el ingreso por ventas en miles de dólares se describe con $v = 0.4x(30 - x)$ mientras que el costo de producción en miles de dólares es $v = 5 + 10 \ln(x)$, siendo x la cantidad producida en toneladas, $1 < x < 30$. Determine el valor del ingreso neto, dado un valor para x.



Ejercicios de programación con decisiones

- 1.- Dados el radio y altura de un cilindro, si la altura es mayor al radio calcule y muestre el valor del volumen, caso contrario muestre el mensaje: 'Error'
- 2.- Dadas las dimensiones de un bloque rectangular, calcule y muestre el área de la cara de mayor dimensión.
- 3.- Lea un número de dos cifras. Determinar si la suma de ambas cifras es un número par o impar. Muestre un mensaje
- 4.- Lea un número. Determine si es entero y múltiplo de 7
- 5.- Lea la cantidad de Kw que ha consumido una familia y el precio por Kw. Si la cantidad es mayor a 700, incremente el precio en 5% para el exceso de Kw sobre 700. Muestre el valor total a pagar.
- 6.- Lea un valor de temperatura t y un código p que puede ser 1 o 2. Si el código es 1 convierta la temperatura t de grados f a grados c con la fórmula $c=5/9(t-32)$. Si el código es 2 convierta la temperatura t de grados c a f con la fórmula: $f=32+9t/5$
- 7.- Dadas las tres calificaciones de un estudiante, encuentre y muestre su calificación final.
- 8.- Dados los tres lados de un triángulo determine su tipo: Equilátero, Isòsceles, o Escaleno
- 9.- El número de pulsaciones que debe tener una persona por cada 10 segundos de ejercicio aeróbico se calcula con la fórmula:
Género femenino (1): número de pulsaciones = $(220 - \text{edad en años})/10$
Género masculino (2): número de pulsaciones = $(210 - \text{edad en años})/10$
Lea la edad y el género y muestre el número de pulsaciones



10.- El valor de una pizza depende de su tamaño:

Tamaño	precio
1	\$5
2	\$8
3	\$12

Cada ingrediente adicional cuesta \$1.5. Lea el tamaño de la pizza y el número de ingredientes y muestre el valor que debe pagar

11.- Dado el peso P en Kg. de una persona y su talla T en mt. El índice de masa corporal IMC se calcula con la fórmula P/T^2

Determine el nivel de sobrepeso de una persona de 25 a 34 años.

IMC	Nivel
Menos de 18.5	Bajo peso
18.5 a 25.5	Peso Normal
25.5 a 30.5	Sobrepeso grado 1
30.5 a 35.5	Sobrepeso grado 2
35.5 a 40.5	Obesidad grado 3
Más de 40.5	Obesidad grado 4



Ejercicios de programación con ciclos

1.- Calcule el promedio, el menor valor y el mayor valor de los pesos de n paquetes en una bodega. Estos datos ingresan uno a la vez dentro de un ciclo. n es un dato ingresado al inicio.

2.- Clasifique los pesos de los n objetos de una bodega en tres grupos: menor a 10 Kg., entre 10 y 20 Kg., mas de 20 Kg. Los datos ingresan uno a la vez en un ciclo.

3.- Dado un número par n , sume los primeros n números impares: Ejemplo. $n=4$: $s = 1+3+5+7 = 16$ Verifique si esta s suma es igual a n^2 y muestre un mensaje

4.- Dado dos números enteros a , b , determine su máximo común divisor m .
Ejemplo: $a = 36$, $b = 45$ entonces $m = 9$

5.- Dado un número entero positivo n , descompóngalo en sus factores primos
Ejemplo: $n = 72$, factores primos: 2, 2, 3, 3, 3

6.- Lea los votos de n personas. Cada voto es un número 1, 2, o 3 correspondiente a tres candidatos. Si el dato es 0 es un voto en blanco. Si es otro número es un voto nulo. Determine el total de votos de cada candidato y el total de votos blancos y nulos.

7.- Lea las coordenadas de u , v de la ubicación de una fábrica y las coordenada x , y de n sitios de distribución. Encuentre cual es la distancia del sitio más alejado de la fábrica

8.- Encuentre el mayor valor de la función $f(x)=\text{sen}(x)+\ln(x)$, para los valores:

$x=1.0, 1.1, 1.2, 1.3, \dots, 4$

9.- Lea las coordenadas de n puntos en un plano. Calcule la suma de las distancias de los puntos en un recorrido consecutivo del punto 1 al punto n .



10.- Determine la suma de los términos de la serie $1^3 + 3^3 + 5^3 + \dots + n^3$ en donde n es un dato

11.- Determine la suma de los n primeros números de la serie:

1, 1, 2, 3, 5, 8, 13, 21,

en la cual cada término, a partir del tercero, se obtiene sumando los dos términos anteriores 12.- Determine la suma de los números $2^1 + 2^2 + 2^3 + 2^4 + \dots + 2^{64}$

13.- Una empresa compra una máquina en \$20000 pagando cuotas anuales durante cinco años. La siguiente fórmula relaciona el costo de la máquina P, el pago anual A, el número de años n y el interés anual r:

$$P = \frac{A \left(\frac{1-r^{n+1}}{1-r} \right)}{1-r}$$

Escriba un programa que permita calcular el valor de P para valores de r = 0.01, 0.02, 0.03, ..., 0.1

14.- Escriba un programa para controlar la cantidad de contenedores en un patio.

Ingrese como dato la cantidad inicial y ofrezca las siguientes opciones:

Salida de contenedores

Llegada de contenedores

Cantidad actual de contenedores

Terminar el control

En cada repetición el operador elige la opción ingresando el número y la cantidad de contenedores.

15.- En un supermercado se hace una promoción, mediante la cual el cliente obtiene un descuento dependiendo de un número de una cifra que se escoge al azar.



Ejercicios de programación con vectores

1.- Lea una lista de los pesos de las cajas en un contenedor. Determine cuantos exceden al peso promedio del grupo.

2.- Lea una lista de los pesos de los objetos en una bodega. Determine cual es el rango de los pesos de los objetos.

3.- Escriba un programa para controlar el uso de los camiones de una empresa. Cada camión tiene un código. Ingrese como dato inicial la lista de los códigos de los camiones. Programe una aplicación con las siguientes opciones:

Salida de un camión

Devolución de un camión

Disponibilidad de un camión

Terminar

El operador elige la opción ingresando el número.

4.- Lea los códigos de las cajas de un contenedor. Determine si hay códigos repetidos

5.- Una bodega contiene n paquetes numerados en forma natural. Para una inspección se debe tomar una muestra aleatoria del 10% de los paquetes. Escriba un programa para elegir la muestra.

6.- Para la inspección de los m paquetes de una bodega se han elegido a m inspectores. Realice aleatoriamente la asignación de tal manera que cada inspector se le asigne la revisión de un solo paquete.



7.- Se tiene la lista de los códigos de las cajas distribuidas en dos bodegas y una lista de los códigos de las cajas que se requieren para una inspección. Determine cuantas de estas cajas están en cada bodega.

8. Almacene en un vector U las abscisas y en un vector V las ordenadas de un conjunto de n puntos en un plano. Determine cuál es el punto más alejado del origen. Use la fórmula de la distancia.

9.- Cálculo del área de un polígono arbitrario, dadas las coordenadas de sus vértices $(x_1, y_1)(x_2, y_2), \dots, (x_n, y_n)$

$$\text{Area} = [(x_1+x_2)(y_1-y_2) + (x_2+x_3)(y_2-y_3) + \dots + (x_{n-1}+x_n)(y_{n-1}-y_n) + (x_n+x_1)(y_n-y_1)]/2$$

Escriba un programa para leer los valores de x en un vector y los valores de y en otro vector. Calcule u muestre el valor del área con la fórmula anterior

10. En un proceso electoral se tienen anotados los n votos para aprobar una moción. Cada voto tiene el número de identificación del elector (números enteros del 1 al n) y un número que representa su decisión: 1 si es a favor, 0 si es en contra, cualquier otro número es nulo. Escriba un programa que lea los n datos conteniendo el número del elector (no suponga que están ordenados) y su voto. Coloque los números de identificación en tres listas: votantes a favor, votantes en contra y votantes nulos. Finalmente busque y muestre si hay números de identificación de electores que están en más de una lista.



VII. Mapa curricular

	PERIODO 1	PERIODO 2	PERIODO 3	PERIODO 4	PERIODO 5	PERIODO 6	PERIODO 7	PERIODO 8	PERIODO 9	PERIODO 10
O B L I G A T O R I A S	Algebra 2 2 4 6	Álgebra lineal 2 2 4 6	Probabilidad y estadística 4 0 4 8	Investigación de operaciones I 2 2 4 6	Investigación de operaciones II 2 2 4 6	Diseño de sistemas de manufactura 3 2 4 8	Procesos de manufactura de transformación 2 2 4 6	Procesos de manufactura de mecanizado 2 2 4 6	Proyecto de ingeniería 1 2 3 4	P r á c t i c a P r o f e s i o n a l : 36
	Geometría analítica 4 0 4 8	Cálculo diferencial e integral 4 0 4 8	Cálculo multivariable 4 0 4 8	Ecuaciones diferenciales 4 0 4 8	Análisis numérico 4 0 4 8	Administración industrial 3 1 4 7	Administración de la producción 2 2 4 6	Administración de proyectos 2 2 4 6	Ingeniería ambiental y seguridad industrial 2 2 4 6	
	Física 3 1 4 7	Estática 2 2 4 6	Dinámica 4 0 4 8	Mecánica de fluidos 4 0 4 8	Balanco de líneas de producción 2 2 4 6	Ingeniería económica 2 2 4 6	Diseño de herramientas 2 2 4 6	Análisis de mecanismos 4 0 4 8	Síntesis de mecanismos 2 2 4 6	
	Química 2 2 4 6	Comunicación y relaciones humanas 2 2 4 6	Liderazgo 3 1 4 7	Dibujo de detalles 2 2 4 6	Modelado paramétrico 3D 2 2 4 6	Ética profesional 3 0 4 6	Electrónica 2 2 4 6	Producción automatizada 2 2 4 6	Control de calidad 2 2 4 6	
	Programación I 2 2 4 6	Programación II 2 2 4 6	Termodinámica 2 2 4 6	Metrología 2 2 4 6	Electricidad industrial 2 2 4 6	Circuitos eléctricos industriales 2 2 4 6	Dinámica de sistemas 2 2 4 6	Control de procesos industriales 2 2 4 6		
	Sociedad e Ingeniería 3 0 4 6			Estudio del trabajo 2 2 4 6	Mecánica de materiales 2 2 4 6	Ciencias de materiales 4 0 4 8	Integrativa profesional* — — — —			
		Inglés 5 2 2 4 6	Inglés 6 2 2 4 6	Inglés 7 2 2 4 6	Inglés 8 2 2 4 6	Diseño de elementos de máquinas 2 2 4 6				
O P T A T I V A S										
							Optativa 1 2 2 4 6	Optativa 3 2 2 4 6		
							Optativa 2 2 2 4 6	Optativa 4 2 2 4 6		
	HT 16 HP 7 TH 23 CR 35	HT 14 HP 10 TH 24 CR 38	HT 19 HP 5 TH 24 CR 43	HT 18 HP 10 TH 28 CR 45	HT 16 HP 12 TH 28 CR 44	HT 19 HP 9 TH 28 CR 47	HT 12 HP 8** TH 25** CR 40	HT 16 HP 12 TH 28 CR 44	HT 11 HP 12 TH 23 CR 34	HT - HP - TH - CR 30



DISTRIBUCIÓN DE LAS UNIDADES DE APRENDIZAJE OPTATIVAS										
	PERIODO 1	PERIODO 2	PERIODO 3	PERIODO 4	PERIODO 5	PERIODO 6	PERIODO 7	PERIODO 8	PERIODO 9	PERIODO 10
OPTATIVAS								Análisis de mercado 3 2 4 5	Plan de energía en Producción Industrial 2 2 4 5	
								Control estadístico del proceso 2 2 4 5	Diseño del producto 2 2 4 5	
								Planeación de recursos para manufactura 2 2 4 5	Metodías analizadas por computadora 2 2 4 5	
								Psicología industrial 3 2 4 5	Robótica 3 2 4 5	
								Tópicos de calidad 3 2 4 5	Sistemas de manufactura flexible 3 2 4 5	

SIMBOLOGÍA	
Unidad de aprendizaje	HT: Horas Teóricas HP: Horas Prácticas TH: Total de Horas CR: Créditos

*Actividad académica
 ** Horas de las actividades académicas
 9 Líneas de seriación →
 Créditos a cursar por periodo escolar: mínimo 20 y máximo 51

	Obligatorio Núcleo Básico
	Obligatorio Núcleo Sustantivo
	Obligatorio Núcleo Integral
	Optativo Núcleo Integral

PARÁMETROS DEL PLAN DE ESTUDIOS

Núcleo Básico Obligatorio: cursar y acreditar 20 UA	55 24 79 134	Total del Núcleo Básico: acreditar 20 UA para cubrir 134 créditos
Núcleo Sustantivo Obligatorio: cursar y acreditar 25 UA	57 35 92 149	Total del Núcleo Sustantivo: acreditar 23 UA para cubrir 149 créditos
Núcleo Integral Obligatorio: cursar y acreditar 10 UA=2*	21 18** 36** 58	Total del Núcleo Integral: acreditar 14 UA + 2* para cubrir 122 créditos
Núcleo Integral Optativo: cursar y acreditar 4 UA	8 8 16 24	

TOTAL DEL PLAN DE ESTUDIOS	
UA Obligatorias	53 = 2 Actividades académicas
UA Optativas	4
UA a acreditar	57 = 2 Actividades académicas
Créditos	405

X. Bibliografía

- Pablo Novara. (2017). PSeInt. Recuperado en Agosto de 2017, con CopyLeft 2017 Sitio web: <http://pseint.sourceforge.net/>
- The MathWorks, Inc. (2017). Matlab R2017b. Recuperado en Agosto 2017, de © 1994-2017 The MathWorks, Inc. Sitio web: <https://www.mathworks.com/>
- Joyanes Aguilar Luis. (2008). Fundamentos de Programación, Algoritmos y Estructura de Datos. Madrid, España: Mc-Graw-Hill 4ta Edición.
- Rodriguez Ojeda Luis. (2000). Matlab Programación. Escuela Superior Politécnica del Litoral Facultad de Ciencias Naturales y Matemáticas, Departamento de Matemáticas. Guayaquil Ecuador:



XI Anexos

Anexo A Glosario de Términos

Término	Definición
Análisis del Problema	Inicia con la correcta definición y comprensión del problema. Se debe conocer que datos se requieren para construir la solución, los procedimientos específicos a seguir, así como que información debe dar como solución.
Diseño	Es la elaboración de los pasos ordenados precisos y finitos que lleven a la solución del problema, ya sea en Algoritmo, Pseudocódigo o Diagrama de Flujo.
Codificación	Es la conversión del algoritmo, Pseudocódigo o Diagrama de Flujo a un código formal. (puede ser código Matlab, C, java, entre otros)
Pruebas	El Programa debe ser ejecutado en la computadora para corregir errores de lógica, para comprobar los resultados obtenidos.
Mantenimiento	Implica los cambios y actualizaciones que sean necesarios una vez que los usuarios hagan uso del programa
Algoritmo	Conjunto de pasos, procedimientos o acciones ordenados que permiten alcanzar un resultado para resolver un problema. Siendo Finito, Definido, Preciso y General.
Finito	Todo algoritmo tiene inicio y tiene fin
Definido	Todo algoritmo siempre se debe interpretar de la misma manera, no debe ser ambiguo
Preciso	Todo algoritmo debe ser ordenado, cada paso tiene un orden específico
General	Todo algoritmo Siempre dará los mismos resultados, ante el mismo tipo de problema
Pseudocódigo	Es la representación de un Algoritmo en un Lenguaje NO Formal de computadora
identificador	Está formado por una secuencia de caracteres


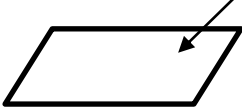
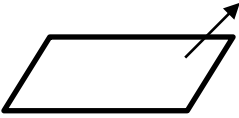

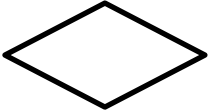
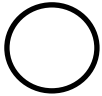





	alfabéticos, numéricos, con los que se da nombre a las variables, constantes, tipos de datos, entre otros.
Palabra reservada	Son identificadores predefinidos, con significado propio. De acuerdo al Lenguaje de programación que se emplee.
Tipos de Dato	Son los diferentes datos con los que un lenguaje de programación trabaja, numéricos, caracteres o lógicos.
Variable	Representa un valor almacenado en memoria que se puede modificar en cualquier momento o conservar para ser usado tantas veces sea necesario
Constantes	Es un dato que permanece sin cambio durante el desarrollo ejecución de un código y no serán alterados por el usuario el valor de Euler, pi, gravedad, entre otros.
Expresiones	Es el resultado de unir operandos mediante operadores
Operadores	Símbolo que permite relacionar dos datos en una expresión y evaluar el resultado de la operación
Diagrama de Flujo	Representación gráfica de la solución, mostrando los pasos a seguir
Prueba de Escritorio	Técnica empleada para realizar pruebas a los códigos, siguiendo su flujo lógico, identificando variables, procesos y lo que el usuario visualizará.
Lenguaje de Programación	Un lenguaje de programación es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Al proceso por el cual se escribe, se prueba, se depura, se compila (de ser necesario) y se mantiene el código fuente de un programa informático se le llama programación.
Matlab	MATLAB (abreviatura de MATrix LABoratory, "laboratorio de matrices") es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). https://www.mathworks.com/
Sentencias selectivas	La sentencia de control evalúa la condición lógica o booleana



If	SI (condicional). Se evalúa una condición, si es verdadera ejecuta un código, si es falsa, ejecuta otro código (o continúa con la ejecución del programa).
If-else	Puede haber varios elseif dentro de la misma sentencia if. La primera expresión elseif (si hay alguna) que se evalúe como TRUE sería ejecutada
switch	Se utiliza para agilizar la toma de decisiones múltiples; trabaja de la misma manera que lo harían sucesivos if , if else o until anidados
Sentencias repetitivas	Permiten repetir una sentencia o conjunto de ellas, Las sentencias iterativas son for, while.
for	El bucle for o ciclo for es una estructura de control en programación en la que se puede indicar de antemano el número mínimo de iteraciones que la instrucción realizará.
while	Una expresión que se evalúa antes de cada paso del bucle. Si esta condición se evalúa como verdadera, se ejecuta sentencia. Cuando la condición se evalúa como false, la ejecución continúa con la sentencia posterior al bucle while.
Arreglo	Es una colección de datos del mismo tipo. Sirve para manejar un número "n" de elementos en común, ya sea de tipos definidos por el Lenguaje, ("int", "float", "String", etc...) así como aquellos definidos por el programador.
Matriz	Es un conjunto de arreglos

Anexo B. Símbolos del Diagrama de Flujo

Símbolo	Nombre	Función
	Inicio / Fin	Representa el inicio y el final del proceso
	Entrada	Representa la lectura de datos en entrada
	Salida	Representa la escritura de datos, la impresión de datos en la salida
	Proceso	Representa cualquier tipo de operación interna
	Decisión	Permite analizar una situación
	Conector	Conexión en una misma página
	Conector	Conexión en una página diferente
	Flecha	Representa el flujo del diagrama
	Línea	Puede unir Flechas u otros símbolos.

Al unir los símbolos permite la creación de toma de decisiones, ciclos iterativos y flujo de procesos.

Anexo C. Diagramas de Flujo Sentencias Selectivas y de Repetición

Sentencia

Sintaxis

If Sentencia Condicional

if *expression*

statements

Las instrucciones se ejecutan si la parte real de la expresión tiene todos los elementos distintos de cero. Las partes ELSE y ELSEIF son opcionales.

elseif *expression*

statements

Cero o más partes de ELSEIF se pueden usar, así como anidadas.

else

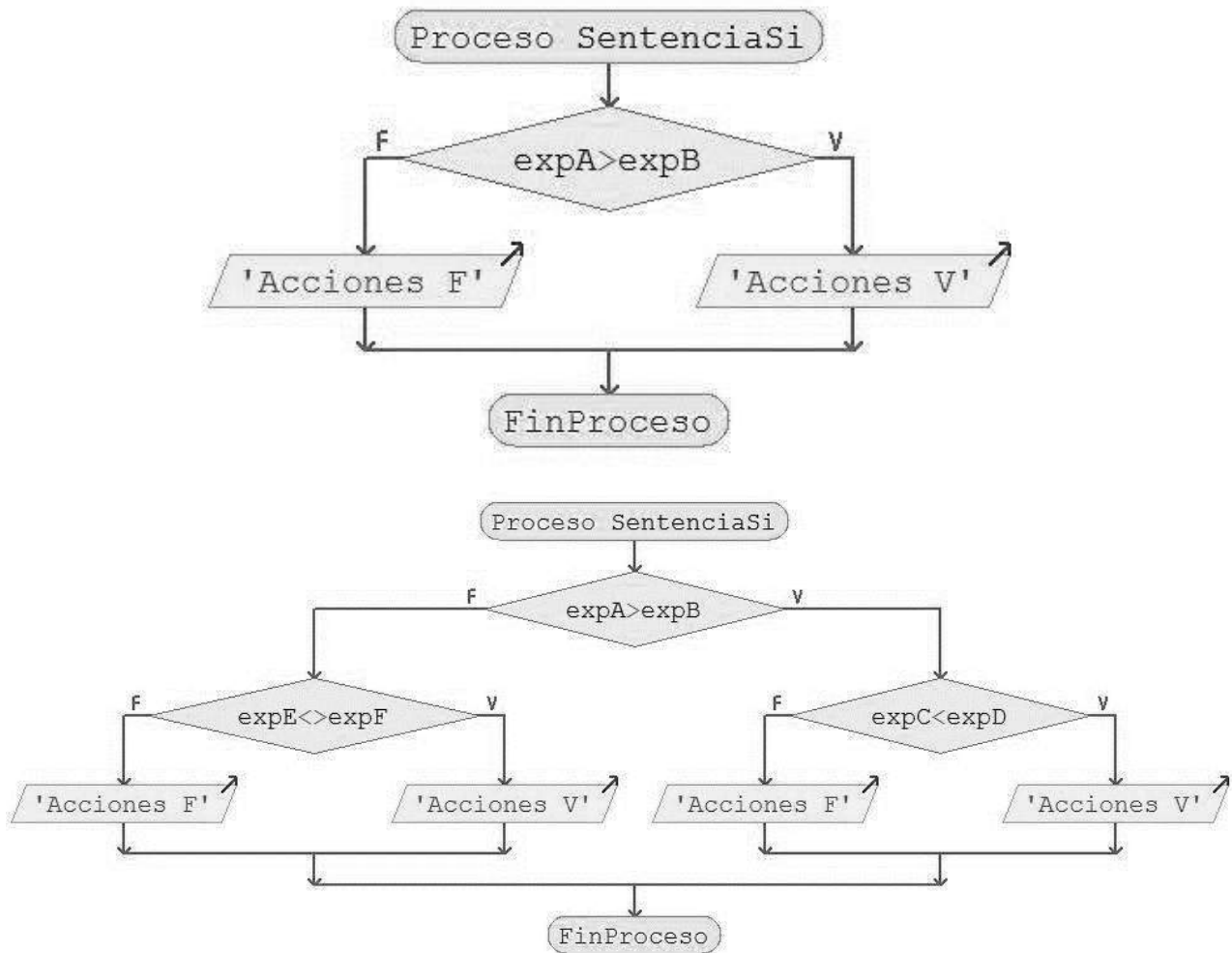
statements

La expresión es usualmente de la forma *exp rop exp* donde:

end

rop es ==, <, >, <=, >=, 0 ~ =.

Diagrama de Flujo



Sentencia

switch Sentencia de Selección.

Entre varios casos basados en la expresión. Las declaraciones que siguen al primer CASE en el que coinciden switch_expr el case_expr se ejecuta.

Cuando la expresión de caso es una matriz de celdas (como en el segundo caso anterior), case_expr coincide si alguno de los los elementos de la matriz de celdas coinciden con la expresión del interruptor.

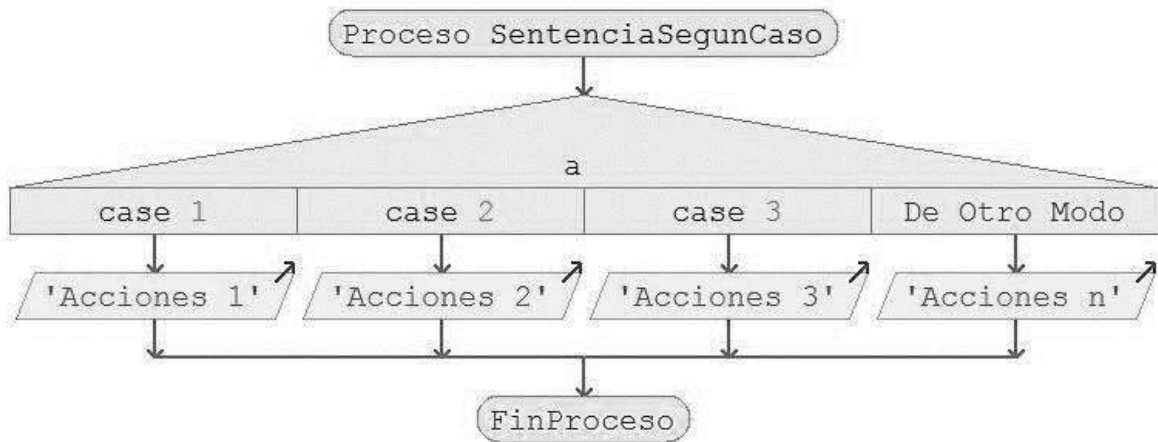
Si nada de las expresiones del caso coinciden con la expresión del cambio y luego el OTHERWISE caso se ejecuta (si existe). Solo se ejecuta un CASE y La ejecución se reanuda con la instrucción después del FIN.

Sintaxis

```

switch switch_expr
  case case_expr,
    statement, ..., statement
  case {case_expr1,
    case_expr2, case_expr3,...}
    statement, ..., statement
  ...
  otherwise,
    statement, ..., statement
END
  
```

Diagrama de Flujo



Sentencia

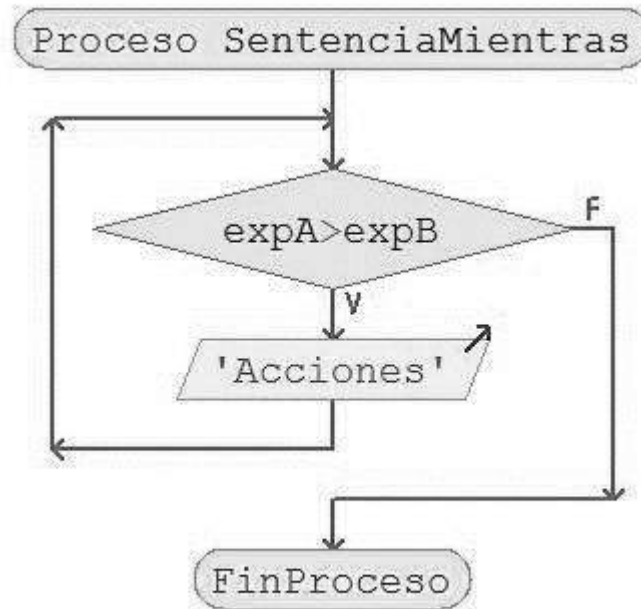
Sintaxis

while

Repita las declaraciones un número indefinido de veces.
Las instrucciones se ejecutan mientras que la parte real de la expresión tiene todos los elementos distintos de cero.
La expresión es generalmente el resultado de `expr rop expr` donde `rop` es `==`, `<`, `>`, `<=`, `>=`, `0 ~ =`.

while expression
statements
end

Diagrama de flujo



Sentencia

for Repite las declaraciones un número específico de veces.

Ejecuta un grupo de sentencias en un ciclo durante un número específico de veces. Los valores tienen una de las siguientes formas:
 initVal: endVal: incremente la variable de índice de initVal a endVal por 1 y repita la ejecución de instrucciones hasta que el índice sea mayor que endVal.

initVal: paso: endVal: aumenta el índice por el paso de valor en cada iteración o disminuye el índice cuando el paso es negativo.

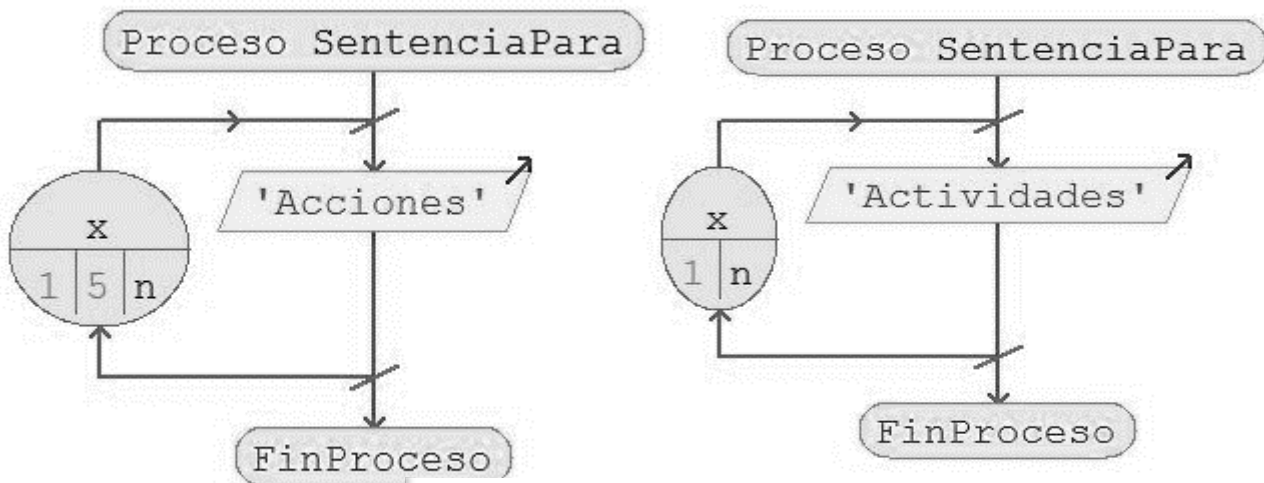
ValArray: cree un vector de columna, índice, de columnas subsiguientes de array valArray en cada iteración. Por ejemplo, en la primera iteración, $index = valArray(:, 1)$. El ciclo ejecuta un máximo de n veces, donde n es el número de columnas de valArray, dado por número ($valArray(1, :)$). La entrada valArray puede ser de cualquier tipo de datos MATLAB®, incluido un vector de caracteres, una matriz de celdas o una estructura.

Sintaxis

for *index* = *values*
statements

end

Diagrama de flujo



Nota. Cuando no se indica incremento, por definición se establece que es 1

ANEXO D Obtener el área de un círculo

Definición de Problema

El área de un Círculo

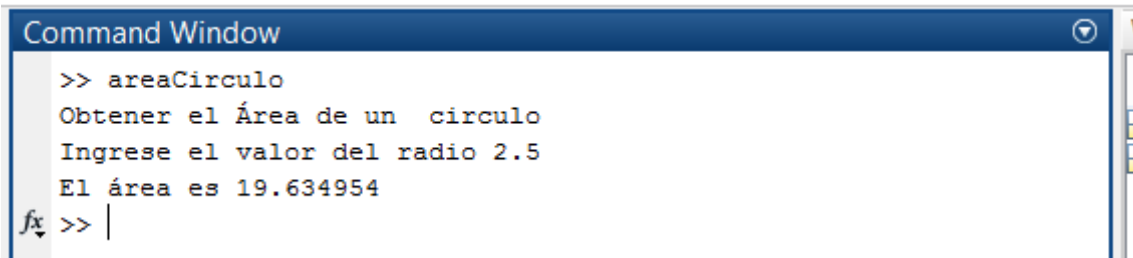
Objetivo

Obtener el área de un círculo usando MATLAB

Codificación a Matlab

En este apartado realiza la Codificación a Código Matlab

```
%Obtener el Área de un círculo  
disp('Obtener el Área de un círculo ');  
radio = input('Ingrese el valor del radio ');  
A=pi*(radio^2);  
fprintf('El área es %f \n', A);
```



```
Command Window  
  
>> areaCirculo  
Obtener el Área de un círculo  
Ingrese el valor del radio 2.5  
El área es 19.634954  
fx >> |
```

El usuario ingresa el valor de radio como 2.5 dando como resultado 19.6349
Se ejecuta correctamente la solución dada al problema planteado.

ANEXO E Obtener Suma de dos números enteros

Definición de Problema

Obtener la suma de dos números enteros

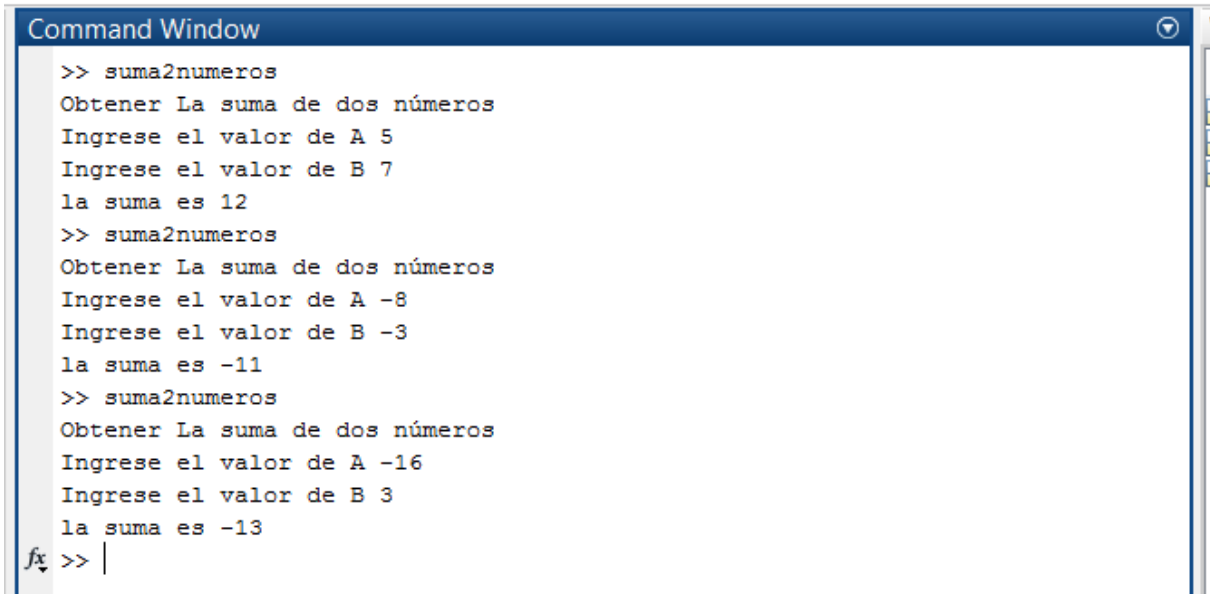
Objetivo

Obtener la suma de dos números enteros usando MATLAB

Codificación a Matlab

En este apartado realiza la Codificación a Código Matlab

```
disp('Obtener La suma de dos números');  
A= input('Ingrese el valor de A ');  
B= input('Ingrese el valor de B ');  
R= A + B;  
fprintf('la suma es %d \n', R);
```



```
Command Window  
>> suma2numeros  
Obtener La suma de dos números  
Ingrese el valor de A 5  
Ingrese el valor de B 7  
la suma es 12  
>> suma2numeros  
Obtener La suma de dos números  
Ingrese el valor de A -8  
Ingrese el valor de B -3  
la suma es -11  
>> suma2numeros  
Obtener La suma de dos números  
Ingrese el valor de A -16  
Ingrese el valor de B 3  
la suma es -13  
fx >> |
```

El usuario ingresa el valor de A y luego el valor de B internamente se realiza la suma y el programa muestra el resultado.



Formato de Reporte de Práctica

Datos Generales		
Docente	Periodo Escolar	Fecha de Elaboración

Ficha Técnica			
Práctica No.		Unidad No.	
Desarrollada por		Nombre de Práctica	
No. Control	Nombre del Alumno		

Definición de Problema
En este apartado redacta el planteamiento del problema
Objetivo General
En este apartado redacta el Objetivo general del problema
Objetivos Específicos
En este apartado redacta los Objetivos Específicos del problema

Análisis de Problema		
En este apartado identifica las variables, genera el algoritmo de solución		
Algoritmo de Solución		
1.-		
2.-		
3.-		
4.-		
5.-		
Datos de Entrada	Proceso	Salida

Prueba de Escritorio							
En este apartado realiza la prueba de escritorio de al menos tres ejemplos con el algoritmo que desarrollo							
Variables identificadas						Visualiza el usuario	



Resultados

En este apartado se realiza la codificación del algoritmo anterior

Pseudocódigo Solución

--

Resultados

En este apartado realiza el diagrama de flujo correspondiente

Diagrama de Flujo

Resultados

En este apartado realiza la codificación del algoritmo planteado a Matlab

Codificación

Resultados

En este apartado realiza la explicación de la ejecución en Matlab

Pantallas Resultado Explicadas

--

Resultados

En este apartado redacta cada alumno las conclusiones a las que llego

Conclusiones

--

Bibliografía En este apartado redacta bibliografía en formato APA

Anexos En este apartado redacta e ilustra aquello adicional de ayuda a la elaboración de tu práctica.



NOTAS