



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA ACADÉMICA
DIRECCIÓN DE EDUCACIÓN MEDIA SUPERIOR
CENTRO DE ESTUDIOS CIENTÍFICOS Y TECNOLÓGICOS No. 13
"RICARDO FLORES MAGÓN"



GUÍA

**de estudio para
presentar ETS**

UNIDAD DE APRENDIZAJE
Herramientas de programación

Semestre: 3
Ciclo escolar: 2022-2023 B

Área: tecnológica	Nombre de la Unidad de Aprendizaje: Herramientas de programación	Nivel/semestre: Medio/tercero
------------------------------------	---	--

1.- Integrantes de Academia:

No	Docente
1.	Alejandro Morales Zavaleta
2.	Alberto Torres Santander

2.- Introducción

La escritura de programas de computadora o simplemente programación implica que la persona que decida involucrarse en esta actividad debería estar capacitado para realizar lo siguiente:
Diseño de algoritmos para la solución de problemas computacionales a través de las metodologías, procedimientos y herramientas de programación.

Las habilidades y actitudes concretas que debería desarrollar serían:

- Capacidad de análisis
- Dominio de las herramientas de programación
- Creativo e innovador
- Puntualidad y Responsabilidad
- Liderazgo
- Hábitos personales.
- Interés por el estudio
- Capacidad de generar y transmitir el conocimiento.
- Capacidad para motivar y propiciar el trabajo colaborativo

3.- Objetivos.

El propósito principal es preparar al estudiante para que desarrolle competencias en el manejo de las Herramientas de Programación que le permita solucionar problemas computacionales, ofreciéndole conocimientos básicos que favorecen su formación académica sirviendo como base para las unidades de aprendizaje relacionadas con el desarrollo de software y apoyando su formación como Técnico en Informática; y su posterior incorporación en estudios de Nivel Superior y/o al campo laboral.

4.- Justificación.

Las competencias profesionales (general y particulares) implican como principales objetos de conocimiento el diseño de algoritmos para solucionar problemas computacionales con el uso de las metodologías, procedimientos y herramientas de programación, mismos que podrá vincular con su entorno socioeconómico y laboral. Asimismo, en la particularidad el estudiante:

- Comprueba los criterios de solución y las heurísticas básicas con base en el planteamiento de problemas de diversa naturaleza.
- Diseña algoritmos para la solución de problemas con base en criterios estructurados.
- Diseña algoritmos a través de herramientas de programación.
- Programa aplicaciones sencillas a través de un entorno de programación estructurada.

Las habilidades y actitudes concretas que desarrolla el estudiante:

- Capacidad de análisis
- Dominio de las herramientas de programación
- Creativo e innovador
- Puntualidad y Responsabilidad
- Liderazgo
- Hábitos personales.
- Interés por el estudio
- Capacidad de generar y transmitir el conocimiento.
- Capacidad para motivar y propiciar el trabajo colaborativo

Asimismo, los principales objetos de conocimiento que se adquirirán y serán cuerpo de las acciones o desempeños a realizar son:

- Diseñar algoritmos computacionales para la solución de problemas con base a criterios estructurados
- Diseñar algoritmos con base en las herramientas de programación
- Programará aplicaciones básicas a través de un entorno de programación estructurada

5.- Estructura y contenidos

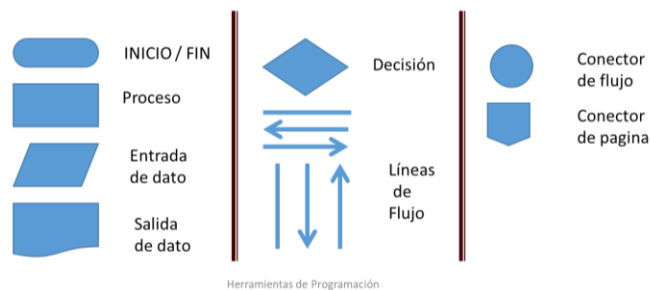
Estructura y contenidos	6.- Materiales para la elaboración de la guía
<p>Unidad I.- Algoritmos computacionales</p> <p>RAP 1: Soluciona problemas con base en métodos y heurísticas básicas</p> <p>RAP 2: Redacta algoritmos cualitativos para la solución de problemas mediante criterios estructurados</p> <p>RAP 3: Aplica algoritmos cuantitativos para la solución de problemas a través de criterios estructurados</p>	
<p>Unidad II.- Herramientas de programación</p> <p>RAP 1: Crea algoritmos con base en diagramas de flujo</p> <p>RAP 2: Crea algoritmos por medio de pseudocódigo</p>	<ul style="list-style-type: none"> • Software para diagramación como Visio o DIA o DRAW IO o se pueden usar las herramientas de dibujo de Office • Para el caso de DRAW IO se puede descargar del siguiente link <ul style="list-style-type: none"> ○ https://github.com/jgraph/drawio-desktop/releases/download/v20.3.0/draw.io-20.3.0-windows-no-installer.exe • Usar el software PSEINT, se puede descargar del siguiente link <ul style="list-style-type: none"> ○ http://prdownloads.sourceforge.net/pseint/pseint-w32-20210609.zip?download
<p>Unidad III.- Principios de programación</p> <p>RAP 1: Explica los lenguajes de programación con base en su taxonomía y funcionamiento</p> <p>RAP 2: Emplea las sentencias básicas de un lenguaje a través de un entorno de programación estructurada.</p> <p>RAP 3: Estructura aplicaciones básicas a través del planteamiento de un problema.</p>	<p>Para la escritura de programas usar el IDE CodeBlocks, se puede descargar del siguiente link:</p> <p>http://sourceforge.net/projects/codeblocks/files/Binaries/20.03/Windows/codeblocks-20.03mingw-setup.exe</p> <p>Nota: Esta aplicación es solo para sistemas con Windows de 64 bits, en caso de otros sistemas consultar a su profesor.</p>

7.- Actividades de estudio.

A. Conocimiento de terminología básica de cada una de las unidades del programa.

- Algoritmo. Conjunto de pasos, procedimientos o acciones que nos permiten resolver un problema o alcanzar un resultado.
2. Características de un algoritmo:
 - Finito
 - Preciso
 - Efectivo
 3. Partes de un algoritmo
 - Cero o más entradas
 - Procesos
 - 1 o más salidas
 4. Diagramas de flujo
 - Representación gráfica de un algoritmo.
 - Muestra el flujo de los pasos del algoritmo usando un símbolo para representar cada operación o paso del algoritmo.
 5. Simbología de un Diagrama de Flujo

✓ Los símbolos típicamente utilizados en los diagramas de flujo para programación son los siguientes:



6. Reglas para uso de un Diagrama de Flujo

- El sentido de un diagrama de flujo generalmente es de arriba hacia abajo y de izquierda a derecha.
- Los símbolos se unen con líneas, las cuales tienen en la punta una flecha que indica la dirección de flujo.

- Las líneas de flujo deben de entrar a un símbolo por la parte superior y salir de él por la parte inferior.
- Se deben de utilizar solamente líneas de flujo horizontales o verticales (nunca diagonales o de otro tipo).
- De un símbolo excepto el de decisión, solo puede salir una línea de flujo.
- En un símbolo solo puede entrar una flecha de flujo si varias líneas se dirigen al mismo símbolo, se deben unir en una sola flecha.

7. Seudocódigo con el software PseInt

- Es una imitación abreviada de instrucciones reales para las computadoras.
- Permite definir las estructuras de datos, las operaciones que se aplicarán a los datos y la lógica que tendrá el programa de computadora para solucionar un determinado problema.
- Similar al lenguaje natural
- Respetar las directrices y los elementos de los lenguajes de programación.

Tipo de Instrucciones	Instrucción a utilizar
Inicio y fin del programa	Inicio Termina
Instrucción de entrada	Leer
Instrucción de salida	Escribe
Operadores aritméticos	+, -, *, /, mod (cálculo de residuo), div (división entera), sqrt (raíz cuadrada), ** (cálculo de potencias)
Operadores lógicos	Y O No
Operadores relacionales	<, >, <=, >=, =
Asignación	←
Concatenación	+

- Estructura general de un algoritmo representado con Pseudocódigo

ALGORITMO NombreAlgoritmo

Declaración de variables

Sentencias del algoritmo

FINALGORITMO

8. Tipos de datos.

- Representación simbólica, atributo o característica de una entidad.
Por ejemplo
 - Rojo
 - 25 años
 - Café

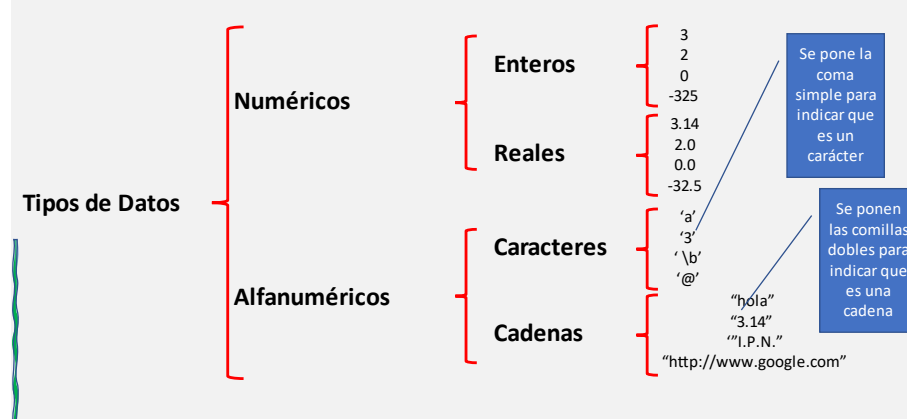
- Delgado
- 9.8
- En programación es el elemento sobre el cual opera un algoritmo



Tipos de datos fundamentales



Ejemplos de los tipos de datos



9. Constantes. Es un dato cuyo valor se mantiene fijo mientras se realiza un algoritmo. Se le asocia el valor del símbolo que representa.

10. Variables.


- Son datos en los cuales el valor puede cambiar al realizar el algoritmo
- Una variable se representa con un símbolo alfanumérico
- En matemáticas se le conoce como literal
- Se acostumbra a representar con una letra, las más común es x
- En programación se le llaman identificadores
- Se le asocia un nombre descriptivo que indique para que se usa el dato variable
- Cuando se “crea” una variable en un programa, la computadora reserva espacio en la memoria de la computadora para almacenar el valor que pueda tomar la variable.

11. Expresiones aritméticas y lógicas

Una expresión lineal se evalúa de acuerdo con las siguientes reglas.

1. Realizar primero las operaciones de mayor jerarquía. Ver tabla de jerarquías
2. Si hay varios operadores de la misma jerarquía hacerlas de izquierda a derecha
3. Si existen paréntesis se procede de acuerdo con lo siguiente.
4. Si hay más de un nivel de paréntesis se comienza a resolver desde el nivel más profundo hacia arriba, aplicando en cada nivel las reglas 1 y 2.
5. Si los paréntesis están al mismo nivel se procede según lo indicado en la segunda regla y se resuelve las operaciones de acuerdo con la regla 1 y 2.
6. Si hay un solo nivel se aplican las reglas 1 y 2.

12. Jerarquía de operadores.

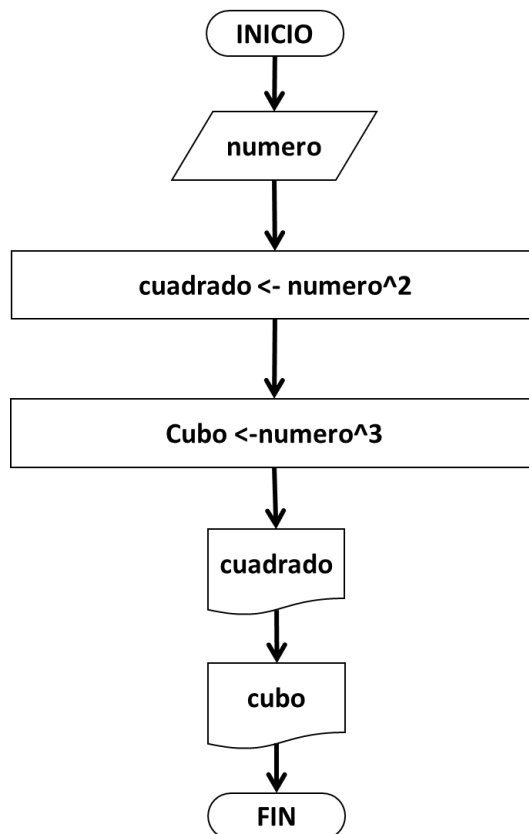
Operación	Símbolo	Jerarquía	
Agrupación y funciones	()	Mayor	
Potencia	^		
Multiplicación, división, división entera, residuo	* / DIV MOD		
Suma, resta	+ -		
Asignación	<-	Menor	

13. Prueba de escritorio.

- Las pruebas de escritorio son simulaciones del comportamiento de un algoritmo que permiten determinar la validez de este. Consisten en generar una tabla con tantas columnas como variables tenga el algoritmo y seguir las instrucciones poniendo los valores correspondientes

14. Representación de resolución de un problema en Diagrama de Flujo

Diseñar un programa que calcule el cuadrado y el cubo de un numero



15. Representación de resolución de un problema en PseInt

Diseñar un programa que calcule el cuadrado y el cubo de un numero

```
ALGORITMO Potencias  
  DEFINIR numero, cuadrado,cubo COMO REAL  
  LEER numero  
  cuadrado<- numero^2  
  cuadrado<- numero^3  
  ESCRIBIR cuadrado  
  ESCRIBIR cubo  
FINALGORITMO
```

16. Representación de resolución de un problema en el lenguaje ANSI C

Diseñar un programa que calcule el cuadrado y el cubo de un numero

Respuesta

```
/*  
    Biblioteca de funciones  
*/  
#include<stdio.h>  
#include<math.h>  
/*  
    Cuando se llama el programa se realiza el código  
    incluido en la sección o función main  
*/  
int main() {  
    /*  
        Declaración de variables  
        En ANSI C es necesario indicar el tipo de dato y el  
        identificador de la variable  
    */  
    float cuadrado;  
    float cubo;  
    float numero;  
    /*  
        Entrada de datos  
        Se ponen mensajes informativos para que el usuario  
        tenga cierta idea de la que se espera que haga  
    */  
    printf("Introduzca un número\n",cubo);  
    scanf("%f",&numero);  
  
    /*  
        Cálculos  
        En ANSI C no existe el operador ^  
        por lo que se usa la función pow, esta función
```

```
        esta en la biblioteca math.h
    */
    cuadrado = pow(numero,2);
    cuadrado = pow(numero,3);
    /*
        Salida de datos
        Se ponen mensajes informativos para que el usuario
comprenda
        lo que muestra el programa
    */
    printf("El cuadrado del número %f es %f\n",numero,cuadrado);
    printf("El cubo del número %f es %f\n",numero,cuadrado);
    return 0;
}
```

B. Para los siguientes problemas:

1. Realizar el análisis.
2. Diseñar su algoritmo en Diagrama de flujo, pseudocódigo (con la sintaxis de PseInt)
3. Probar su diseño haciendo la prueba de escritorio
4. Desarrollar su programa en lenguaje ANSI C.

1. Escriba un programa que calcule el valor del I.V.A. de una cantidad dada. El programa deberá desplegar en pantalla la cantidad original, la cantidad de I.V.A y la cantidad total.

Respuesta

Análisis

Datos de Salida

Cantidad
Cantidad de IVA
Cantidad total

Datos de Entrada.

Cantidad

Otros datos

El IVA es 16% por de la cantidad

Condiciones.

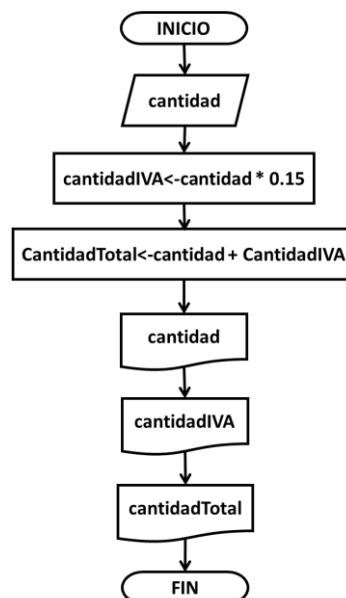
Cantidad mayor que cero

Algoritmo en lenguaje natural o cualitativo.

1. Conocer la cantidad
2. Realizar la siguiente operación para conocer la cantidad de IVA
 $\text{Cantidad IVA} = \text{Cantidad} * 0.16$
3. Para conocer la cantidad total hacer lo siguiente
 $\text{Cantidad total} = \text{cantidad} + \text{cantidad IVA}$
4. Fin

Diseño

Diagrama



Seudocódigo

Algoritmo IVA

```
leer cantidad
cantidadIVA <- cantidad * 0.16
CantidadTotal <- cantidad + cantidadIVA
Escribir cantidad
Escribir cantidadIVA
Escribir CantidadTotal
```

FinAlgoritmo

Prueba de escritorio

Entrada	Procesos		Salidas		
Cantidad	$\text{cantidadIVA} \leftarrow \text{cantidad} * 0.16$	$\text{CantidadTotal} \leftarrow \text{cantidad} + \text{cantidadIVA}$	Cantidad	CantidadIVA	CantidadTotal
100	$\text{cantidadIVA} \leftarrow 100 * 0.16$ $\text{cantidadIVA} \leftarrow 16$	$\text{CantidadTotal} \leftarrow 100 + 16$	100	16	116

Programa

```
#include<stdio.h>
```

```
int main() {
    float cantidad;
    float cantidadiva;
    float cantidadtotal;
    printf("Introducir cantidad: ");
    scanf("%f",&cantidad);
    cantidadiva = cantidad*0.16;
    cantidadtotal = cantidad+cantidadiva;
    printf("Cantidad: %f\n",cantidad);
    printf("Monto IVA: %f\n",cantidadiva);
    printf("TOTAL (Cantidad + IVA): %f\n",cantidadtotal);
    return (0);
}
```

2. Haga un programa para calcular el perímetro, área y volumen de un cilindro.

Respuesta

Análisis

Datos de Salida

Área y volumen de un cilindro

Datos de Entrada.

Radio y altura

Otros datos

Poner valor de PI en 3.1416

Condiciones.

Un cilindro no tiene perímetro.

El área se calcula con la fórmula $A = 2\pi r(h+r)$

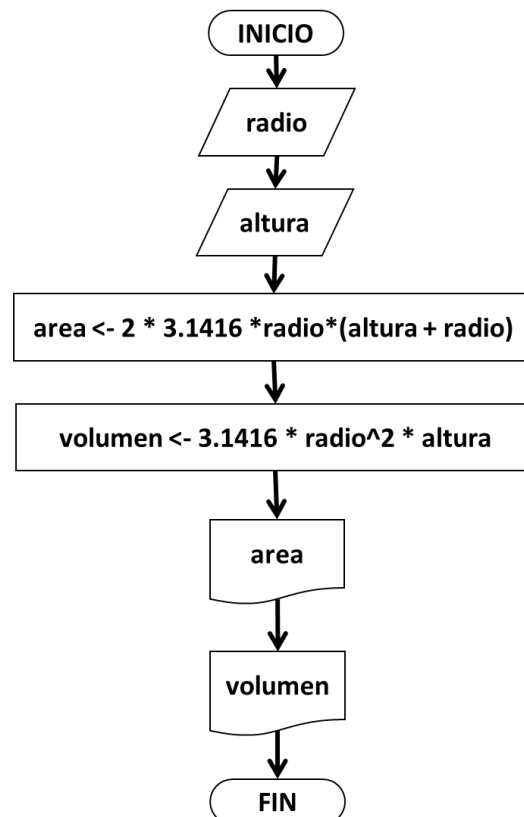
El volumen se calcula con la fórmula $V = \pi r^2 h$

Algoritmo en lenguaje natural o cualitativo.

1. Obtener los valores del radio y la altura
2. Calcular el área con
 $A = 2\pi r(h+r)$
3. Calcular el volumen con
 $V = \pi r^2 h$
4. Mostrar el área y el volumen calculados
5. Fin

Diseño

Diagrama



Seudocódigo

Algoritmo cilindro

Definir area,volumen,radio,altura Como Real

leer radio

leer altura

area <- $2 * 3.1416 * \text{radio} * (\text{radio} + \text{altura})$

volumen <- $3.1416 * \text{radio}^2 * \text{altura}$

Escribir area

Escribir volumen

FinAlgoritmo

Prueba de escritorio

Entrada		Procesos		Salidas	
radio	altura	area <- $2 * 3.1416 * \text{radio} * (\text{radio} + \text{altura})$	volumen <- $3.1416 * \text{radio}^2 * \text{altura}$	area	volumen
1	1	area <- $2 * 3.1416 * 1 * (1 + 1)$ area <- $2 * 3.1416 * 1 * (2)$ area <- $2 * 3.1416 * 1 * (2)$ area <- 12.5664	volumen <- $3.1416 * 1^2 * 1$ volumen <- $3.1416 * 1 * 1$ volumen <- 3.1416	12.5664	3.1416

Programa

```
#include<stdio.h>
```

```
#include<math.h>
```

```
int main() {
```

```
    float altura;
```

```
    float area;
```

```
    float radio;
```

```
    float volumen;
```

```
    printf("Introducir radio de la base del cilindro");
```

```
    scanf("%f",&radio);
```

```
    printf("Introducir altura del cilindro");
```

```
    scanf("%f",&altura);
```

```
    area =  $2 * 3.1416 * \text{radio} * (\text{radio} + \text{altura})$ ;
```

```
    volumen =  $3.1416 * \text{pow}(\text{radio}, 2) * \text{altura}$ ;
```

```
    printf("El area del cilindro es %f\n",area);
```

```
    printf("La altura del cilindro es %f\n",volumen);
```

```
    return 0;
```

```
}
```

3. Proporcionar el precio promedio de un producto, calculado a partir del precio de este en tres establecimientos distintos. Los precios para el cálculo son introducidos por el usuario.

Respuesta

Análisis

Datos de Salida

Precio promedio

Datos de Entrada.

3 precios

Otros datos

Sin datos iniciales

Condiciones.

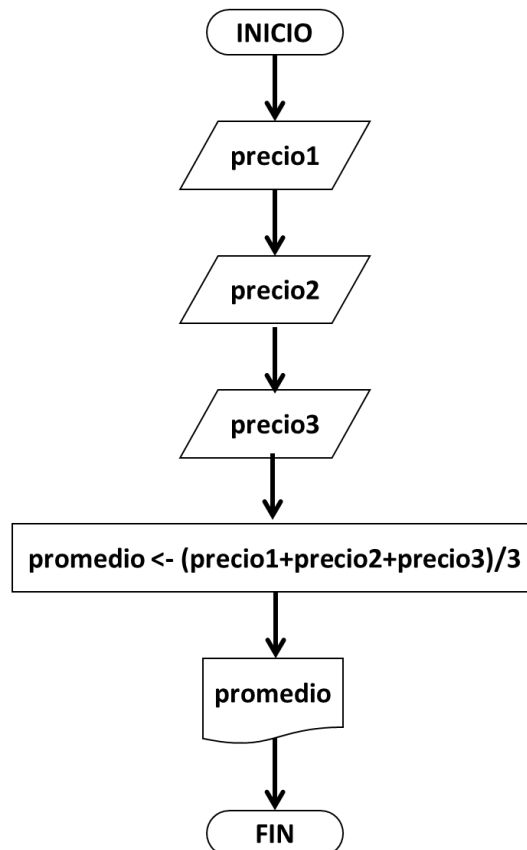
Los precios son del mismo producto de diferentes establecimientos

Algoritmo en lenguaje natural o cualitativo.

1. Introducir precio 1, precio 2 y precio 3
2. Calcular el promedio de la siguiente manera.
$$\text{Promedio} = (\text{precio 1} + \text{precio 2} + \text{precio 3}) / 3$$
3. Mostrar el promedio
4. Fin

Diseño

Diagrama



Seudocódigo

```
ALGORITMO Promedio_precios
    DEFINIR precio1,precio2,precio3,promedio COMO REAL
    LEER precio1
    LEER precio2
    LEER precio3
    promedio<-(precio1+precio2+precio3)/3
    ESCRIBIR promedio
FINALGORITMO
```

Prueba de escritorio

Entrada			Procesos	Salidas
Precio1	Precio2	Precio3	promedio<-(precio1+precio2+precio3)/3	promedio
100	129	125	promedio<-(100+129+125)/3	118

Programa

```
#include <stdio.h>
#include <stdlib.h>
// Ejercicios resueltos aprenderaprogramar.com
int main() {
    double precio1, precio2, precio3, promedio;
    // Datos
    printf ("Introduzca el precio en establecimiento 1, en pesos: ");
    scanf ("%f", &precio1);
    printf ("Introduzca el precio en establecimiento 2, en pesos: ");
    scanf ("%f", &precio2);
    printf ("Introduzca el precio en establecimiento 3, en pesos: ");
    scanf ("%f", &precio3);
    //Cálculo
    promedio = (precio1 + precio2 + precio3) / 3;
    printf ("El precio medio del producto es de %f pesos", media);
    return 0;
}
```

4. Escriba el pseudocódigo para calcular la distancia entre dos puntos dados P1 y P2.

Desarrollar el ejercicio, siguiendo la metodología propuesta.

Análisis

Datos de Salida

Datos de Entrada.

Otros datos

Condiciones.

Algoritmo en lenguaje natural o cualitativo.

Diseño

Diagrama

Seudocódigo

Prueba de escritorio

Programa en lenguaje ANSI C.

5. Dado como dato el sueldo de un trabajador, considere un aumento del 15% si su sueldo es inferior a \$1000 y de un 12% en caso contrario. Imprima el sueldo con el aumento incorporado.

Respuesta

Análisis

Datos de Salida

Sueldo final

Datos de Entrada.

Sueldo

Otros datos

Sin otros datos

Condiciones.

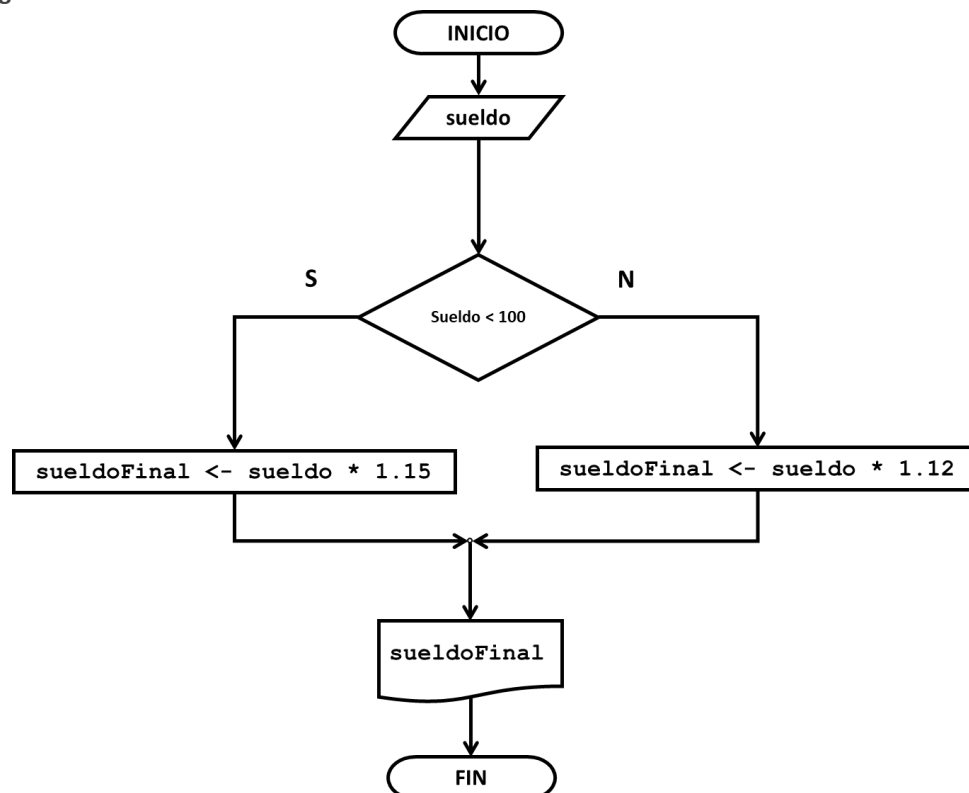
Si el sueldo es menor que 1000 aumentar 15% en caso contrario aumentar 12%

Algoritmo en lenguaje natural o cualitativo.

1. Obtener el sueldo
2. Si el sueldo es menor a 1000 aumentarle el 15%, hacer la siguiente operación para conocer el sueldo final
 $\text{Sueldo final} \leftarrow \text{sueldo} * 1.15$
Si no aumentar el 12%, hacer la siguiente operación para conocer el sueldo final
 $\text{Sueldo final} \leftarrow \text{sueldo} * 1.12$
3. Mostrar el sueldo final
4. Fin

Diseño

Diagrama



Seudocódigo

```
Algoritmo Aumento_de_sueldo
    DEFINIR sueldo, sueldofinal COMO REAL
    LEER sueldo
    SI sueldo < 1000 ENTONCES
        sueldofinal<-sueldo*1.15
    Sino
        sueldofinal<-sueldo*1.12
    FinSi
    Escribir sueldofinal
FinAlgoritmo
```

Prueba de escritorio

Entrada	Procesos			Salidas
Sueldo	Sueldo <100	sueldofinal<- sueldo*1.15	sueldofinal<-sueldo*1.12	SueldoFinal
500	500 <100 Si	sueldofinal<-500*1.15 sueldofinal<-575		575
2000	2000 <100 Si	sueldofinal<-500*1.15 sueldofinal<-575	sueldofinal<-2000*1.12 sueldofinal<-2240	2240

Programa

```
#include<stdio.h>

int main() {
    float sueldo;
    float sueldofinal;
    scanf("%f",&sueldo);
    if (sueldo<1000) {
        sueldofinal = sueldo*1.15;
    } else {
        sueldofinal = sueldo*1.12;
    }
    printf("%f\n",sueldofinal);
    return 0;
}
```

6. Escriba el pseudocódigo, que dados como datos la matrícula (o boleta) y 5 calificaciones de un alumno, imprima la matrícula, el promedio y la palabra “aprobado” si el alumno tiene un promedio mayor o igual que 6, y la palabra “no aprobado” en caso contrario.

Respuesta

Análisis

Datos de Salida

Matricula

Promedio

Reprobado según la condición

Datos de Entrada.

Matricula

5 calificaciones

Otros datos

Sin Otros datos

Condiciones.

Alumno reprobado si el promedio es menor a 6

Algoritmo en lenguaje natural o cualitativo.

1. Obtener la matrícula

2. Obtener las 5 calificaciones

3. Calcular el promedio de la siguiente manera

$$\text{Promedio} = (\text{suma de las 5 calificaciones})/5$$

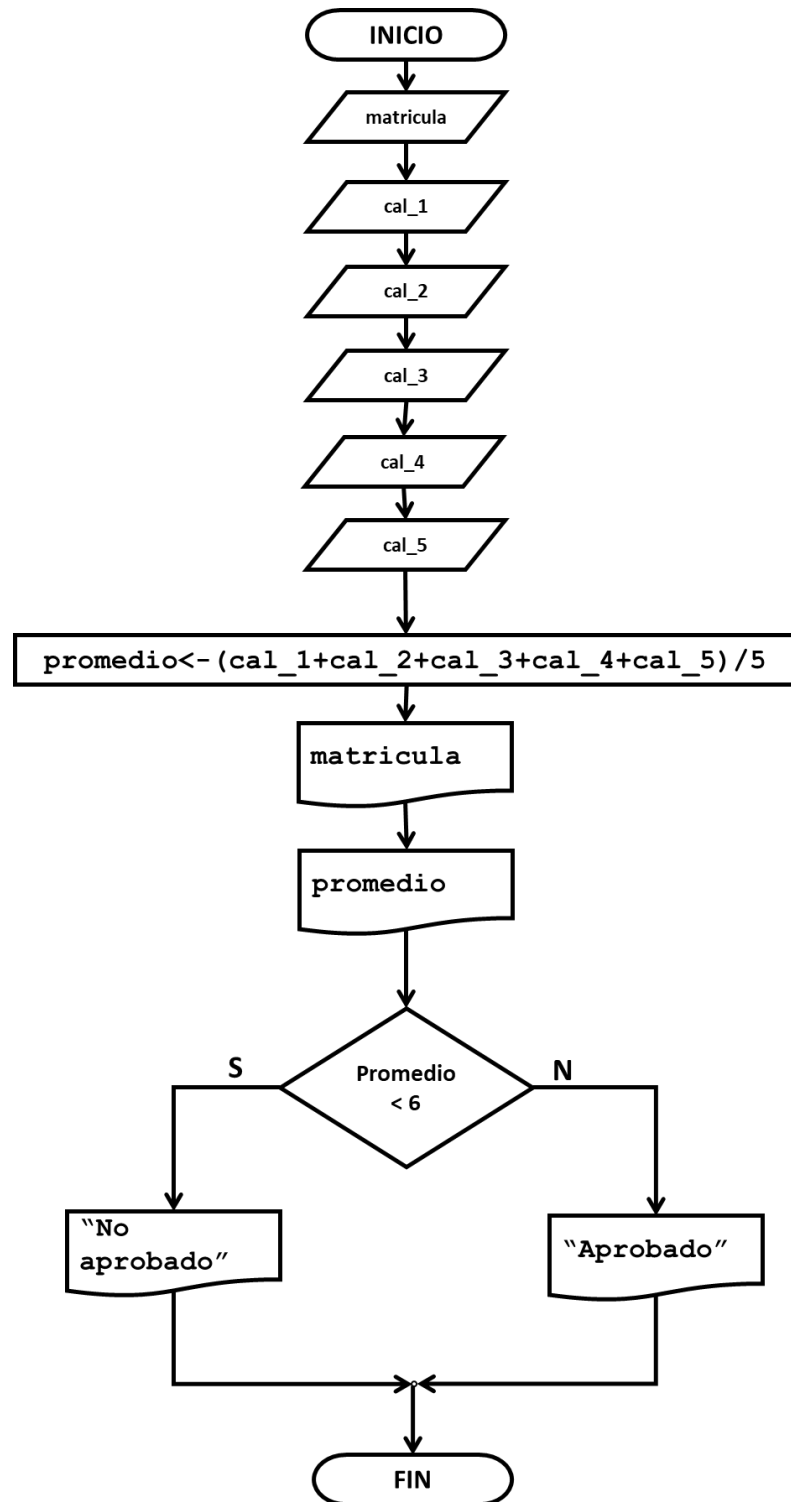
4. Mostrar la matrícula

5. Mostrar el promedio

6. Si el promedio es menor a 6, Mostrar no aprobado, si es igual a 6 o mayor mostrar aprobado

7. Fin

Diseño
Diagrama



Seudocódigo

ALGORITMO Calificaciones

DEFINIR matricula,promedio,cal_1,cal_2, cal_3, cal_4,cal_5 COMO REAL

LEER matricula

LEER cal_1

LEER cal_2

LEER cal_3

LEER cal_4

LEER cal_5

promedio<- (cal_1+cal_2+cal_3+cal_4+cal_5)/5

ESCRIBIR matricula

ESCRIBIR promedio

SI promedio < 6 ENTONCES

 ESCRIBIR "No aprobado"

SiNo

 ESCRIBIR "Aprobado"

FINSI

FINALGORITMO

Prueba de escritorio

Entrada						Procesos		Salidas			
Matricula	Cal_1	Cal_2	Cal_3	Cal_4	Cal_5	promedio<- (cal_1+cal_2+cal_3+cal_4+cal_5)/5	promedio < 6	matricula	promedio	No aprobado	Aprobado
10001	5	6	6	5	5	promedio<- (5+6+6+5+5)/5 promedio<- (27)/5 promedio<- 5.4	5.4 <6 S	10001	5.4	No aprobado	
10002	7	7	8	8	8	promedio<- (7+7+8+8+8)/5 promedio<- (38)/5 promedio<- 7.6	7.6 <6 N	10002	7.6		Aprobado

Programa

```
#include<stdio.h>
```

```
int main() {  
    float cal_1;  
    float cal_2;  
    float cal_3;  
    float cal_4;  
    float cal_5;  
    float matricula;  
    float promedio;  
    scanf("%f",&matricula);  
    scanf("%f",&cal_1);  
    scanf("%f",&cal_2);  
    scanf("%f",&cal_3);
```

```
scanf("%f",&cal_4);
scanf("%f",&cal_5);
promedio = (cal_1+cal_2+cal_3+cal_4+cal_5)/5;
printf("%f\n",matricula);
printf("%f\n",promedio);
if (promedio<6) {
    printf("No aprobado\n");
} else {
    printf("Aprobado\n");
}
return 0;
}
```


7. Construya un diagrama de flujo que le permita calcular la tangente de un ángulo, considerado que se conoce el valor del seno y del coseno de este.

Respuesta

Análisis

Datos de Salida

Tangente de un ángulo

Datos de Entrada.

Angulo

Otros datos

Seno y coseno del ángulo

$\text{Tangente ángulo} = \text{seno ángulo} / \text{coseno ángulo}$

Condiciones.

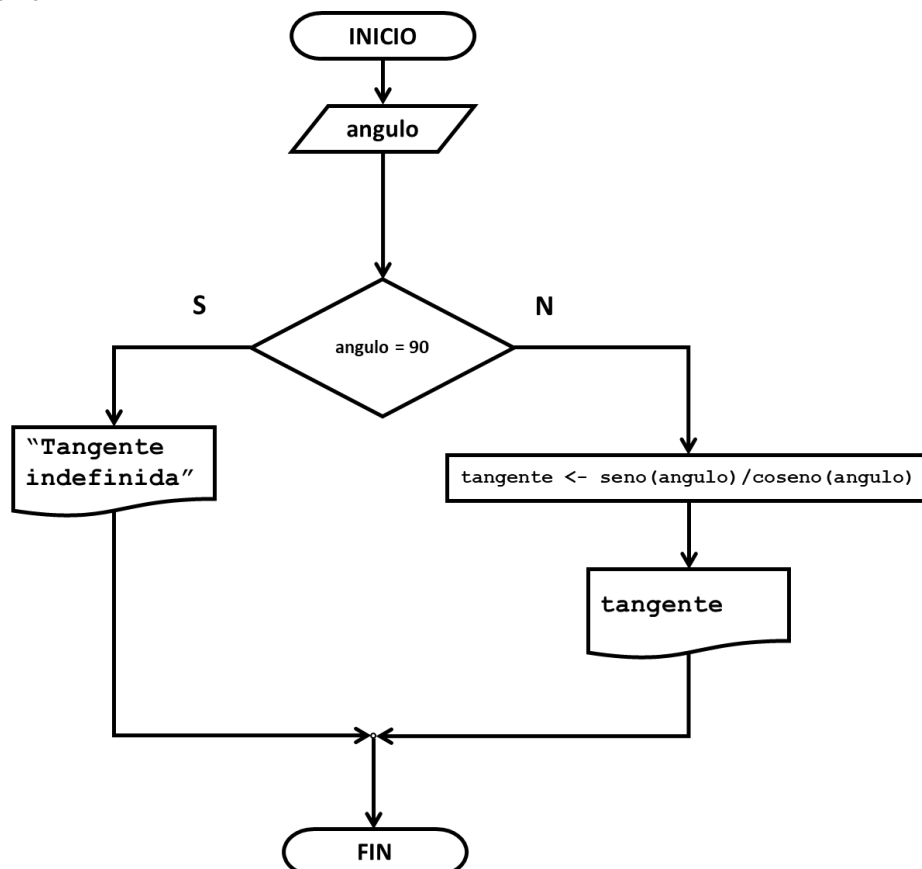
Para el caso de ángulos de 90° la tangente no está definida

Algoritmo en lenguaje natural o cualitativo.

1. Capturar valor del Angulo en grados
2. Si el Angulo es de 90 grados enviar un error y terminar si no ir al paso 3
3. La tangente se calcula de la siguiente forma
 $\text{Tangente} = \text{seno} / \text{coseno}$
4. Mostrar valor de la tangente
5. Fin

Diseño

Diagrama



Seudocódigo

```
Algoritmo Calculatangente
    definir tangente,angulo Como Real
    leer angulo
    si angulo = 90 Entonces
        Escribir "tangente indefinida"
    sino
        tangente <- sen(angulo)/cos(angulo)
        Escribir tangente
    FinSi
FinAlgoritmo
```

Prueba de escritorio

Entrada	Procesos		Salidas	
angulo	Angulo = 90	Tangente <- seno(angulo)/coseno(angulo)	Tangente	"tangente indefinida"
90	90=90 S			"tangente indefinida"
45	45 = 90 N	Tangente <- seno(45)/coseno(45) Tangente <- 0.7/0.7 Tangente <-1	1	

Programa

```
#include<stdio.h>
#include<math.h>
int main() {
    float angulo;
    float pi=3.1416;
    float radianes;
    float tangente;
    scanf("%f",&angulo);
    //En lenguaje las funciones sin y cos usan ángulos en radianes por lo que hay que
    convertir
    //Grados a radianes
    radianes = (angulo*pi)/180;
    if (angulo==90) {
        printf("tangente indefinida\n");
    } else {
        tangente = sin(radianes)/cos(radianes);
        printf("%f\n",tangente);
    }
    return 0;
}
```

8. Construya un diagrama de flujo tal que, dados como datos de entrada tres números enteros, determine si los mismos están en orden creciente.

Desarrollar el ejercicio, siguiendo la metodología propuesta.

Análisis

Datos de Salida

Datos de Entrada.

Otros datos

Condiciones.

Algoritmo en lenguaje natural o cualitativo.

Diseño

Diagrama

Seudocódigo

Prueba de escritorio

Programa en lenguaje ANSI C.

9. Construya un diagrama de flujo tal que, dados como datos, la matrícula de un alumno, la carrera en la que está inscrito su semestre y su promedio; determine si el mismo es apto para pertenecer a alguna de las facultades menores que tiene la universidad. Si el alumno es aceptado teniendo en cuenta las especificaciones que se listan abajo, se debe imprimir su matrícula, carrera y la palabra aceptado:

Especificaciones para pertenecer a las facultades menores:

Economía: Desde sexto semestre y promedio mayor o igual a 8.8

Computación: Desde sexto semestre y promedio mayor a 8.5

Administración: Desde quinto semestre y promedio mayor a 8.5

Contabilidad: Desde quinto semestre y promedio mayor a 8.0

Respuesta

Análisis

Datos de Salida

Facultad

Datos de Entrada.

Matricula, carrera, semestre, promedio

Otros datos

Sin Otros datos

Condiciones.

Las especificaciones para pertenecer a una facultad menor son:

Para Economía: Desde sexto semestre y promedio mayor o igual a 8.8

Para Computación: Desde sexto semestre y promedio mayor a 8.5

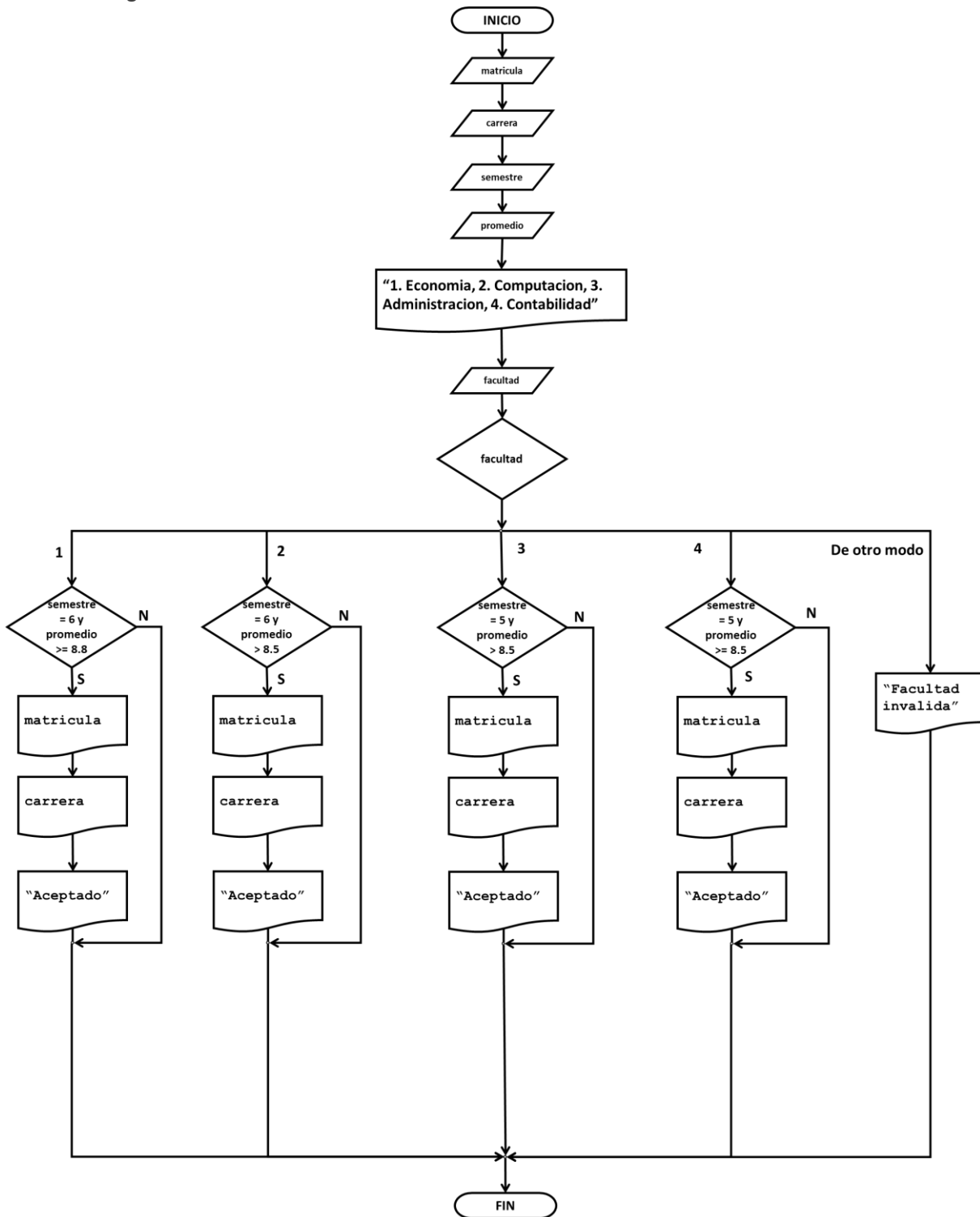
Para Administración: Desde quinto semestre y promedio mayor a 8.5

Para Contabilidad: Desde quinto semestre y promedio mayor a 8.5

Algoritmo en lenguaje natural o cualitativo.

1. Introducir matricula, carrera, semestre, promedio
2. Preguntar que facultad desea
3. Para Economía debe cumplir que promedio ≥ 8.8 y semestre es sexto
4. Para Computación debe cumplir que promedio > 8.5 y semestres es sexto
5. Para Administración debe cumplir que Si promedio > 8.5 y semestre es quinto
6. Para Contabilidad debe cumplir que Si promedio > 8.5 y semestre es quinto.
7. Fin

Diseño
Diagrama



Seudocódigo

Algoritmo Elegir_facultad

```
Definir matricula,promedio Como Real
Definir semestre Como Entero
Definir carrera Como Cadena
Definir facultad Como Entero
Leer matricula
Leer carrera
Leer semestre
Leer promedio
Escribir "1. Economía, 2. Computación, 3. Administración, 4.
Contabilidad"
Leer facultad
```

Segun facultad Hacer

```
1:
    si semestre = 6 y promedio >= 8.8 Entonces
        Escribir matricula
        Escribir carrera
        Escribir "Aceptado"
    FinSi
2:
    si semestre = 6 y promedio > 8.5 Entonces
        Escribir matricula
        Escribir carrera
        Escribir "Aceptado"
    FinSi
3:
    si semestre = 5 y promedio > 8.5 Entonces
        Escribir matricula
        Escribir carrera
        Escribir "Aceptado"
    FinSi
4:
    si semestre = 5 y promedio > 8.5 Entonces
        Escribir matricula
        Escribir carrera
        Escribir "Aceptado"
    FinSi
De Otro Modo:
    Escribir "Facultad invalida"
```

Fin Segun

FinAlgoritmo

Prueba de escritorio

Entrada					Procesos				Salidas		
matricula	carrera	semestre	Promedio	facultad	si semestre = 6 y promedio >= 8.8	si semestre = 6 y promedio > 8.5	si semestre = 5 y promedio > 8.5	si semestre = 6 y promedio >= 8.5	matricula	carrera	"aceptado"
1000	derecho	sexto	8.5	2					1000	derecho	Aceptado

Programa

```
#include<stdio.h>

int main() {
    char carrera[256]; //forma para declarar datos tipo cadena en ANSI C
    int facultad;
    float matricula;
    float promedio;
    int semestre;
    printf("Introducir matricula\t");
    scanf("%f",&matricula);
    printf("Introducir carrera\t");
    scanf("%s",carrera);
    printf("Introducir semestre\t");
    scanf("%d",&semestre);
    printf("Introducir promedio\t");
    scanf("%f",&promedio);
    printf("Elige una facultad\n1. Economía\n2. Computación\n3.
Administración\n4. Contabilidad\n");
    scanf("%d",&facultad);
    switch (facultad) {
        case 1:
            if (semestre==6 && promedio>=8.8) {
                printf("Matricula: %f\n",matricula);
                printf("Carrera: %s\n",carrera);
                printf("Aceptado\n");
            }
            break;
        case 2:
            if (semestre==6 && promedio>8.5) {
                printf("Matricula: %f\n",matricula);
                printf("Carrera: %s\n",carrera);
                printf("Aceptado\n");
            }
            break;
        case 3:
            if (semestre==5 && promedio>8.5) {
                printf("Matricula: %f\n",matricula);
                printf("Carrera: %s\n",carrera);
                printf("Aceptado\n");
            }
    }
```

```
        break;
case 4:
    if (semestre==5 && promedio>8.5) {
        printf("Matricula: %f\n",matricula);
        printf("Carrera: %s\n",carrera);
        printf("Aceptado\n");
    }
    break;
default:
    printf("Facultad invalida\n");
}
return 0;
}
```


10. En una tienda efectúan un descuento a los clientes dependiendo del monto de la compra. El descuento se efectúa en base al siguiente criterio:

Si el monto es menor que \$500, no hay descuento.

Si el monto está comprendido entre \$500 y hasta \$1,000, 5% de descuento.

Si el monto está comprendido entre \$1,000 y hasta \$7,000, 11% de descuento.

Si el monto está comprendido entre \$7,000 y hasta \$15,000, 18% de descuento.

Si el monto es mayor a \$15,000, 25% de descuento.

Desarrolle un programa que dado el monto de un cliente determine lo que debe de pagar el cliente.

Desarrollar el ejercicio, siguiendo la metodología propuesta.

Análisis

Datos de Salida

Datos de Entrada.

Otros datos

Condiciones.

Algoritmo en lenguaje natural o cualitativo.

Diseño

Diagrama

Seudocódigo

Prueba de escritorio

Programa en lenguaje ANSI C.

11. Hacer un Diagrama de Flujo que genere e imprima los números del 100 al 0 en forma decreciente.

Respuesta

Análisis

Datos de Salida

Lista de números del 100 al 0

Datos de Entrada.

Sin datos de entrada

Otros datos

Sin Otros datos

Condiciones.

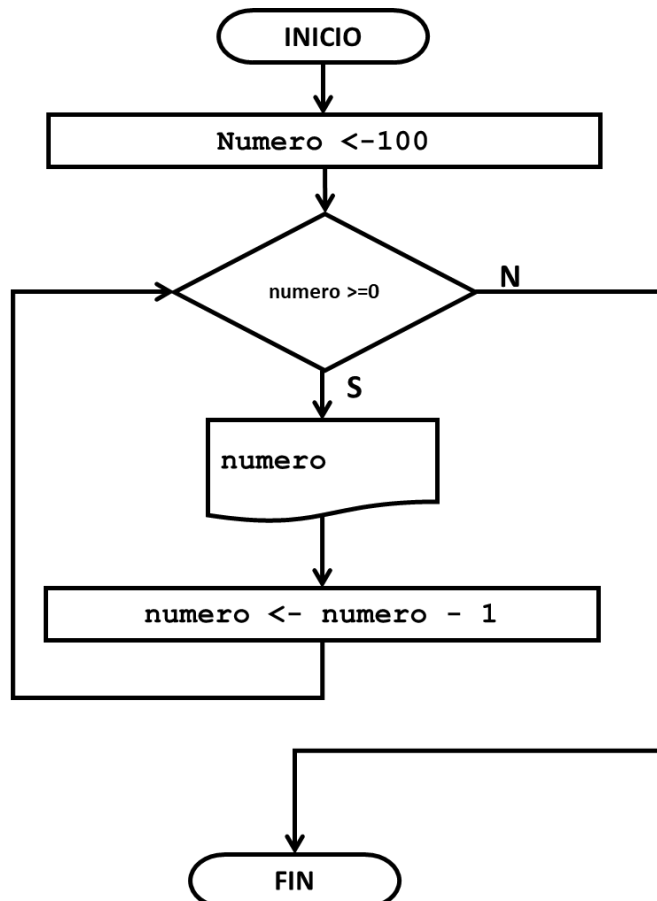
Terminar programa cuando se llegue a 0

Algoritmo en lenguaje natural o cualitativo.

1. Número igual a 100
2. Mostrar el valor del numero
3. A número restarle 1
4. Repetir desde el paso 2 hasta que numero sea igual a 0, cuando número sea 0 terminar.
5. Fin

Diseño

Diagrama



Seudocódigo

```
Algoritmo lista
    DEFINIR numero COMO Entero
    numero <- 100
    mientras numero >=0
        Escribir numero
        numero <- numero -1
    finmientras
FinAlgoritmo
```

Prueba de escritorio

En el caso de programas con ciclos se puede omitir la prueba de escritorio.

Programa

```
#include<stdio.h>

int main() {
    int numero;
    numero = 100;
    while (numero>=0) {
        printf("%i\n",numero);
        numero = numero-1;
    }
    return 0;
}
```

12. Hacer un Diagrama de Flujo que lea un número N y que genere e imprima los números impares desde el número 1 hasta el número leído, así como cuantos son.

Respuesta

Análisis

Datos de Salida

Números impares entre 1 y n

Cantidad de impares

Datos de Entrada.

Numero

Otros datos

Se sabe si un número es impar cuando su residuo entre 2 es 1.

Condiciones.

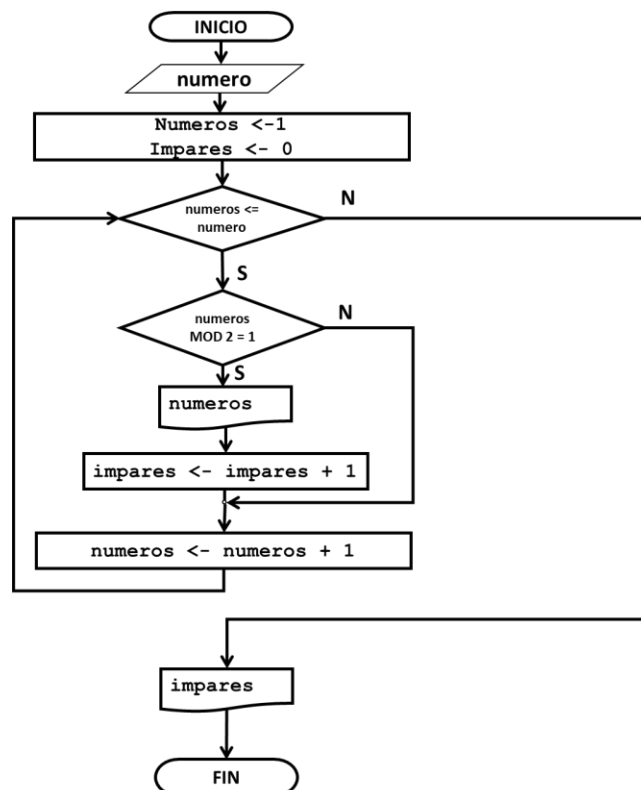
Usar números enteros

Algoritmo en lenguaje natural o cualitativo.

1. Introducir el numero
2. Impares es cero
3. Números es 1
4. Si números modulo 2 es 1 mostrar el valor de números Impares crece en 1
5. Números crece en 1
6. Repetir desde el paso 4 hasta que números sea mayor a numero
7. Mostrar impares
8. Fin

Diseño

Diagrama



Seudocódigo

```

Algoritmo Numeros_impares
  definir numero, numeros, impares Como entero
  leer numero
  numeros<-1
  impares<-0
  Mientras numeros <= numero
    si numeros mod 2 = 1 Entonces
      Escribir numeros
      impares <- impares+1
    FinSi
    numeros <- numeros+1
  FinMientras
  Escribir impares
FinAlgoritmo
  
```

Prueba de escritorio

Entrada	Procesos						Salidas	
numero	numeros <-1	impares <-0	numeros <= numero	numeros mod 2 = 1	impares <- impares+1	numeros <- numeros+1	numeros	impares
5	numero s<-1	impare s<-0	5 <=1 S	1 MOD 2 = 1 s	Impares <-1+1 Impares <-2	numeros <-1+1 numeros <- 2	1	
			5 <=2 S	2 MOD 2 = 1 N		numeros <-2+1 numeros <- 3		
			5 <=3 S	3 MOD 2 = 1 S	Impares <-1+1 Impares <-2	numeros <-3+1 numeros <- 4	3	
			5 <=4 S	4 MOD 2 = 1 n		numeros <-4+1 numeros <- 5		
			5 <=5 S	5 MOD 2 = 1 s	Impares <-2+1 Impares <-3	numeros <-5+1 numeros <- 6	5	
			6<=5 N					3

Programa

```
#include<stdio.h>

int main() {
    int impares;
    int numero;
    int numeros;
    printf("Introduce un numero: ");
    scanf("%i",&numero);
    numeros = 1;
    impares = 0;
    while (numeros<=numero) {
        if (numeros%2==1) {
            printf("%i es impar\n",numeros);
            impares = impares+1;
        }
        numeros = numeros+1;
    }
    printf("Se encontraron %i impares\n",impares);
    return 0;
}
```

13. Hacer un Diagrama de Flujo para imprimir los primeros 30 términos de la secuencia 1, 1/2, 1, 1/4, 1, 1/8, 1, 1/16....

Respuesta

Análisis

Datos de Salida

Los 30 elementos de la serie 1, 1/2, 1, 1/4, 1, 1/8, 1, 1/16...

Datos de Entrada.

Sin datos de entrada

Otros datos

Sin Otros datos

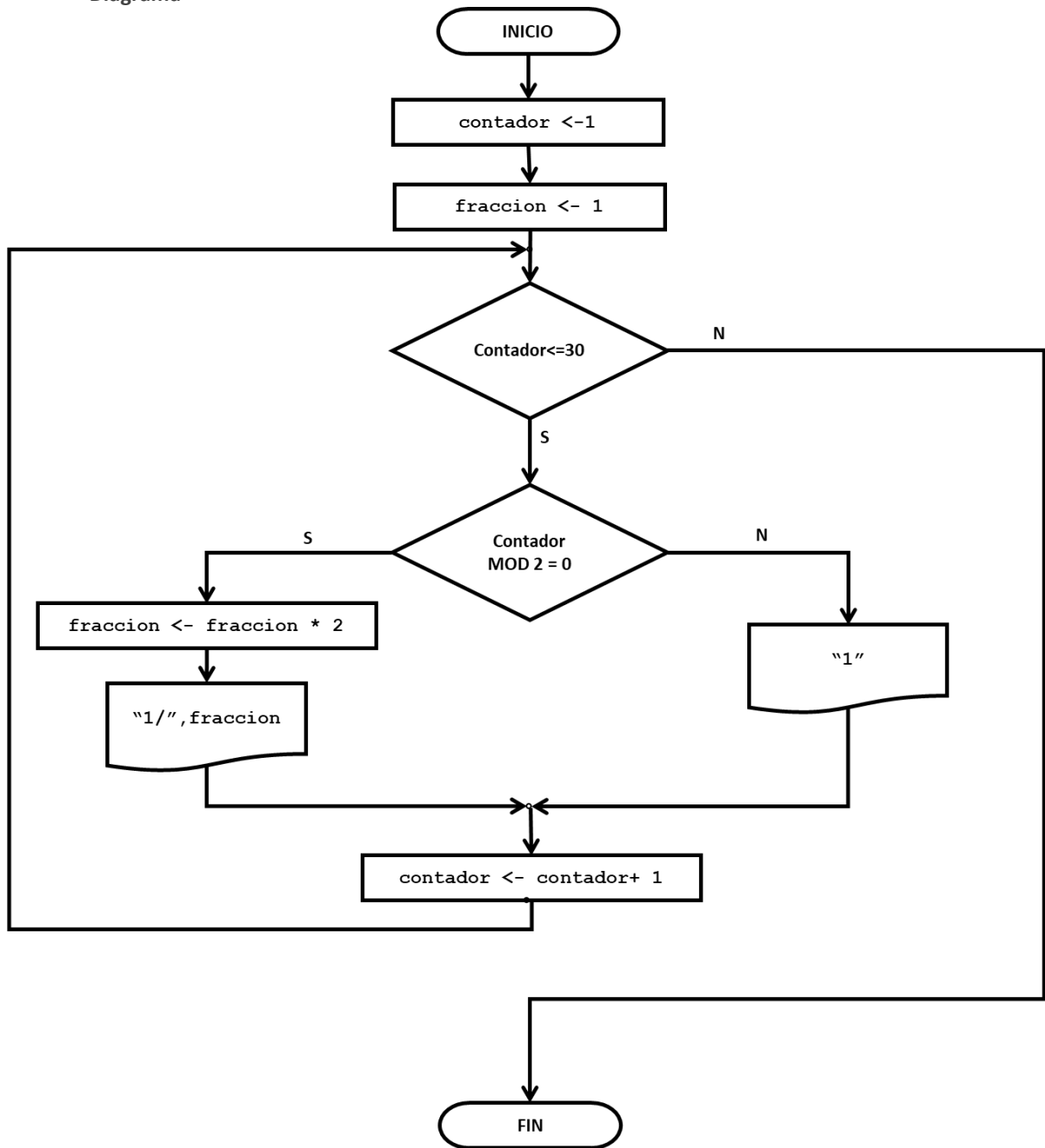
Condiciones.

Solo mostrar 30 elementos

Algoritmo en lenguaje natural o cualitativo.

1. Contador = 1
2. Fracción = 1
3. Si contador MOD 2 fracción = fracción*2 y mostrar 1/fracción
4. Sino mostrar 1
5. Al contador sumarle 1
6. Repetir desde el paso 3 mientras contador <=30
7. Fin

Diseño
Diagrama



Seudocódigo

Algoritmo serie

Definir contador,fraccion Como Entero

contador<-1

fraccion<-1

mientras contador <=30 Hacer

si contador MOD 2=0 Entonces

fraccion=fraccion*2

Escribir "1/",fraccion

sino

Escribir "1"

FinSi

contador<-contador+1

FinMientras

FinAlgoritmo

Prueba de escritorio

Entra da	Procesos						Salidas	
	contador<-1	fraccion<-1	contador <=30	contador MOD 2=0	fraccion=fraccio n*2	contador<- contador+1	1/fraccion	1
	contador<-1	fraccion<-1	1<=30 s	1 MOD 2 = 0 N		Contador <-1 +1 Contador<-2		1
			2<=30 s	2 MOD 2 = 0 S	Fracion <-1*2 Fraccion<-2	Contador <-2 +1 Contador<-3	1/2	
			3<=30 s	3 MOD 2 = 0 N		Contador <-3 +1 Contador<-4		1
			4<=30 s	4 MOD 2 = 0 S	Fracion <-2*2 Fraccion<-4	Contador <-4 +1 Contador<-5	1/4	
			5<=30 s	5 MOD 2 = 0 N		Contador <-5 +1 Contador<-6		1
			6<=30 s	6 MOD 2 = 0 S	Fracion <-4*2 Fraccion<-8	Contador <-6 +1 Contador<-7	1/8	
			7<=30 s	7 MOD 2 = 0 N		Contador <-7 +1 Contador<-8		1
			8<=30 s	8 MOD 2 = 0 S	Fracion <-8*2 Fraccion<-16	Contador <-8+1 Contador<-9	1/16	
			9<=30 s	9 MOD 2 = 0 N		Contador <-9+1 Contador<-10		1
			10<=30 s	10 MOD 2 = 0 S	Fracion <- 16*2 Fraccion<-32	Contador <-10 + 1 Contador<-11	1/32	
			11<=30 s	11 MOD 2 = 0 N		Contador <-11 +1 Contador<-12		1
			12<=30 s	12 MOD 2 = 0 S	Fracion <- 32*2 Fraccion<-64	Contador <-12 + 1 Contador<-13	1/64	
			13<=30 s	13 MOD 2 = 0 N		Contador <-13 +1 Contador<-14		1
			14<=30 s	14 MOD 2 = 0 S	Fracion <- 64*2 Fraccion<-128	Contador <-14 + 1 Contador<-15	1/128	

			15<=30 s	15 MOD 2 = 0 N		Contador <-15 + 1 Contador<- 16		1
			16<=30 s	16 MOD 2 = 0 S	Fracion <- 128*2 Fraccion<-256	Contador <-16 + 1 Contador<- 17	1/256	
			17<=30 s	17 MOD 2 = 0 N		Contador <-17 + 1 Contador<- 18		1
			18<=30 s	18 MOD 2 = 0 S	Fracion <- 256*2 Fraccion<-512	Contador <-18 + 1 Contador<-19	1/512	
			19<=30 s	19 MOD 2 = 0 S		Contador <-19+1 Contador<-20		1
			20<=30 s	20 MOD 2 = 0 S	Fracion <- 512*2 Fraccion<- 1024	Contador <-20 + 1 Contador<-21	1/1024	
			21<=30 s	21 MOD 2 = 0 N		Contador <-21 + 1 Contador<-22		1
			22<=30 s	22 MOD 2 = 0 S	Fracion <- 1024*2 Fraccion<- 2048	Contador <-22 + 1 Contador<-23	1/2048	
			23<=30 s	23 MOD 2 = 0 N		Contador <-23 + 1 Contador<-24		1
			24<=30 s	24 MOD 2 = 0 S	Fracion <- 2048*2 Fraccion<- 4096	Contador <-24 + 1 Contador<-25	1/4096	
			25<=30 S	25 MOD 2 = 0 N		Contador <-25 + 1 Contador<-26		1
			26<=30 s	26 MOD 2 = 0 S	Fracion <- 4096*2 Fraccion<- 8192	Contador <-26 + 1 Contador<-27	1/8192	
			27<=30 s	27 MOD 2 = 0 N		Contador <-27 + 1 Contador<-28		1
			28<=30 s	28 MOD 2 = 0 S	Fracion <- 8192*2 Fraccion<- 16384	Contador <-28 + 1 Contador<-29	1/1638 4	
			29<=30 s	29 MOD 2 = 0 N		Contador <-29 + 1 Contador<-30		1
			30<=30 s	30 MOD 2 = 0 S	Fracion <- 16384*2 Fraccion<- 32768	Contador <-30+1 Contador<-31	1/3276 8	
			31<=30 N					

Programa

```
#include<stdio.h>

int main() {
    int contador;
    int fraccion;
    contador = 1;
    fraccion = 1;
    printf("Muestra 30 elementos de la serie 1,1/2,1,1/4..\n");
    while (contador<=30) {
        if (contador%2==0) {
            fraccion = fraccion*2;
            printf("1/%d, ",fraccion);
        } else {
            printf("1, ");
        }
        contador = contador+1;
    }
    return 0;
}
```

14. Durante las siguientes 3 semanas usted trabajará en la cenaduría de José, este pagará 1 dólar el primer día, 2 dólares el segundo día, 4 dólares el tercer día, y así sucesivamente, cada día pues, usted recibirá un salario doble que el recibido el día anterior. Hacer un Diagrama de Flujo para calcular cuál será su salario al término fijado.

Desarrollar el ejercicio, siguiendo la metodología propuesta.

Análisis

Datos de Salida

Datos de Entrada.

Otros datos

Condiciones.

Algoritmo en lenguaje natural o cualitativo.

Diseño

Diagrama

Seudocódigo

Prueba de escritorio

Programa en lenguaje ANSI C.

15. Hacer un Diagrama de Flujo que lea N matrículas, para cada matrícula se leerán 2 calificaciones, que calcule el promedio de las 2 calificaciones y que imprima la matrícula, las dos calificaciones y el promedio de las dos calificaciones para cada una de las N matrículas, además deberá imprimir cuantos alumnos aprobaron y cuántos alumnos reprobaron. El diagrama terminará cuando se lea una matrícula igual a 9999 que no se procesará.

Desarrollar el ejercicio, siguiendo la metodología propuesta.

Análisis

Datos de Salida

Datos de Entrada.

Otros datos

Condiciones.

Algoritmo en lenguaje natural o cualitativo.

Diseño

Diagrama

Seudocódigo

Prueba de escritorio

Programa en lenguaje ANSI C.

- 16. Hacer un Diagrama de Flujo que calcule la suma de los números impares y la suma de los números pares que hay entre el 300 y el 450.**

Desarrollar el ejercicio, siguiendo la metodología propuesta.

Análisis

Datos de Salida

Datos de Entrada.

Otros datos

Condiciones.

Algoritmo en lenguaje natural o cualitativo.

Diseño

Diagrama

Seudocódigo

Prueba de escritorio

Programa en lenguaje ANSI C.

17. Hacer un diagrama de flujo que calcule el área o el perímetro, según lo decida el usuario, de las siguientes figuras:

- A. Cuadrado**
- B. Rectángulo**
- C. Triangulo**
- D. Circulo**

Después de realizar el cálculo el programa preguntara al usuario si desea realizar otra operación, si la respuesta es positiva repita el procedimiento, si es negativa finalice el programa enviando un mensaje de despedida.

Respuesta

Análisis

Datos de Salida

Área o perímetro de figura seleccionada

Datos de Entrada.

Datos de la figura seleccionada

Otros datos

Pi igual a 3.1416

Condiciones.

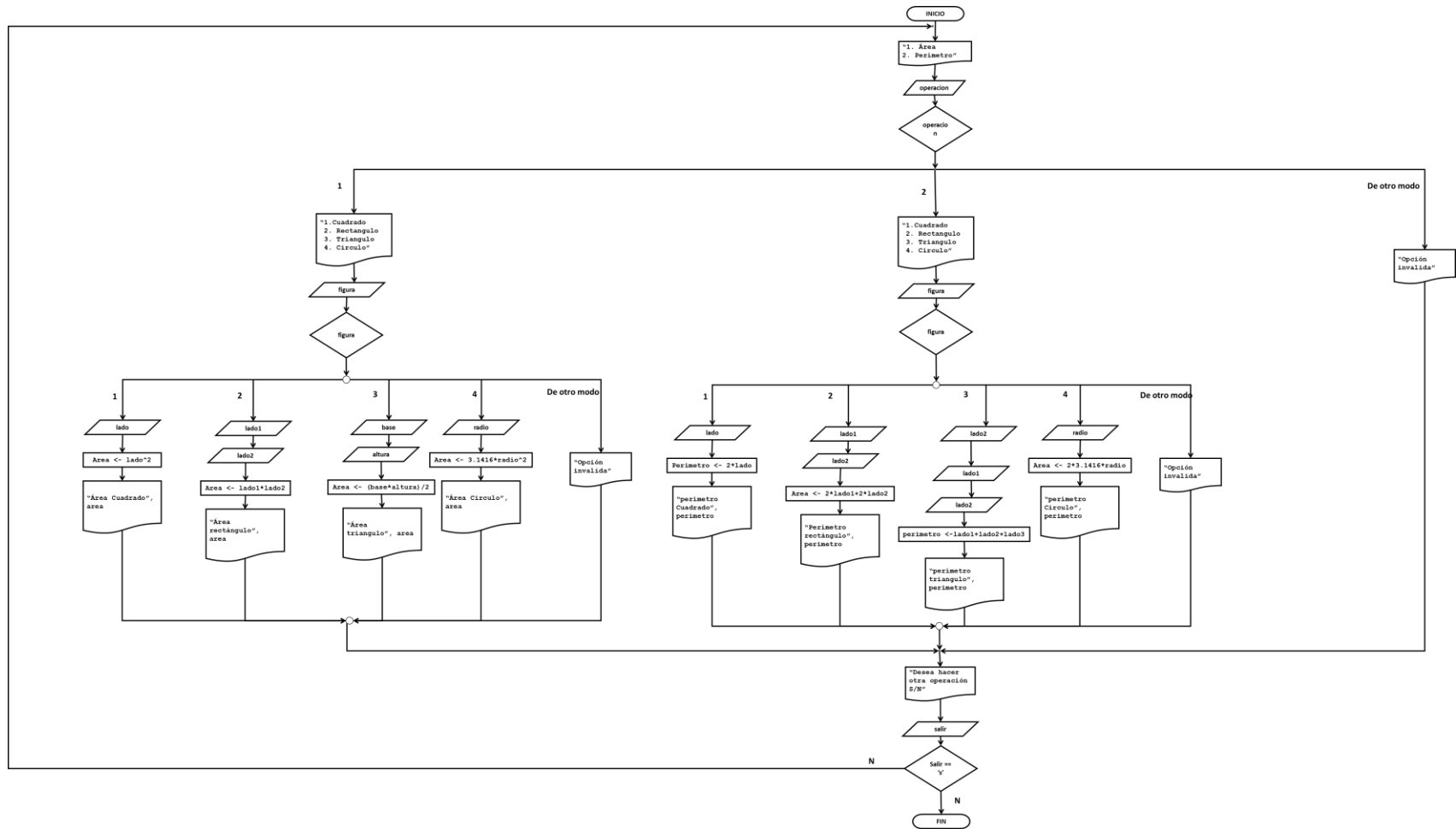
Dimensiones mayores a cero

Algoritmo en lenguaje natural o cualitativo.

1. Seleccionar área o perímetro
2. Si selecciona área
3. Seleccionar figura
4. Si selecciona cuadrado pedir longitud de un lado
5. El área se calcula con $\text{área} = \text{lado} * \text{lado}$
6. Si selecciona rectángulo pedir longitud de lado1 y de lado2
7. El área se calcula con $\text{área} = \text{lado1} * \text{lado2}$
8. Si selecciona triangulo pedir longitud de base y altura
9. El área se calcula con $\text{área} = (\text{base} * \text{altura})/2$
10. Si selecciona circulo pedir longitud del radio
11. El área se calcula con $\text{área} = \pi * \text{radio} * \text{radio}$
12. Si selecciona perímetro
13. Seleccionar figura
14. Si selecciona cuadrado pedir longitud de un lado
15. El perímetro se calcula con $\text{perímetro} = 4 * \text{lado}$
16. Si selecciona rectángulo pedir longitud de lado1 y de lado2
17. El área se calcula con $\text{área} = 2 * \text{lado1} + 2 * \text{lado2}$
18. Si selecciona triangulo pedir longitud de los lados
19. El perímetro se calcula con $\text{perímetro} = \text{lado1} + \text{lado2} + \text{lado3}$
20. Si selecciona circulo pedir longitud del radio
21. El perímetro se calcula con $\text{perímetro} = 2 * \pi * \text{radio}$
22. Para hacer otro calculo repetir desde el paso 1 si no desea repetir

23. Mostrar gracias por usar la aplicación
24. Fin

Diseño Diagrama



Seudocódigo

Algoritmo figuras

Definir area,perimetro,lado1,lado2,lado3,base,altura,radio Como Real

Definir operacion,figura Como Entero

Definir salir Como Caracter

Repetir

 Escribir "1. Área"

 Escribir "2. Perímetro"

 leer operacion

 Segun operacion Hacer

 1:

 Escribir "1. Cuadrado"

 Escribir "2. Rectángulo"

 Escribir "3. Triangulo"

 Escribir "4. Circulo"

 Leer figura

 Segun figura Hacer

 1:

 Leer lado1

$area \leftarrow lado1^2$

 Escribir "El area del cuadrado es ",area

 2:

 Leer lado1

 Leer lado2

$area \leftarrow lado1 * lado2$

 Escribir "El area del rectangulo es ",area

 3:

 Leer base

 Leer altura

$area \leftarrow (base * altura) / 2$

 Escribir "El area del triangulo es ",area

 4:

 Leer radio

$area \leftarrow 3.1416 * radio^2$

 Escribir "El area del circulo es ",area

 De Otro Modo:

 Escribir "Opción invalida"

 Fin Segun

2:

 Escribir "1. Cuadrado"

 Escribir "2. Rectángulo"

 Escribir "3. Triangulo"

 Escribir "4. Circulo"

 Leer figura

 Segun figura Hacer

 1:

 Leer lado1

$perimetro \leftarrow 4 * lado1$

 Escribir "El perimetro del cuadrado es ",perimetro

2:

Leer lado1
Leer lado2
 $\text{perimetro} \leftarrow 2 * \text{lado1} + 2 * \text{lado2}$
Escribir "El perimetro del rectangulo es ",perimetro

3:

Leer lado1
Leer lado2
Leer lado3
 $\text{perimetro} \leftarrow \text{lado1} + \text{lado2} + \text{lado3}$
Escribir "El perimetro del triangulo es ",perimetro

4:

Leer radio
 $\text{perimetro} \leftarrow 2 * 3.1416 * \text{radio}$
Escribir "El perimetro del circulo es ",perimetro

De Otro Modo:

Escribir "Opción invalida"

Fin Segun

De Otro Modo:

Escribir "Opción invalida"

Fin Segun

Escribir "Desea realizar otra operacion S/N"

Leer salir

Hasta Que salir \neq 's'

Escribir "Gracias por usar la aplicación"

FinAlgoritmo

Prueba de escritorio

[illegible]

Programa

```
#include<stdio.h>
#include<math.h>
#include<ctype.h>

int main() {
    float altura;
    float area;
    float base;
    int figura;
    float lado1;
    float lado2;
    float lado3;
    int operacion;
    float perimetro;
    float radio;
    char salir;
    do {
        printf("1. Área\n");
        printf("2. Perímetro\n");
        scanf("%i",&operacion);
        switch (operacion) {
            case 1:
                printf("1. Cuadrado\n");
                printf("2. Rectángulo\n");
                printf("3. Triangulo\n");
                printf("4. Circulo\n");
                scanf("%i",&figura);
                switch (figura) {
                    case 1:
                        printf("Introducir longitud de lado: ");
                        scanf("%f",&lado1);
                        area = pow(lado1,2);
                        printf("El area del cuadrado es %f\n",area);
                        break;
                    case 2:
                        printf("Introducir longitud de lado: ");
                        scanf("%f",&lado1);
                        printf("Introducir longitud de lado: ");
                        scanf("%f",&lado2);
                        area = lado1*lado2;
                        printf("El area del rectángulo es %f\n",area);
                        break;
                    case 3:
                        printf("Introducir longitud de la base: ");
                        scanf("%f",&base);
                        printf("Introducir longitud de la altura: ");
                        scanf("%f",&altura);
                        area = (base*altura)/2;
                        printf("El área del triángulo es %f\n",area);
                        break;
```

```

        case 4:
            printf("Introducir longitud del radio: ");
            scanf("%f",&radio);
            arrea = 3.1416*pon(radio,2);
            printf("El área del circulo es %f\n",area);
            break;
        default:
            printf("Opción invalida\n");
    }
    break;
case 2:
    printf("1. Cuadrado\n");
    printf("2. Rectángulo\n");
    printf("3. Triangulo\n");
    printf("4. Circulo\n");
    scanf("%i",&figura);
    switch (figura) {
        case 1:
            printf("Introducir longitud de lado: ");
            scanf("%f",&lado1);
            perimetro = 4*lado1;
            printf("El perímetro del cuadrado es %f\n",perimetro);
            break;
        case 2:
            printf("Introducir longitud de lado: ");
            scanf("%f",&lado1);
            printf("Introducir longitud de lado: ");
            scanf("%f",&lado2);
            perimetro = 2*lado1+2*lado2;
            printf("El perímetro del rectángulo es %f\n",perimetro);
            break;
        case 3:
            printf("Introducir longitud de lado: ");
            scanf("%f",&lado1);
            printf("Introducir longitud de lado: ");
            scanf("%f",&lado2);
            printf("Introducir longitud de lado: ");
            scanf("%f",&lado3);
            perimetro = lado1+lado2+lado3;
            printf("El perímetro del triángulo es %f\n",perimetro);
            break;
        case 4:
            printf("Introducir longitud del radio: ");
            scanf("%f",&radio);
            perimetro = 2*3.1416*radio;
            printf("El perímetro del circulo es %f\n",perimetro);
            break;
        default:
            printf("Opción invalida\n");
    }
    break;
default:
    printf("Opción invalida\n");
}

```

```
        fflush(stdin);
        printf("Desea realizar otra operación S/N\n");
        scanf("%c",&salir);
    } while (salir==tolower('s'));
    printf("Gracias por usar la aplicación\n");
    return 0;
```

8.- Presidente de Academia.

Docente	
Alberto Torres Santander	Presidente de Academia