



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

FACULTAD DE INGENIERÍA



INGENIERÍA EN COMPUTACIÓN

**UNIDAD DE APRENDIZAJE:
PROGRAMACIÓN ESTRUCTURADA**

**TEMA:
ESTRUCTURAS DE CONTROL**

**MATERIAL DIDÁCTICO
TRI-MEMORIA
(VERSIÓN PSEUDOCÓDIGO-DIAGRAMA DE FLUJO-
CODIFICACIÓN)**

**ELABORADO POR:
M. EN I. MIREYA SALGADO GALLEGOS**

MAYO 2016

ÍNDICE¹

| | |
|-------------------------------------|-----------|
| Presentación..... | 3 |
| Guía de uso..... | 5 |
| Forma de Aplicación..... | 5 |
| Ejemplos..... | 6 |
| Bibliografía..... | 11 |
| Ejercicios..... | 12 |

¹ Debido a la naturaleza del material didáctico y de la dinámica que se aplica con éste no se puede presentar un índice específico para el material, sin embargo se anexa el listado de programas a resolver en la parte final de este documento

PRESENTACIÓN

El programa de Programación Estructurada tiene por objetivo *aplicar el paradigma de la programación estructurada para representar en términos algorítmicos la solución de un problema real automatizable, así como elaborar programas completos utilizando el paradigma de la programación estructurada y mostrando en ellos el dominio pleno de variables simples, vectores, matrices, registros y modularidad*; con base en éste, el programa está conformado en la actualidad de 5 unidades de competencia, la principal de ellas es la que corresponde a *Aplicar la programación estructurada en la solución de problemas utilizando lenguaje informal y diagramas de flujo*, el cual comprende el aprendizaje de las estructuras de control secuenciales, selectivas e iterativas, contexto dentro del cual se elaboró este material didáctico orientado principalmente a los alumnos del primer periodo de la licenciatura de Ingeniería en Computación; con la finalidad de apoyar en el desarrollo de habilidades de interpretación, razonamiento, análisis y aplicación para la solución de programas informáticos en un lenguaje informal.

Este material está enfocado únicamente a la creación y realización de pseudocódigos, diagramas de flujo y codificación en lenguaje “C” de un conjunto de problemas seleccionados en donde se puedan aplicar los conocimientos previos adquiridos sobre estructuras de control secuenciales y decisivas (simples y dobles).

Cabe mencionar que se omite toda teoría debido a que es sólo una herramienta de apoyo para ésta, ya contemplada en el libro de texto de Programación Estructurada².

El material presentado, es una recopilación de programas representados en un conjunto de fichas de trabajo considerando su diagrama de flujo, pseudocódigo y codificación en lenguaje “C” que tienen la finalidad de “jugarse” como tri-memoria y ser utilizado por el profesor para la práctica de estos temas dentro del salón de clase.

² Albarrán Trujillo S.E, Salgado Gallegos M, **Programación Estructurada**, UAEM, 2010

Por motivos de la naturaleza del material no se anexa un apartado de la solución de los problemas debido a que el objetivo del material es que el alumno reconozca correctamente el pseudocódigo, diagrama de flujo y descripción y no podría cumplirse si se diera la solución.

Al final de este documento se anexa una bibliografía con la finalidad de que el alumno pueda, si así lo requiere, consultar algún material para su apoyo.

Los conocimientos previos requeridos para este material son el tema de estructuras de control secuenciales y decisivas (simple y doble).

GUÍA DE USO

La presente guía de uso pretende orientar la aplicación de este material, describiendo las partes de éste y ejemplificando con un programa.

Cada ejercicio está compuesto por: 3 fichas de trabajo las cuales se dividen en:

- Diagrama de Flujo
- Pseudocódigo
- Codificación en lenguaje “C”

Estas fichas se deben encontrar agrupadas de forma “revuelta” y colocadas al reverso.

El objetivo a lograr es que los alumnos reconozcan el diagrama de flujo, el pseudocódigo y la codificación del mismo programa de forma correcta.

FORMAS DE APLICACIÓN

Puede aplicarse de 2 formas, las cuales se describen a continuación:

FORMA N° 1:

Los alumnos forman un círculo alrededor del salón de clases (o cualquier otro lugar). Las fichas se agrupan en 1) pseudocódigos, 2) diagramas de flujo y 3) codificación, una vez agrupadas las fichas, éstas se colocan al reverso (de manera que no se vea lo escrito) en la parte central del círculo, por turnos cada alumno voltea 1 ficha de cada grupo, debe analizarlas y describir qué hace cada una de ellas para que los demás compañeros sepan qué fichas se han abierto. Si el alumno adivina las 3 fichas que resuelven el programa (revisadas por el profesor) el alumno se quedará con las 3 fichas y podrá repetir el turno, esto hasta que ya no adivine el conjunto de fichas del programa, y pasará el turno al siguiente compañero, y así sucesivamente hasta que las fichas se hayan terminado.

Gana el alumno que haya encontrados más tercias de programas.

FORMA N° 2:

Los alumnos forman un círculo alrededor del salón de clases (o cualquier otro lugar), en la parte central se colocan todas las fichas al reverso (de manera que no se vea lo escrito) y revueltas (sin grupos de pseudocódigo, diagrama de flujo y codificación), por turnos cada alumno voltea 3 fichas y debe analizarlas y describir qué hace cada una para que los demás compañeros sepan qué fichas ha abierto, el alumno que reconozca las tres fichas correspondientes a un programa tendrá la oportunidad de continuar su turno hasta que ya no adivine la tercia de fichas, y continuará el turno del siguiente compañero y así sucesivamente hasta que las fichas se hayan terminado.

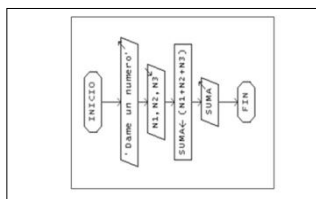
Gana el alumno que haya encontrados más tercias de programas.

EJEMPLO FORMA 1

Para mayor entendimiento se presentan a continuación unos ejercicios resueltos.

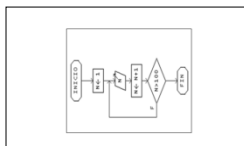
- Este ejemplo está formado por 6 fichas de trabajo (1 de diagrama de flujo, 1 de pseudocódigo y 1 de codificación) que forman un juego de tres fichas por programa, las cuales se presentan a continuación, considerando que no están en orden.

```
main(){
  n=1;
  do{
    printf("%d", n);
    n++;
  }while(n<=100);
}
```



```
Proceso
n<-1;
Repetir
  Escribir n;
  n<-n+1;
Hasta Que n>100
FinProceso
```

```
main(){
  printf("Dame tres números");
  scanf("%d%d%d",&n1,&n2,&n3);
  suma=n1+n2+n3;
  printf("La suma es: %d",suma);
}
```



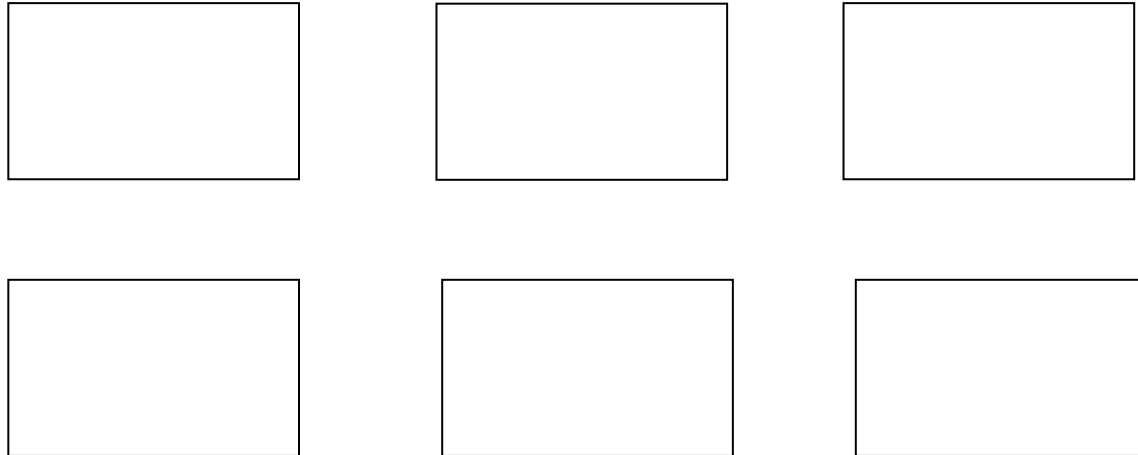
```
Proceso
  Escribir "Dame un numero";
  Leer N1, N2, N3;
  suma<-(N1+N2+N3);
  Escribir suma;
FinProceso
```

GRUPO 1
(Codificación)

GRUPO 2
(Diagramas de flujo)

GRUPO 3
(Pseudocódigo)

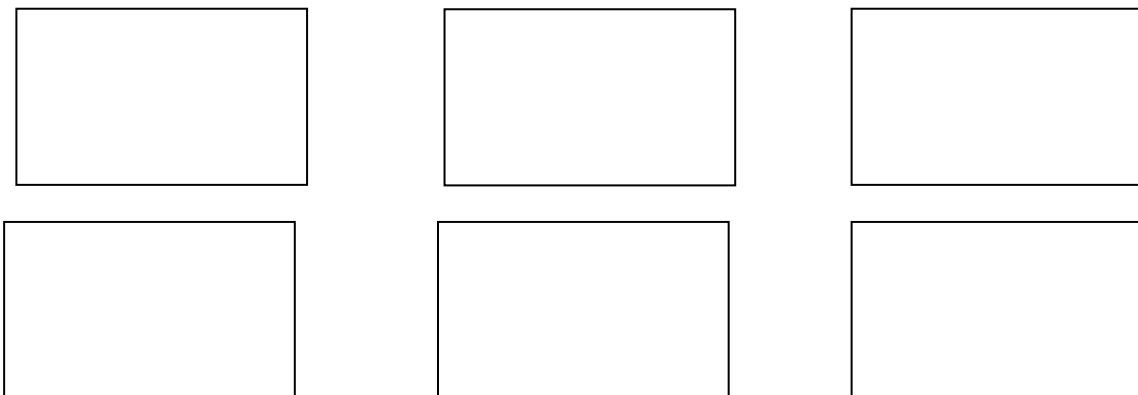
Estas fichas se revuelven por grupo y voltean para no ser vistas por los alumnos



El alumno 1 voltea 3 fichas

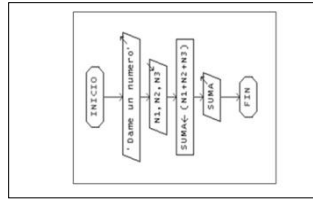
| | | |
|--|--|--|
| | | <p>Proceso Escribir "Dame un numero"; Leer N1, N2, N3; suma<-(N1+N2+N3); Escribir suma; FinProceso</p> |
| <pre>main(){ printf("Dame tres números"); scanf("%d%d%d",&n1,&n2,&n3); suma=n1+n2+n3; printf("La suma es: %d",suma); }</pre> | <pre> graph TD Inicio([Inicio]) --> Leer[Leer n1, n2, n3] Leer --> Proceso[suma = n1 + n2 + n3] Proceso --> Salida([Escribir suma]) Salida --> Fin([Fin]) </pre> | |

El alumno 1 analiza las 3 fichas, como no corresponden a la tercia correcta de un programa, vuelve a voltearlas y toca turno al alumno 2.



El alumno 2 voltea 3 fichas

1



```

Proceso
  Escribir "Dame un numero";
  Leer N1, N2, N3;
  suma<-(N1+N2+N3);
  Escribir suma;
FinProceso
    
```

```

main(){
  printf("Dame tres números");
  scanf("%d%d%d",&n1,&n2,&n3);
  suma=n1+n2+n3;
  printf("La suma es: %d",suma);
}
    
```



El alumno 2 adivina la tercia correcta del programa de la suma de 3 números y se queda con ellas.

El juego se repite hasta que ya no haya más fichas que adivinar.

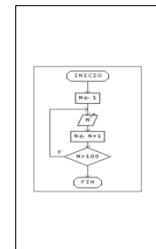
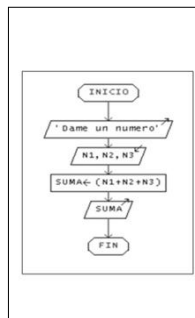
EJEMPLO FORMA 2

Para mayor entendimiento se presentan a continuación unos ejercicios resueltos.

- Este ejemplo está formado por 6 fichas de trabajo (1 de descripción, 1 de diagrama de flujo y 1 de pseudocódigo) que forman un juego de tres fichas por programa, las cuales se presentan a continuación, considerando que no están en orden.

```

main(){
  n=1;
  do{
    printf("%d", n);
    n++;
  }while(n<=100);
}
    
```



```

main(){
  printf("Dame tres números");
  scanf("%d%d%d",&n1,&n2,&n3);
  suma=n1+n2+n3;
  printf("La suma es: %d",suma);
}
    
```

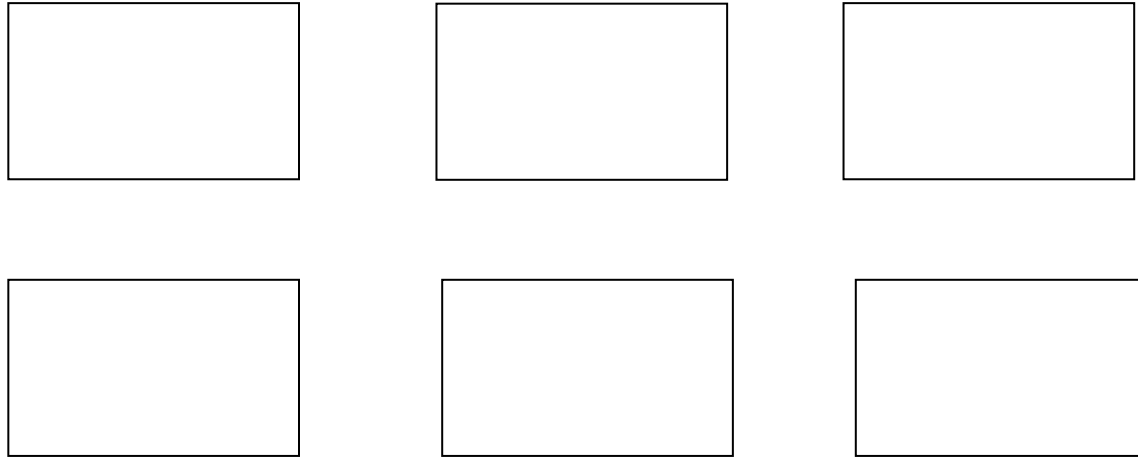
```

Proceso
  Escribir "Dame un numero";
  Leer N1, N2, N3;
  suma<-(N1+N2+N3);
  Escribir suma;
FinProceso
    
```

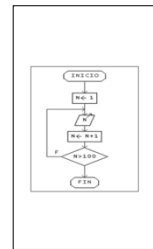
```

Proceso
  n<-1;
  Repetir
    Escribir n;
    n<-n+1;
  Hasta Que n>100
FinProceso
    
```


Estas fichas se revuelven y voltean para no ser vistas por los alumnos



El alumno 1 voltea 3 fichas

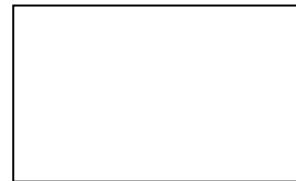


```

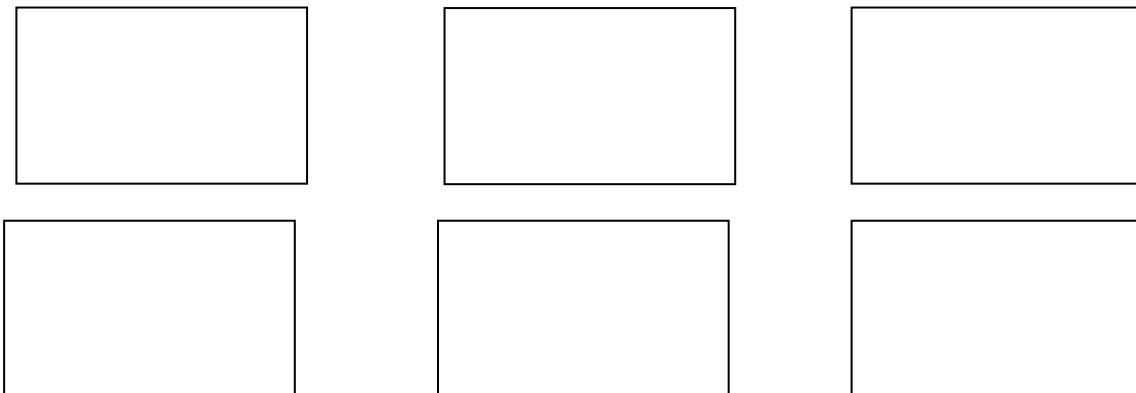
main(){
printf("Dame tres números");
scanf("%d%d%d",&n1,&n2,&n3);
suma=n1+n2+n3;
printf("La suma es: %d",suma);
}
  
```

```

Proceso
Escribir "Dame un numero";
Leer N1, N2, N3;
suma<-(N1+N2+N3);
Escribir suma;
FinProceso
  
```

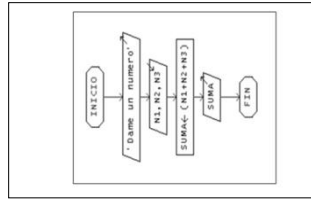


El alumno 1 analiza las 3 fichas, como no corresponden a la terna correcta de un programa, vuelve a voltearlas y toca turno al alumno 2.



El alumno 2 voltea 3 fichas

1



```

main(){
  printf("Dame tres números");
  scanf("%d%d%d",&n1,&n2,&n3);
  suma=n1+n2+n3;
  printf("La suma es: %d",suma);
}
  
```

```

Proceso
  Escribir "Dame un numero";
  Leer N1, N2, N3;
  suma<-(N1+N2+N3);
  Escribir suma;
FinProceso
  
```



El alumno 2 adivina la tercia correcta del programa de la suma de 3 números y se queda con ellas.

El juego se repite hasta que ya no haya más fichas que adivinar.

La finalidad de esta material es que el alumno analice que hace un DF y un pseudocódigo para poder identificar a manera de memoria la tercia correcta que corresponde a un programa.

De esta manera el alumno desarrolla la habilidad de análisis, entendimiento y desarrollo de un programa de una forma dinámica y divertida.

BIBLIOGRAFÍA

- 1.- Balcázar, J. L. (1993). **Programación metódica**. España: McGraw-Hill.
- 2.- Cairó, O. (1995). **Metodología de Programación**, Tomo I. Colombia: Computec.
- 3.- Cairó, O. (1995). **Metodología de Programación**, Tomo II. Colombia: Computec.
- 4.- Cairó, O. (2006) **Fundamentos de Programación. Piensa en C**, Pearson Educación, 2006.
- 5.- Criado, A. (2006). **Programacion en lenguajes estructurados**. México: Alfaomega-Rama.
- 6.- Harel, D. (1987). **Algorithmics, the Spirit of Computing**. Massachusetts: Addison Wesley.
- 7.- Joyanes A. L. (2008). **Fundamentos de programación. Algoritmos, Estructura de datos y objetos** (4ª ed.). México: McGraw-Hill.
- 8.- Joyanes A. L., Castillo, S. A., Sánchez, G. L, & Zahonero, M. I. (2002). **Programación en C**. Libro de problemas. México: Mc Graw Hill.
- 9.- Norton, P. (2006). **Introducción a la Computación** México: Mc Graw Hill.

EJERCICIOS

Son **77** los ejercicios desarrollados para este material los cuales se enumeran a continuación.

1. Calcular la suma de 3 números
2. Multiplicar 2 números y desplegar el resultado
3. Pedir un número y calcular su cuadrado
4. Pedir el nombre del usuario y desplegar un mensaje de “Hola“ y el nombre que se leyó.
5. Calcular el promedio de 5 calificaciones dadas por el usuario
6. Calcular el área de un terreno rectangular dadas sus dimensiones.
7. Leer la edad del usuario y desplegar los días, las horas, los minutos y los segundos que tiene la edad del usuario.
8. Leer por separado el nombre, el apellido paterno y el apellido materno de un usuario y desplegar su nombre completo empezando por apellidos.
9. Calcular la edad en años del usuario, preguntando su año de nacimiento.
10. Calcular los metros, centímetros y milímetros, de una distancia dada en kilómetros.
11. Calcular la cuenta total de cafés (\$5.00), tortas (\$15.00), dulces (\$1.50) y refrescos (\$6.00) pidiendo la cantidad consumida por cada uno de éstos.
12. Calcular el sueldo de un empleado ahora que se le ha dado el 27% de aumento.
13. Calcular la distancia entre dos puntos dados.
14. Calcular la multiplicación de dos fracciones dadas por separado el numerador del denominador y desplegarlas en forma de fracción.
15. Diseñar un programa para determinar la hipotenusa de un triángulo rectángulo, conocidas las longitudes de los catetos.
16. Calcular la división de dos fracciones dadas por separado el numerador del denominador y desplegarlas en forma de fracción.
17. Pedir el número de días y desplegar la cantidad de segundos contenidos en esos días.
18. Calcular la suma de dos fracciones dadas por separado el numerador del denominador y desplegarlas en forma de fracción.
19. Leer una cantidad expresada en dólares y transformarlos a pesos.
20. Leer dos números e intercambiar los contenidos de las variables de esos números.
21. Leer dos números, realizar la suma y decir si es positiva o negativa.
22. Calcular el perímetro y área de un círculo.
23. Calcular el perímetro de un triángulo, utilizando la fórmula de distancia entre dos puntos para cada lado del triángulo.
24. Pedir dos números, calcular la suma, resta y multiplicación de estos.
25. Calcular el área de un triángulo
26. Encontrar las raíces de x_1 y x_2 $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
27. Leer el peso en pies y transformarlo a libras
28. Leer un número y la potencia al cual se quiere elevar, calcular el resultado.
29. Determinar el área y volumen de un cilindro
30. Calcular el costo de un terreno rectangular, considerando que el precio por metro cuadrado es de \$250.00
31. Calcular el cambio de una cuenta, dando el costo de la cuenta y la cantidad a pagar
32. Pedir la marca y modelo de un vehículo, y desplegar modelo y marca
33. Transformar Km a millas, los Km son dados por el usuario

34. Calcular el cuadrado del número 243
35. Leer centímetros y transformarlos a pulgadas
36. Resolver la ecuación $Ax+B = C$
37. Leer un número y decir si es positivo
38. Leer un número y decir si es positivo o negativo
39. Leer un número y decir si es cero
40. Calcular el promedio de 3 calificaciones de un alumno y decir si está aprobado ($\text{prom} \geq 60$) o reprobado

ESTOS SON LOS EJEMPLOS QUE LOS ALUMNOS UTILIZARÁN PARA EL MEMORAMA TRIPLE DE PROGRAMAS, LOS CUALES YA SE ENCUENTRAN EN FICHAS DE TRABAJO PARA SU USO.

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|--|--|---|
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() { int N1, N2, N3, Suma; printf("Dame 3 números:"); scanf ("%d%d%d",&N1,&N2,&N3); Suma= N1+N2+N3; printf("La suma es:%d",Suma); return 0; }</pre> | <pre>Algoritmo sin_titulo escribir("Dame 3 números:"); leer N1,N2,N3; Suma= N1+N2+N3; escribir("La suma es:",Suma); FinAlgoritmo</pre> | <pre>graph TD Start([Algoritmo sin_titulo]) --> Input[/Dame 3 numeros:/] Input --> Read[/N1, N2, N3/] Read --> Process[Suma ← N1+N2+N3] Process --> Output[/La suma es:/] Output --> End([FinAlgoritmo])</pre> |
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() { float N1,N2, Resultado; printf("Dame 2 números: "); scanf("%f%f", &N1,&N2); Resultado= N1*N2; printf("La multiplicación es:%f",Resultado); }</pre> | <pre>Algoritmo sin_titulo escribir("Dame 2 números: "); leer N1,N2 Resultado= N1*N2; escribir("La multiplicación es:",Resultado); FinAlgoritmo</pre> | <pre>graph TD Start([Algoritmo sin_titulo]) --> Input[/Dame 2 numeros:/] Input --> Read[/N1, N2/] Read --> Process[Resultado ← N1*N2] Process --> Output[/La multiplicacion es.../] Output --> End([FinAlgoritmo])</pre> |
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() { float N1, Resultado; printf("Dame un número: "); scanf("%f", &N1); Resultado= N1*N1; printf("El cuadrado es: %f", Resultado); return 0; }</pre> | <pre>Algoritmo sin_titulo escribir("Dame un número: ") leer N1; Resultado= N1*N1; escribir("El cuadrado es:", Resultado); FinAlgoritmo</pre> | <pre>graph TD Start([Algoritmo sin_titulo]) --> Input[/Dame un numero:/] Input --> Read[/N1/] Read --> Process[Resultado ← N1*N1] Process --> Output[/El cuadrado es:/] Output --> End([FinAlgoritmo])</pre> |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|--|---|---|
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() { char nom[30]; printf("Escribe tu nombre:"); scanf("%s",nom); printf("Hola %s",nom); return 0; }</pre> | <pre>Algoritmo sin_titulo Escribir ('Escribe tu nombre:') Leer nom Escribir ("Hola ", nom) FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> Input[/('Escribe tu nombre:')/] Input --> Process[/nom/] Process --> Output[/('nom')/] Output --> End([FinAlgoritmo]) </pre> |
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() { float C1,C2,C3,C4,C5,Promedio_Total; printf("Para obtener tu promedio total introduce las calificaciones obtenidas de tus 5 materias del semestre, separadas por una coma: "); scanf("%f,%f,%f,%f,%f", &C1,&C2,&C3,&C4,&C5); Promedio_Total=(C1+C2+C3+C4+C5)/(5); printf("El promedio total de tus calificaciones del semestre es: %f", Promedio_Total); return 0; }</pre> | <pre>Algoritmo sin_titulo Escribir ("Para obtener tu promedio total introduce las calificaciones obtenidas de tus 5 materias del semestre, separadas por una coma: "); Leer C1,C2,C3,C4,C5 Promedio_Total=(C1+C2+C3+C4+C5) /(5); Escribir ("El promedio total de tus calificaciones del semestre es:", Promedio_Total); FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> Input[/('Para obtener tu prom...')/] Input --> Process[/C1,C2,C3,C4,C5/] Process --> Process2[/Promedio_Total ← (C1+C...)] Process2 --> Output[/('El promedio total de...')/] Output --> End([FinAlgoritmo]) </pre> |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|---|--|---|
| <pre> #include <iostream> #include <stdio.h> using namespace std; int main() { float ancho,largo,area; printf("Dame el ancho y largo:"); scanf("%f%f",&ancho,&largo); area=ancho*largo; printf("El Area es:%f",area); return 0; } </pre> | <pre> Algoritmo sin_titulo escribir("Dame el ancho y largo:"); leer ancho,largo area=ancho*largo; escribir("El Area es:",area) FinAlgoritmo </pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> Input[/('Dame el ancho y larg...')/] Input --> Process[ancho, largo] Process --> Calc[area ← ancho*largo] Calc --> Output[/('El Area es:', area)/] Output --> End([FinAlgoritmo]) </pre> |
| <pre> #include <iostream> #include <stdio.h> using namespace std; int main() {int E, ED, EH, EM, ES; printf("Cuál es tu edad?"); scanf("%d",&E); ED=E*365; EH=ED*24; EM=EH*60; ES=EM*60; printf("Tu edad en años es:%d", E); printf("Tu edad en días es:%d", ED); printf("Tu edad en horas es:%d", EH); printf("Tu edad en minutos es:%d", EM); printf("Tu edad en segundos es:%d", ES); return 0; } </pre> | <pre> Algoritmo sin_titulo escribir("Cuál es tu edad?"); leer E; ED=E*365; EH=ED*24; EM=EH*60; ES=EM*60; escribir("Tu edad en años es:", E); escribir("Tu edad en días es:", ED); escribir("Tu edad en horas es:", EH); escribir("Tu edad en minutos es:", EM); escribir("Tu edad en segundos es:", ES); FinAlgoritmo </pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> Input[/('Cuál es tu edad?')/] Input --> Process[E] Process --> Calc1[ED ← E*365] Calc1 --> Calc2[EH ← ED*24] Calc2 --> Calc3[EM ← EH*60] Calc3 --> Calc4[ES ← EM*60] Calc4 --> Output1[/('Tu edad en años es:', E)/] Output1 --> Output2[/('Tu edad en días es:'...)/] Output2 --> Output3[/('Tu edad en horas es:'...)/] Output3 --> Output4[/('Tu edad en minutos e...)/] Output4 --> Output5[/('Tu edad en segundos ...)/] Output5 --> End([FinAlgoritmo]) </pre> |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|--|---|-------------------|
| <pre> #include <iostream> #include <stdio.h> using namespace std; int main() { char Nom[20],Ap[20],Am[20]; printf("Nom:"); scanf("%s",&Nom); printf("Ap:"); scanf("%s",&Ap); printf("Am:"); scanf("%s",&Am); printf("Tu nombre completo es: %s",Ap); printf(" %s",Am); printf(" %s",Nom); return 0; } </pre> | <pre> Algoritmo sin_titulo escribir("Nom:"); leer Nom; escribir("Ap:"); leer Ap; escribir("Am:"); leer Am escribir("Tu nombre completo es:",Ap); escribir(,Am); escribir(,Nom); FinAlgoritmo </pre> | |
| <pre> #include <iostream> #include <stdio.h> using namespace std; int main() {int N1, E; printf("Dame tu año de nacimiento:"); scanf("%d", &N1); E=2016-N1; printf("Tu edad es:%d", E); return 0; } </pre> | <pre> Algoritmo sin_titulo Escribir ("Dame tu año de nacimiento:"); leer N1; E=2016-N1; escribir("Tu edad es:", E); FinAlgoritmo </pre> | |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|---|---|-------------------|
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() { float Distancia,M,Cm,Mm; printf("Dame la distancia en Km:"); scanf("%f",&Distancia); M=Distancia/1000; Cm=M/1000; Mm=Cm/1000; printf("La distancia en metros es:%f",M); printf("La distancia en centímetros es:%f",Cm); printf("La distancia en Milímetros es:%f",Mm); return 0; }</pre> | <p>Algoritmo sin_titulo escribir("Dame la distancia en Km:"); leer Distancia; $M = \text{Distancia} / 1000$; $Cm = M / 1000$; $Mm = Cm / 1000$; escribir("La distancia en metros es:",M); escribir("La distancia en centímetros es:",Cm); escribir("La distancia en Milímetros es:",Mm) FinAlgoritmo</p> | |
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() {int N1, N2, N3, N4; float T, TD; printf("cuantos cafés fueron vendidos?"); scanf("%d",&N1); printf("cuantas tortas fueron vendidas?"); scanf("%d",&N2); printf("cuantos dulces fueron vendidos?"); scanf("%d",&N3); printf("cuantos refrescos fueron vendidos?"); scanf("%d",&N4); T=(N1*5)+(N2*15)+(N3*1.5)+(N4*6); TD= T-(T*.08); printf("%d", TD); return 0;}</pre> | <p>Algoritmo sin_titulo escribir("cuantos cafés fueron vendidos?"); leer N1; escribir("cuantas tortas fueron vendidas?"); leer N2; escribir("cuantos dulces fueron vendidos?"); leer N3; escribir("cuantos refrescos fueron vendidos?"); leer N4; $T = (N1 * 5) + (N2 * 15) + (N3 * 1.5) + (N4 * 6)$; $TD = T - (T * .08)$; escribir(TD) Finalgoritmo</p> | |
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() { float Sueldo,Aumento; printf("Dame tu sueldo actual:"); scanf("%f",&Sueldo); Aumento=Sueldo*1.027; printf("Tu sueldo es:%f",Aumento); return 0; }</pre> | <p>Algoritmo sin_titulo escribir("Dame tu sueldo actual:"); leer Sueldo; $Aumento = \text{Sueldo} * 1.027$; escribir("Tu sueldo es:",Aumento) FinAlgoritmo</p> | |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|--|--|--|
| <pre>#include <iostream> #include <stdio.h> #include <math.h> using namespace std; int main() {float x1, y1, x2, y2, D; printf("Dame x1 y y1 de P1:"); scanf("%f%f",&x1, &y1); printf("Dame x2 y y2 de P2"); scanf("%f%f", &x2, &y2); D=sqrt(pow(x1-x2, 2)+pow(y1-y2,2)); printf("la distancia es:%f", D); return 0; }</pre> | <pre>Algoritmo sin_titulo escribir("Dame x1 y y1 de P1."); leer x1,y1 escribir("Dame x2 y y2 de P2"); leer x2,y2 D=sqrt(pow(x1-x2, 2)+pow(y1-y2,2)); escribir("la distancia es:", D); FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> Input1[/Dame x1 y y1 de P1:/] Input1 --> Process1[x1, y1] Process1 --> Input2[/Dame x2 y y2 de P2:/] Input2 --> Process2[x2, y2] Process2 --> Process3["D = sqrt[pow[x1-x2,2]+..."] Process3 --> Output1[/la distancia es:', D)/] Output1 --> End([FinAlgoritmo]) </pre> |
| <pre>#include <iostream> #include<stdio.h> using namespace std; int main(){ int A,B,C,D,Num,Den; printf("Dame el numerador de la primer fracción:"); scanf("%d",&A); printf("Dame el denominador de la primer fracción:"); scanf("%d",&B); printf("Dame el numerador de la segunda fracción:"); scanf("%d",&C); printf("Dame el denominador de la segunda fracción:"); scanf("%d",&D); Num=A*C; Den=B*D; printf("El resultado es:%d",Num); printf("/"); printf("%d",Den); return 0;}</pre> | <pre>Algoritmo sin_titulo escribir("Dame el numerador de la primer fracción:"); leer A escribir("Dame el denominador de la primer fracción:"); leer B escribir("Dame el numerador de la segunda fracción:"); leer C escribir("Dame el denominador de la segunda fracción:"); leer D Num=A*C; Den=B*D; escribir("El resultado es:",Num); escribir("/"); escribir(Den); FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> Input1[A] Input1 --> Process1[A] Process1 --> Input2[B] Input2 --> Process2[B] Process2 --> Input3[C] Input3 --> Process3[C] Process3 --> Input4[D] Input4 --> Process4["Num = A*C"] Process4 --> Process5["Den = B*D"] Process5 --> Output1[/El resultado es:', Num)/] Output1 --> Output2[/ /] Output2 --> Output3[/ Den] Output3 --> End([FinAlgoritmo]) </pre> |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|---|---|--|
| <pre>#include <iostream> #include <stdio.h> #include <math.h> using namespace std; int main(){ float Co,Ca,hipotenusa; printf("Dame el cateto opuesto:"); scanf("%f",&Co); printf("Dame el cateto adyacente:"); scanf("%f",&Ca); hipotenusa=sqrt((Co*Co)+(Ca*Ca)); printf("La hipotenusa es:%f",hipotenusa); return 0;}</pre> | <pre>Algoritmo sin_titulo escribir("Dame el cateto opuesto:"); leer Co; escribir("Dame el cateto adyacente:"); leer Ca; hipotenusa=sqrt((Co*Co)+(Ca*Ca)); escribir("La hipotenusa es:",hipotenusa) FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> P1[/Dame el cateto opues.../] P1 --> I1[/Co/] I1 --> P2[/Dame el cateto adyac.../] P2 --> I2[/Ca/] I2 --> P3[hipotenusa ← sqrt[(Co*...)] P3 --> P4[/La hipotenusa es:',h.../] P4 --> End([FinAlgoritmo]) </pre> |
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() {int a, b, c, d,ad, bc, bd, nf, df; printf("dame el numerador de una fracción:"); scanf("%d", &a); printf("dame el denominador de la misma fracción:"); scanf("%d", &b); printf("escribe el numerador de otra fracción:"); scanf("%d", &c); printf("escribe el denominador de la misma fracción:"); scanf("%d", &d); ad= a*d; bc= b*c; nf= ad; df= bc; printf("el resultado en fracción es:%d", nf); printf("/"); printf("%d", df); }</pre> | <pre>Algoritmo sin_titulo escribir("dame el numerador de una fracción:"); leer a; escribir("dame el denominador de la misma fracción:"); leer b; escribir("escribe el numerador de otra fracción:"); leer c; escribir("escribe el denominador de la misma fracción:"); leer d; ad= a*d; bc= b*c; nf= ad; df= bc; escribir("el resultado en fracción es:", nf); escribir("/"); escribir (df); FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> P1[/dame el numerador de.../] P1 --> I1[/a/] I1 --> P2[/dame el denominador .../] P2 --> I2[/b/] I2 --> P3[/escribe el numerador.../] P3 --> I3[/c/] I3 --> P4[/escribe el denominad.../] P4 --> I4[/d/] I4 --> P5[ad ← a*d] P5 --> P6[bc ← b*c] P6 --> P7[nf ← ad] P7 --> P8[df ← bc] P8 --> P9[/el resultado en frac.../] P9 --> P10[/ /] P10 --> P11[/df/] P11 --> End([FinAlgoritmo]) </pre> |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|--|--|--|
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() {int D, S; printf("Dame el número de días:"); scanf("%d", &D); S=D*86400; printf("el número de segundos es:%d", S); return 0; }</pre> | <pre>Algoritmo sin_titulo escribir("Dame el número de días:"); leer D; S=D*86400; escribir("el número de segundos es:", S) FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> Input[/Dame el número de días.../] Input --> D[/D/] D --> Process[S ← D*86400] Process --> Output[/el número de segundos.../] Output --> End([FinAlgoritmo]) </pre> |
| <pre>#include <iostream> #include<stdio.h> using namespace std; int main() { int A,B,C,D,Num,Den; printf("Dame el numerador de la primer fracción:"); scanf("%d",&A); printf("Dame el denominador de la primer fracción:"); scanf("%d",&B); printf("Dame el numerador de la segunda fracción:"); scanf("%d",&C); printf("Dame el denominador de la segunda fracción:"); scanf("%d",&D); Num=(A*D)+(B*C); Den=(B*D); printf("El resultado es:%d",Num); printf("/"); printf("%d",Den); return 0; }</pre> | <pre>Algoritmo sin_titulo escribir("Dame el numerador de la primer fracción:"); leer A; escribir("Dame el denominador de la primer fracción:"); leer B; escribir("Dame el numerador de la segunda fracción:"); leer C; escribir("Dame el denominador de la segunda fracción:"); leer D; Num=(A*D)+(B*C); Den=(B*D); escribir("El resultado es:",Num); escribir("/"); escribir(Den); FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> Input1[/Dame el numerador de.../] Input1 --> A[/A/] A --> Input2[/Dame el denominador .../] Input2 --> B[/B/] B --> Input3[/Dame el numerador de.../] Input3 --> C[/C/] C --> Input4[/Dame el denominador .../] Input4 --> D[/D/] D --> Process1[Num ← (A*D)+(B*C)] Process1 --> Process2[Den ← (B*D)] Process2 --> Output[/El resultado es: /] Output --> DenOutput[/Den/] DenOutput --> End([FinAlgoritmo]) </pre> |
| <pre>#include <iostream> #include<stdio.h> using namespace std; int main() { float N,Pesos; printf("Cantidad de dólares:"); scanf("%f",&N); Pesos=N*15.85; printf("La cantidad en pesos es:%f",Pesos); return 0; }</pre> | <pre>Algoritmo sin_titulo escribir ("Cantidad de dólares:"); leer N; Pesos=N*15.85; escribir ("La cantidad en pesos es:",Pesos); FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> Input[/Cantidad de dólares:/] Input --> N[/N/] N --> Process[Pesos ← N*15.85] Process --> Output[/La cantidad en pesos.../] Output --> End([FinAlgoritmo]) </pre> |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|---|---|-------------------|
| <pre> #include <iostream> #include <stdio.h> using namespace std; int main() { float N1, N2, N3; printf("Dame el primer valor:"); scanf("%f", &N1); printf("Dame el segundo valor:"); scanf("%f", &N2); scanf("%f", &N3); N3=N1; N2=N3; N1=N2; printf("valor uno:%f", N1); printf("valor dos:%f", N2); return 0; } </pre> | <pre> Algoritmo sin_titulo Escribir ('Dame el primer valor:') Leer N1 Escribir ('Dame el segundo valor:') Leer N2 Leer N3 N3 <- N1 N2 <- N3 N1 <- N2 Escribir (N1) Escribir (N2) FinAlgoritmo </pre> | |
| <pre> #include <iostream> #include <stdio.h> using namespace std; int main() {float r, P, A; printf("Dame el radio:"); scanf("%f",&r); P=2*r*3.1416; A=r*r*3.1416; printf("El perímetro es:%f", P); printf("El área es:%f", A); return 0; } </pre> | <pre> Algoritmo sin_titulo escribir("Dame el radio:"); leer r; P=2*r*3.1416; A=r*r*3.1416; escribir("El perímetro es:", P); escribir ("El área es:", A) FinAlgoritmo </pre> | |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|--|---|---|
| <pre> #include <iostream> #include <stdio.h> #include <math.h> using namespace std; int main(){ float x1, y1, x2, y2, x3, y3, DA, DB, DC, P; printf("Dame x1 y y1:"); scanf("%f%f", & x1, &y1); printf("Dame x2 y y2:"); scanf("%f%f", & x2, &y2); printf("Dame x3 y y3:"); scanf("%f%f", & x3, &y3); DA=sqrt(pow(x1-x2,2)+pow(y1-y2, 2)); DB=sqrt(pow(x2-x3,2)+pow(y2-y3, 2)); DC=sqrt(pow(x3-x1,2)+pow(y3-y1, 2)); P=DA+DB+DC; printf("El perímetro es:%f", P); return 0; } </pre> | <pre> Algoritmo sin_titulo escribir("Dame x1 y y1:"); leer x1,y1; escribir("Dame x2 y y2:"); leer x2,y2; escribir("Dame x3 y y3:"); leer x3,y3; DA=sqrt(pow(x1-x2,2)+pow(y1-y2, 2)); DB=sqrt(pow(x2-x3,2)+pow(y2-y3, 2)); DC=sqrt(pow(x3-x1,2)+pow(y3-y1, 2)); P=DA+DB+DC; escribir("El perímetro es:"); P; FinAlgoritmo </pre> | <pre> graph TD Start([Algoritmo sin titulo]) --> I1[/('Dame x1 y y1:')/] I1 --> P1[x1, y1] P1 --> I2[/('Dame x2 y y2:')/] I2 --> P2[x2, y2] P2 --> I3[/('Dame x3 y y3:')/] I3 --> P3[DA ← sqrt[pow[x1-x2, 2]...]] P3 --> P4[DB ← sqrt[pow[x2-x3, 2]...]] P4 --> P5[DC ← sqrt[pow[x3-x1, 2]...]] P5 --> P6[P ← DA+DB+DC] P6 --> O1[/('El perímetro es:', P)/] O1 --> End([FinAlgoritmo]) </pre> |
| <pre> #include <iostream> #include<stdio.h> using namespace std; int main(){ float N1,N2,suma,resta,multiplicacion; printf("Dame 2 números:"); scanf("%f%f",&N1,&N2); suma=N1+N2; resta=N1-N2; multiplicacion=N1*N2; printf("la suma es:%f",suma); printf("la resta es:%f",resta); printf("la multiplicación es:%f", multiplicación); return 0; } </pre> | <pre> Algoritmo sin_titulo escribir("Dame 2 números:"); leer N1,N2; suma=N1+N2; resta=N1-N2; multiplicacion=N1*N2; escribir("la suma es:",suma); escribir("la resta es:",resta); escribir("la multiplicación es:", multiplicación); FinAlgoritmo </pre> | <pre> graph TD Start([Algoritmo sin titulo]) --> I1[/('Dame 2 números:')/] I1 --> P1[N1, N2] P1 --> P2[suma ← N1+N2] P2 --> P3[resta ← N1-N2] P3 --> P4[multiplicacion ← N1*N2] P4 --> O1[/('la suma es:', suma)/] O1 --> O2[/('la resta es:', resta)/] O2 --> O3[/('la multiplicacion es...')/] O3 --> End([FinAlgoritmo]) </pre> |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|--|---|--|
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() { float base, altura, area; printf("Dame la base y altura:"); scanf("%f%f", &base, &altura); area=(base*altura)/2; printf("El area es:%f", area); return 0; }</pre> | <pre>Algoritmo sin_titulo escribir("Dame la base y altura:"); leer base, altura; area=(base*altura)/2; escribir("El área es:", area); FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> Input[/('Dame la base y altura...')/] Input --> Process[/base, altura/] Process --> Calc[area ← (base*altura)/2] Calc --> Output[/('El area es:', area)/] Output --> End([FinAlgoritmo]) </pre> |
| <pre>#include <iostream> #include <stdio.h> #include <math.h> using namespace std; int main() {float A, B, C, X1, X2; printf("dame el valor cuadrático:"); scanf("%f", &A); printf("dame el valor lineal:"); scanf("%f", &B); printf("dame el valor independiente:"); scanf("%f", &C); X1=(-B+(sqrt(B*B-(4*A*C)))/2; X2=(-B-(sqrt(B*B-(4*A*C)))/2; printf("las soluciones son: %f%f", X1, X2); return 0; }</pre> | <pre>Algoritmo sin_titulo escribir("dame el valor cuadrático:"); leer A; escribir("dame el valor lineal:"); leer B; escribir("dame el valor independiente:"); leer C; X1=(-B+(sqrt(B*B-(4*A*C)))/2; X2=(-B-(sqrt(B*B-(4*A*C)))/2; escribir("las soluciones son:", X1, X2); FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> Input1[/('dame el valor cuadrático:')/] Input1 --> A[/A/] A --> Input2[/('dame el valor lineal:')/] Input2 --> B[/B/] B --> Input3[/('dame el valor independiente:')/] Input3 --> C[/C/] C --> Calc1[X1 ← (-B+(sqrt[B*B-(4*...)])] Calc1 --> Calc2[X2 ← (-B-(sqrt[B*B-(4*...)])] Calc2 --> Output[/('las soluciones son: '...')/] Output --> End([FinAlgoritmo]) </pre> |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|---|---|-------------------|
| <pre> #include <iostream> #include <stdio.h> using namespace std; int main() { float N1,N2,N3,N4,total; printf("Cantidad de lápices:"); scanf("%f",&N1); printf("Cantidad de gomas:"); scanf("%f",&N2); printf("Cantidad de libretas:"); scanf("%f",&N3); printf("Cantidad de plumas:"); scanf("%f",&N4); total=(N1*5)+(N2*2)+(N3*10)+(N4*3.5); printf("costo total:%f",total); return 0; } </pre> | <p>Algoritmo sin_titulo escribir("Cantidad de lápices:"); leer N1; escribir("Cantidad de gomas:"); leer N2; escribir("Cantidad de libretas:"); leer N3; escribir("Cantidad de plumas:"); leer N4;</p> <p>total=(N1*5)+(N2*2)+(N3*10)+(N4*3.5) ; escribir("costo total:",total); FinAlgoritmo</p> | |
| <pre> #include <iostream> #include <stdio.h> using namespace std; int main() {float ft, lb, m, kg; printf("Dame la longitud en pies:"); scanf("%f", &ft); printf("Dame el peso en libras:"); scanf("%f", &lb); m=ft*0.09290; kg=lb*0.45359; printf("la longitud en metros es:%f", m); printf("el peso en kilos es:%f", kg); return 0; } </pre> | <p>Algoritmo sin_titulo escribir("Dame la longitud en pies:"); leer ft; escribir("Dame el peso en libras:"); leer lb; m=ft*0.09290; kg=lb*0.45359; escribir("la longitud en metros es:", m); escribir("el peso en kilos es:",kg) FinAlgoritmo</p> | |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|--|---|--|
| <pre>#include <iostream> #include <stdio.h> #include <math.h> using namespace std; int main(){ int B,P,R; printf("Base:") scanf("%d",&B) printf("Potencia:") scanf("%d",&P); R=pow(B,P); printf("Resultado:%d",R); return 0;}</pre> | <pre>Algoritmo sin_titulo escribir("Base:"); leer B; escribir("Potencia:"); leer P; R=pow(B,P); escribir("Resultado:",R); FinAlgoritmo</pre> | <pre>graph TD Start([Algoritmo sin_titulo]) --> Input1[/Base:/] Input1 --> Process1{B} Process1 --> Input2[/Potencia:/] Input2 --> Process2{P} Process2 --> Process3[R ← pow[B,P]] Process3 --> Output1[/Resultado:/] Output1 --> End([FinAlgoritmo])</pre> |
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main(){ float r,h,a,v; printf("radio:"); scanf("%f",&r); printf("altura:"); scanf("%f",&h); a=2*3.1416*r*h; v=3.1416*r*r*h; printf("Area:%f",a); printf("Volumen:%f",v); return 0;}</pre> | <pre>Algoritmo sin_titulo escribir("radio:"); leer r; escribir("altura:"); leer h; a=2*3.1416*r*h; v=3.1416*r*r*h; escribir("Área:",a) escribir ("Volumen:",v) FinAlgoritmo</pre> | <pre>graph TD Start([Algoritmo sin_titulo]) --> Input1[/radio:/] Input1 --> Process1{r} Process1 --> Input2[/altura:/] Input2 --> Process2{h} Process2 --> Process3[a ← 2*3.1416*r*h] Process3 --> Process4[v ← 3.1416*r*r*h] Process4 --> Output1[/Area:/] Output1 --> Output2[/Volumen:/] Output2 --> End([FinAlgoritmo])</pre> |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|--|--|---|
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() {float L, an, A, C; printf("Dame el largo del terreno:"); scanf("%f", &L); printf("Dame el ancho del terreno:"); scanf("%f", &an); A=L*an; C=A*250; printf("El costo es:%f", C); return 0; }</pre> | <pre>Algoritmo sin_titulo escribir("Dame el largo del terreno:"); leer L; escribir("Dame el ancho del terreno:"); leer an; A=L*an; C=A*250; escribir("El costo es:%", C); FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> L[/L/] L --> an[/an/] an --> A["A ← L*an"] A --> C["C ← A*250"] C --> PrintC["('El costo es:%', C)"] PrintC --> End([FinAlgoritmo]) </pre> |
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() {float prec, cant, C; printf("Dame el precio del articulo:"); scanf("%f", &prec); printf("Dame la cantidad entregada por el cliente:"); scanf("%f", &cant); C=cant-prec; printf("el cambio es:%f", C); return 0; }</pre> | <pre>Algoritmo sin_titulo escribir("Dame el precio del articulo:"); leer prec; escribir("Dame la cantidad entregada por el cliente:"); leer cant; C=cant-prec; escribir("el cambio es:", C) FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> prec[/prec/] prec --> cant[/cant/] cant --> C["C ← cant-prec"] C --> PrintC["('el cambio es:', C)"] PrintC --> End([FinAlgoritmo]) </pre> |
| <pre>#include <iostream> #include<stdio.h> using namespace std; int main() { char Marca[10],Modelo[10]; printf("Dame la marca del coche:"); scanf("%s",&Marca); printf("Dame el modelo:"); scanf("%s",&Modelo); printf("Tu auto es:%s",Modelo); printf(", %s",Marca); return 0; }</pre> | <pre>Algoritmo sin_titulo escribir("Dame la marca del coche:"); leer Marca; escribir("Dame el modelo:"); leer Modelo; escribir("Tu auto es:",Modelo); escribir(", ",Marca); FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> Marca[/Marca/] Marca --> Modelo[/Modelo/] Modelo --> PrintModel["('Tu auto es:', Modelo)"] PrintModel --> PrintBrand["(', ', Marca)"] PrintBrand --> End([FinAlgoritmo]) </pre> |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|--|--|---|
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() {float km, m; printf("Dame una medida en kilometros:"); scanf("%f", &km); m=km/1.609; printf("En millas es:%f", m); return 0; }</pre> | <pre>Algoritmo sin_titulo escribir("Dame una medida en kilometros:"); leer km; m=km/1.609; escribir("En millas es:", m) FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> Input[/('Dame una medida en k...')/] Input --> Var[km] Var --> Process[m ← km/1.609] Process --> Output[/('En millas es:', m)/] Output --> End([FinAlgoritmo]) </pre> |
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() { int cuadrado; cuadrado=243*243; printf("El cuadrado del número 243 es:%d",cuadrado); return 0; }</pre> | <pre>Algoritmo sin_titulo cuadrado=243*243; escribir("El cuadrado del número 243 es:",cuadrado); FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> Process[cuadrado ← 243*243] Process --> Output[/('El cuadrado del número...')/] Output --> End([FinAlgoritmo]) </pre> |
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() {float N, pul; printf("Dame los centímetros:"); scanf("%f", &N); pul=N/2.54; printf("La medida en pulgadas es:%f", pul); return 0; }</pre> | <pre>Algoritmo sin_titulo escribir("Dame los centímetros:"); leer N; pul=N/2.54; escribir("La medida en pulgadas es:", pul) FinAlgoritmo</pre> | <pre> graph TD Start([Algoritmo sin_titulo]) --> Input[/('Dame los centímetros:')/] Input --> Var[N] Var --> Process[pul ← N/2.54] Process --> Output[/('La medida en pulgada...')/] Output --> End([FinAlgoritmo]) </pre> |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|--|---|---|
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() { float A,B,C,X; printf("valor de A:"); scanf("%f",&A); printf("valor de B:"); scanf("%f",&B); printf("valor de C:"); scanf("%f",&C); X=(C-B)/A; printf("Valor de X:%f",X); return 0; }</pre> | <pre>Algoritmo sin_titulo Escribir ('valor de A:') Escribir ('valor de B:') Leer A Leer B Escribir ('valor de C:') Leer C X <- (C-B)/A Escribir ('Valor de X:',X) FinAlgoritmo</pre> | <pre>graph TD Start([Algoritmo sin_titulo]) --> A1[('valor de A:')] A1 --> P1[A] P1 --> A2[('valor de B:')] A2 --> P2[B] P2 --> A3[('valor de C:')] A3 --> P3[C] P3 --> P4[X ← (C-B)/A] P4 --> A4[('Valor de X:', X)] A4 --> End([FinAlgoritmo])</pre> |
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() {int num; printf("Dame un numero:"); scanf("%d",&num); if (num > 0){ printf("Es positivo"); } return 0; }</pre> | <pre>Algoritmo sin_titulo Escribir ("Dame un numero."); leer num; si(num > 0) escribir("Es positivo"); FinSi FinAlgoritmo</pre> | <pre>graph TD Start([Algoritmo sin_titulo]) --> A1[('Dame un numero:')] A1 --> P1[num] P1 --> D1{num > 0} D1 -- V --> A2[('Es positivo')] D1 -- F --> End([FinAlgoritmo]) A2 --> End</pre> |
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main(){ int N1; printf("Dame un numero: "); scanf("%d",&N1); if (N1>0) { printf("Es positivo"); } else{ printf("Es negativo"); } return 0; }</pre> | <pre>Algoritmo sin_titulo Escribir ("Dame un numero: "); leer N1; si (N1>0) escribir("Es positivo"); sino escribir("Es negativo"); finsi FinAlgoritmo</pre> | <pre>graph TD Start([Algoritmo sin_titulo]) --> A1[('Dame un numero:')] A1 --> P1[N1] P1 --> D1{N1 > 0} D1 -- V --> A2[('Es positivo')] D1 -- F --> A3[('Es negativo')] A2 --> End([FinAlgoritmo]) A3 --> End</pre> |

| CODIGO | PSEUDOCODIGO | DIAGRAMA DE FLUJO |
|---|--|--|
| <pre>#include <iostream> #include <stdio.h> using namespace std; int main() {int N; printf("Dame un número."); scanf("%d", &N); if (N==0){ printf("Es cero");} else{ printf("No es cero");} return 0; }</pre> | <pre>Algoritmo sin_titulo Escribir ("Dame un número."); leer N; si (N==0) escribir("Es cero"); sino escribir("No es cero"); finsi FinAlgoritmo</pre> | <pre>graph TD Start([Algoritmo sin_titulo]) --> Input[/Dame un número:/] Input --> Decision{N==0} Decision -- F --> Output1[/No es cero/] Decision -- V --> Output2[/Es cero/] Output1 --> End([FinAlgoritmo]) Output2 --> End</pre> |
| <pre>#include <iostream> #include <stdio.h> int main(){ float Cal1,Cal2,Cal3,Prom; printf("Dame tus calificaciones."); scanf("%f%f%f",&Cal1,&Cal2,&Cal3); Prom=(Cal1+Cal2+Cal3)/3; if(Prom>=6){ printf(" Aprobado"); } else{ printf(" Reprobado"); } return 0; }</pre> | <pre>Algoritmo sin titulo escribir ("Dame tus calificaciones."); leer Cal1,Cal2,Cal3; Prom=(Cal1+Cal2+Cal3)/3; si(Prom>=6) escribir(" Aprobado"); sino escribir(" Reprobado"); finsi FinAlgoritmo</pre> | <pre>graph TD Start([Algoritmo sin_titulo]) --> Input1[/Dame tus calificaciones.../] Input1 --> Input2[/Cal1, Cal2, Cal3/] Input2 --> Process[Prom = (Cal1+Cal2+Cal3)/3] Process --> Decision{Prom >= 6} Decision -- F --> Output1[/Reprobado/] Decision -- V --> Output2[/Aprobado/] Output1 --> End([FinAlgoritmo]) Output2 --> End</pre> |