

Implementación de la estructura de control lógica secuencial

Estudiante:   
Cristian Miranda  
Módulo o Unidad de Aprendizaje:  
Programación de Sistemas informáticos  
Fecha: 04-06-23  
otro: Otros

Actividad 3

## **Introducción**

¡Bienvenido al mundo de la resolución de problemas informáticos! En este documento, te ofrezco la oportunidad de adentrarte en el fascinante campo de la construcción de algoritmos computacionales. ¿Qué te parecería utilizar pseudocódigos que combinan lógica matemática con los componentes de software necesarios para resolver problemas prácticos?

Nuestro enfoque se basa en aplicar buenas prácticas de codificación para asegurar la eficiencia y robustez de tus algoritmos. Podrás analizar, diseñar e implementar soluciones para una amplia variedad de aplicaciones, donde podrás determinar aspectos específicos a calcular y explorar diferentes figuras geométricas.

Además, te invito a expandir tus habilidades investigativas, ya que en ocasiones deberás buscar y descubrir datos relacionados con las figuras que no se mencionan explícitamente en el enunciado del problema. De esta manera, podrás abordar desafíos más complejos y obtener soluciones aún más completas.

¡Únete y desata tu creatividad en el mundo de los algoritmos computacionales!

Tabla de contenido

[**Introducción** 1](#_Toc136818254)

[**Tabla de Figuras** 2](#_Toc136818255)

[**Implementación de la estructura de control lógica secuencial** 4](#_Toc136818256)

[13. Ortoedro 4](#_Toc136818257)

[14. Cono 10](#_Toc136818258)

[15. Prisma 16](#_Toc136818259)

[**Conclusión** 22](#_Toc136818260)

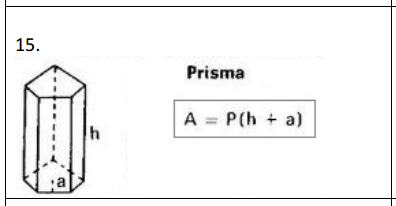
## **Tabla de Figuras**

Imagen que contiene Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente con confianza baja



**Desarrollo**

Las figuras geométricas son elementos fundamentales en el estudio de la geometría, una rama de las matemáticas que se dedica al análisis y la descripción de las propiedades y relaciones espaciales de los objetos. Estas figuras se encuentran presentes en nuestro entorno cotidiano y son de vital importancia en diversos campos, como la arquitectura, el diseño, la ingeniería y la física.

En geometría, las figuras geométricas se clasifican en dos dimensiones (2D) y tres dimensiones (3D). Las figuras 2D son aquellas que tienen solo largo y ancho, mientras que las figuras 3D tienen largo, ancho y altura, lo que les confiere una mayor complejidad y profundidad.

Entre las figuras geométricas más comunes en el plano 2D se encuentran el círculo, el triángulo, el cuadrado, el rectángulo, el rombo, el trapecio, el pentágono, el hexágono y el octógono, entre otros. Cada una de estas figuras tiene características específicas que las distinguen, como la longitud de sus lados, el número de ángulos y la simetría presente.

En el espacio 3D, encontramos figuras como la esfera, el cilindro, el cono, el cubo, la pirámide, el prisma y el toro, entre otras. Estas figuras tienen una mayor variedad de características, como la presencia de caras, aristas y vértices, así como la posibilidad de calcular su volumen y área superficial.

El estudio de las figuras geométricas implica el análisis de sus propiedades geométricas, como el perímetro, el área, el volumen, la simetría y la congruencia. Además, se exploran conceptos como los ángulos, las rectas y los planos que intervienen en la definición y descripción de estas figuras.

La geometría es una herramienta poderosa que nos permite comprender y representar el mundo que nos rodea. Desde la construcción de edificios hasta la creación de modelos virtuales en la informática, el conocimiento de las figuras geométricas y sus propiedades es esencial para resolver problemas prácticos y desarrollar soluciones innovadoras.

En resumen, las figuras geométricas son elementos fundamentales en el estudio de la geometría y desempeñan un papel crucial en numerosos campos de aplicación. Su comprensión nos permite analizar y describir las propiedades espaciales de los objetos, así como desarrollar algoritmos y soluciones eficientes en el ámbito de la informática y otras disciplinas relacionadas.

## **Implementación de la estructura de control lógica secuencial**

## 13. Ortoedro

**Analizar, diseñar, evaluar e implementar un algoritmo para calcular el área, perímetro y diagonal de un Ortoedro.**

Imagen que contiene Diagrama

Descripción generada automáticamente

X = 2(ab + ac + bc)

V = a\*b\*c

D = √(a\*a + b\*b + c\*c)

**Descripción Narrativa**

Declara a, b, c, D, X, V

Obtener a

Obtener b

Obtener c

Si ((a>0) y (a<10) y (b>0) y (b<10) y (c>0) y (c<10))

Calcular X = 2(ab + ac + bc)

V = a\*b\*c

D = √(a\*a + b\*b + c\*c)

Mostrar X

Mostrar V

Mostrar D

Entonces

Mostrar “Valores Invalidos”

**Diagrama de flujo de detalle**

              Inicio

                  a, b, c, D, X, V

“Calcula área, volumen y diagonal de un Ortoedro”

“Ingresa a”

a

b

“Ingresa b”

c

“Ingresa c”

((a>0) AND (a<10))

AND

((b>0) AND (b<10))

AND

((c>0) y (c<10))

No Si

X = 2(ab + ac + bc)

“Valores Invalidos”

V = a\*b\*c

D = √(a\*a + b\*b + c\*c)

“El área del Ortoedro es”, X

“La diagonal del Ortoedro es”, D

“El volumen del Ortoedro es”, V

**Pseudocodigo**

Inicio\_programa

declara a, b, c, D, X, V

escribe “Calcula el área, diagonal y volumen de un ortoedro”

escribe “Ingresa el lado a”

lee a

escribe “Ingresa el lado b”

lee b

escribe “Ingresa el lado c”

lee c

si ((a>0) y (a<10)) y ((b>0) y (b<10)) y ((c>0) y (c<10))

X = 2(ab + ac + bc)

V = a\*b\*c

D = √(a\*a + b\*b + c\*c)

escribe “El área del Ortoedro es”, X

escribe “La diagonal del Ortoedro es”, D

escribe “El volumen del Ortoedro es”, V

de\_lo\_contrario

escribe “Valores Inválidos”

fin\_del\_si

fin\_del\_programa

**Diagrama de chapin**

Inicio\_programa

declara a, b, c, D, X, V

escribe “Calcula el área, diagonal y volumen de un ortoedro”

escribe “Ingresa el lado a”

lee a

escribe “Ingresa el lado b”

lee b

escribe “Ingresa el lado c”

lee c

Si ((a>0) y (a<10)) y ((b>0) y (b<10)) y ((c>0) y (c<10))

Si No

X = 2(ab + ac + bc) Escribe “Valores Inválidos”

V = a\*b\*c

D = √(a\*a + b\*b + c\*c)

escribe “El área del Ortoedro es”, X

escribe “La diagonal del Ortoedro es”, D

escribe “El volumen del Ortoedro es”, V

fin\_del\_si

fin\_del\_programa

**Código en C**

#include <stdio.h>

#include <math.h>

void main ()

{

int a, b, c;

float D, X, V;

printf ("Calcula el área, diagonal y volumen de un ortoedro \n");

printf ("Ingresa el lado a \n");

scanf ("%i",&a);

printf ("Ingresa el lado b \n");

scanf ("%i",&b);

printf ("Ingresa el lado c \n");

scanf ("%i",&c);

if ((a>0) && (a<10) && (b>0) && (b<10) && (c>0) && (c<10))

{

X = 2\*((a\*b) + (a\*c) + (b\*c));

V = a\*b\*c;

D = sqrt(pow(a,2) + pow(b,2) + pow(c,2));

printf("El area del Ortoedro es %f \n", X);

printf("La diagonal del Ortoedro es %f \n", D);

printf("El volumen del Ortoedro es %f \n", V);

}

else

{

printf("Valores Invalidos \n");

}

getch();

}

## 14. Cono

**Analizar, diseñar, evaluar e implementar un algoritmo para calcular el área y volumen de un Cono.**

r = radio del cono

Diagrama

Descripción generada automáticamente con confianza bajaX = área del cono

V = volumen del cono

PI = 3.14

h = altura del cono

g = generatriz del cono

**Descripción Narrativa**

Declara X, V, r, h, g

Obtener h

Obtener g

Si ((h>0) y (h<10) y (g>0) y (g<10))

Calcular

                 r = √g²-h²

                 X = PI\*r(g+r)

                 V = PI\*r²\*h

3

Mostrar r

Mostrar X

Mostrar V

Entonces

Mostrar “Valores Inválidos”

**Diagrama de Flujo de detalle**

                  X, V, r, h, g

“Calcula área y volumen de un Cono”

“Ingresa h”

h

g

“Ingresa g”

((h>0) AND (h<10))

AND

((g>0) AND (g<10))

No Si

r = √g²-h²

X = PI\*r(g+r)

“Valores Invalidos”

V = PI\*r²\*h

3

“El radio del Cono es”, r

“El área del Cono es”, X

“El volumen del Cono es”, V

**Pseudocodigo**

Inicio\_programa

declara X, V, r, h, g

escribe “Calcular el área y volumen de un Cono”

escribe “Ingresar el valor de altura”

lee h

escribe “Ingresar el valor de generatriz”

lee g

Si ((h>0) y (h<10) y (g>0) y (g<10))

                 r = √g²-h²

                 X = PI\*r(g+r)

                 V = PI\*r²\*h

3

escribe “El radio del Cono es”, r

escribe “El área del Cono es”, X

escribe “El volumen del Cono es”, V

de\_lo\_contrario

escribe “Valores Inválidos”

fin\_del\_si

fin\_del\_programa

**Diagrama de chapin**

Inicio\_programa

declara X, V, r, h, g

escribe “Calcula el área y volumen de un Cono”

escribe “Ingresa el valor de altura”

lee h

escribe “Ingresa el valor de generatriz”

lee g

Si ((h>0) y (h<10) y (g>0) y (g<10))

Si No

                 r = √g²-h²

                 X = PI\*r(g+r) escribe “Valores inválidos”

                 V = PI\*r²\*h

3

escribe “El radio del Cono es”, r

escribe “El área del Cono es”, X

escribe “El volumen del Cono es”, V

fin\_del\_si

fin\_del\_programa

**Código en C**

#include <stdio.h>

#include <math.h>

void main ()

{

int h, g;

float r, X, V, PI;

PI = 3.14;

h=g=0;

r=X=V=0.0;

printf ("Calcula el área y volumen de un cono \n");

printf ("Ingresa el valor de altura \n");

scanf ("%i",&h);

printf ("Ingresa el valor de generatriz \n");

scanf ("%i",&g);

if ((h>0) && (h<10) && (g>0) && (g<10))

{

r = sqrt(pow(g,2)-(pow(h,2)));

X = PI\*r\*(g+r);

V = (PI\*(pow(r,2))\*h)/3;

printf("El radio del Cono es %f \n", r);

printf("El area del Cono es %f \n", X);

printf("El volumen del Cono es %f \n", V);

}

else

{

printf("Valores Invalidos \n");

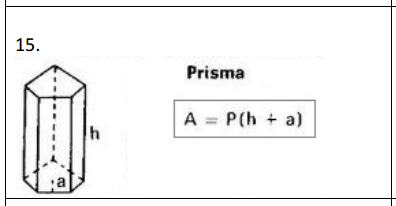
}

getch();

}

## 15. Prisma

**Analizar, diseñar, evaluar e implementar un algoritmo para calcular el área y volumen de un Prisma pentagonal.**

P = perímetro del prisma

X = área del prisma

V = volumen del prisma

Ab = área de la base

l = lado del prisma

h = altura del prisma

a = apotema del prisma

**Descripción Narrativa**

Declara P, X, V, Ab, l, h, a

Obtener l

Obtener h

Obtener a

Si ((l>0) y (l<10)) y (h>0) y (h<10) y (a>0) y (a<10))

Calcular

P = l\*5

Ab = P\*a/2

V = Ab\*h

X = P\*(h+a)

Mostrar P

Mostrar Ab

Mostrar V

Mostrar X

Entonces

Mostrar “Valores Inválidos”

**Diagrama de Flujo de detalle**

                  P, X, V, Ab, l, h, a

“Calcula área y volumen de un Prisma”

“Ingresa l”

l

h

“Ingresa h”

a

“Ingresa a”

((l>0) AND (l<10))

AND

((h>0) AND (h<10))

AND

((a>0) AND (a<10))

No Si

P = l\*5

Ab= P\*a/2

“Valores Invalidos”

V = Ab\*h

X = P\*(h+a)

“El área del Prisma es”, X

“El volumen del Prisma es”, V

**Pseudocodigo**

Inicio\_programa

declara P, X, V, Ab, l, h, a

escribe “Calcular el área y volumen de un Prisma”

escribe “Ingresar el lado”

lee l

escribe “Ingresar la altura”

lee h

escribe “Ingresar la apotema”

lee a

Si ((l>0) y (l<10)) y (h>0) y (h<10) y (a>0) y (a<10))

P = l\*5

Ab = P\*a/2

V = Ab\*h

X = P\*(h+a)

escribe “El área del Prisma es”, X

escribe “El volumen del Prisma es”, V

de\_lo\_contrario

escribe “Valores Inválidos”

fin\_del\_si

fin\_del\_programa

**Diagrama de chapin**

Inicio\_programa

declara P, X, V, Ab, l, h, a

escribe “Calcula el área y volumen de un Prisma”

escribe “Ingresar el lado”

lee l

escribe “Ingresar la altura”

lee h

escribe “Ingresar la apotema”

lee a

Si ((l>0) y (l<10)) y (h>0) y (h<10) y (a>0) y (a<10))

Si No

P = l\*5

Ab = P\*a/2 escribe “Valores invalidos”

V = Ab\*h

X = P\*(h+a)

escribe “El área del Prisma es”, X

escribe “El volumen del Prisma es”, V

fin\_del\_si

fin\_del\_programa

**Código en C**

#include <stdio.h>

void main ()

{

int l, h, a, P;

float Ab, X, V;

l=h=a=P= 0;

Ab=X=V=0.0;

printf ("Calcula el área y volumen de un Prisma \n");

printf ("Ingresa el valor de lado \n");

scanf ("%i",&l);

printf ("Ingresa el valor de la altura \n");

scanf ("%i",&h);

printf ("Ingresa el valor de la apotema \n");

scanf ("%i",&a);

if ((l>0) && (l<10) && (h>0) && (h<10) && (a>0) && (a<10))

{

P = l\*5;

Ab = P\*a/2;

V = Ab\*h;

X = P\*(h+a);

printf("El perimetro del Prisma es %i \n", P);

printf("El area del Prisma es %f \n", X);

printf("El volumen del Prisma es %f \n", V);

}

else

{

printf("Valores Invalidos \n");

}

getch();

}

## **Conclusión**

La implementación de la estructura de control lógica secuencial en figuras geométricas es de gran importancia debido a su capacidad para organizar y ejecutar de manera secuencial las instrucciones necesarias para dibujar y manipular figuras geométricas en un programa.

Entre algunos aspectos:

1. Orden y coherencia: La estructura de control lógica secuencial nos permite organizar las instrucciones de manera ordenada y coherente. Al seguir una secuencia lógica, podemos dibujar las diferentes partes de una figura geométrica paso a paso, asegurando que cada instrucción se ejecute en el orden correcto.
2. Construcción progresiva: Al utilizar la estructura de control lógica secuencial, podemos construir figuras geométricas progresivamente. Comenzamos con una forma básica y luego vamos agregando detalles y características adicionales en cada paso. Esto nos permite crear figuras más complejas de manera estructurada y comprensible.
3. Control preciso: La estructura de control secuencial nos permite tener un control preciso sobre cada paso en la construcción de una figura geométrica. Podemos ajustar y modificar fácilmente las coordenadas, los ángulos y otras propiedades de la figura en cada paso, lo que nos brinda flexibilidad y precisión en el dibujo.
4. Reutilización de código: Al utilizar la estructura de control secuencial, podemos encapsular y reutilizar fragmentos de código relacionados con el dibujo de figuras geométricas. Podemos definir funciones o métodos para dibujar formas básicas, y luego llamar a estas funciones en secuencia para construir figuras más complejas. Esto facilita el mantenimiento del código y promueve la reutilización de código en futuros proyectos.

En resumen, la implementación de la estructura de control lógica secuencial en figuras geométricas nos permite organizar, construir y controlar las instrucciones necesarias para dibujar y manipular figuras geométricas en un programa. Proporciona orden, construcción progresiva, control preciso y reutilización de código. Al utilizar esta estructura de control de manera efectiva, podemos crear programas que generen figuras geométricas de manera estructurada y precisa.