

Diagrama de flujo pseudocódigo sintaxis

**Lenguajes de
Programación**

Aritmética
conversiones
prácticas clase 2



Introducción Programas de :

1

Aritméticos

Suma, Resta,
Multiplicación, División, Potencias, Raíz.

2

Lógicos

AND (&&), OR (||), NOT (!).

3

Relacionales

== igual a, (!=) distinto, (>) Mayor
que, (<) Menor que, (>=) Mayor o
igual que, (<=) Menor o igual que.

4

Geométricos (Área, Radio)

Cuadrado, Rectángulo, Círculo,
Pirámide.

5

Conversiones

Binario, Octal, Hexadecimal

Programas Aritméticos



Aritmética

Suma

1. Diagrama de flujo

```
Inicio
|
v
Leer a
|
v
Leer b
|
v
suma = a + b
|
v
Mostrar "el resultado es :", suma
|
v
Fin
```

2. Pseudocódigo

```
Inicio
    Escribir "ingrese un
valor :"
    Leer a
    Escribir "ingrese otro
valor :"
    Leer b
    suma ← a + b
    Escribir "el resultado es
:", suma
Fin
```

3. Sintaxis Básica

```
a=float(input("ingresa un
valor :"))
b=float(input("ingrese otro
valor :"))
suma=a+b
print("el resultado es
:",suma)
```

Aritmética

Resta

1. Diagrama de flujo

```
Inicio
|
v
Leer a
|
v
Leer b
|
v
resta = a - b
|
v
Mostrar "el resultado es :", resta
|
v
Fin
```

2. Pseudocódigo

```
Inicio
  Escribir "ingrese un valor :"
  Leer a
  Escribir "ingrese otro valor :"
  Leer b
  resta ← a - b
  Escribir "el resultado es :",
  resta
Fin
```

3. Sintaxis Básica

```
a=float(input("ingresa un
valor :"))
b=float(input("ingrese otro
valor :"))
resta=a-b
print("el resultado es
:",resta)
```

Aritmética

Multiplicación

1. Diagrama de flujo

```
Inicio
|
v
Leer a
|
v
Leer b
|
v
multiplicación = a * b
|
v
Mostrar "el resultado es :",
multiplicación
|
v
Fin
```

2. Pseudocódigo

```
Inicio
  Escribir "ingrese un valor
  :."
  Leer a
  Escribir "ingrese otro
  valor :."
  Leer b
  multiplicación ← a * b
  Escribir "el resultado es
  :", multiplicación
Fin
```

3. Sintaxis Básica

```
a=float(input("ingresa un
valor :"))
b=float(input("ingrese otro
valor :"))
multiplicación=a*b
print("el resultado es
:",multiplicación)
```

Aritmética

División

1. Diagrama de flujo

```
Inicio
|
v
Leer a
|
v
Leer b
|
v
división = a / b
|
v
Mostrar "El resultado es: ", división
|
v
Fin
```

2. Pseudocódigo

```
Inicio
  Escribir "Ingrese un valor: "
  Leer a
  Escribir "Ingrese otro valor: "
  Leer b
  división ← a / b
  Escribir "El resultado es: ",
  división
Fin
```

3. Sintaxis Básica

Solicita al usuario
dos valores y
calcula la división

```
a=float(input("ingresa un  
valor : "))  
b=float(input("ingrese otro  
valor : "))  
división= a/b  
print("El resultado es :  
{división}")
```

Aritmética

Potencias

1. Diagrama de flujo

```
Inicio
|
v
Leer a
|
v
Leer b
|
v
operación = a ** b
|
v
Mostrar "{a} elevado a la potencia
{b} es {operación}"
|
v
Fin
```

2. Pseudocódigo

```
Inicio
  Escribir "Ingresa un
  número : "
  Leer a
  Escribir "Ingresa la
  potencia a elevar : "
  Leer b
  operación ← a ^ b
  Escribir a, " elevado a la
  potencia ", b, " es ",
  operación
Fin
```

3. Sintaxis Básica

```
a=int(input("Ingresa un
número : "))
b=int(input("ingresa la
potencia a elevar : "))
operación=a**b
print("{a} elevado a la
potencia {b} es {operación}")
```


Aritmética

Raíz

1. Diagrama de flujo

```
Inicio
|
v
Leer numero
|
v
raíz = pow(numero, 0.5)
|
v
Mostrar "La raíz cuadrada
de", número, "es", raíz
|
v
Fin
```

2. Pseudocódigo

```
Inicio
  Escribir "Introduce un
número: "
  Leer numero
  raíz ← pow(numero, 0.5)
  Escribir "La raíz cuadrada
de ", número, " es ", raíz
Fin
|
```

3. Sintaxis Básica

```
numero =
float(input("Introduce un
número: "))

# Calcula la raíz cuadrada
usando pow (potencia 0.5)
raíz = pow(numero, 0.5)

# Muestra el resultado
print("La raíz cuadrada de
{número} es {raíz}")
```

Programas Lógicos



Lógicos

AND

1. Diagrama de flujo

```
a > 0
|
v
b > 0
|
v
(a > 0) AND (b > 0)?
|
v
Verdadero -> Mostrar
"Ambos son positivos"
Falso -> Mostrar "Al
menos uno NO es positivo"
```

2. Pseudocódigo

```
Inicio
  Leer a
  Leer b
  Si (a > 0) Y (b > 0)
  Entonces
    Escribir "Ambos son
    positivos"
  SiNo
    Escribir "Al menos
    uno NO es positivo"
Fin
```

3. Sintaxis Básica

```
#Pide al usuario que ingrese Dos
números
a = int(input("Ingresa un número: "))
b = int(input("Ingresa otro número:
"))
#Valida si los dos números son
mayores a 0
if a > 0 and b > 0:
    #si es verdadero imprime que
    ambos son positivos
    print("Ambos son positivos")
    #De no cumplirse alguno de los
    dos se imprime "al menos uno no es
    positivo"
else:
    print("Al menos uno NO es
    positivo")
```

Lógicos

OR

1. Diagrama de flujo

```
a % 2 == 0
|
v
b % 2 == 0
|
v
(a es par) OR (b es par)?
|
v
Verdadero -> Mostrar "Al menos
uno es par"
Falso -> Mostrar "Ninguno es
par"
```

2. Pseudocódigo

```
Inicio
    Leer a
    Leer b
    Si (a es par) O (b es par)
Entonces
    Escribir "Al menos
uno es par"
SiNo
    Escribir "Ninguno es
par"
Fin
```

3. Sintaxis Básica

```
#Pide al usuario que ingrese dos
números
a = int(input("Ingresa un número: "))
b = int(input("Ingresa otro número:
"))

# Verifica si al menos uno de los dos
números es par usando el operador OR
if a % 2 == 0 or b % 2 == 0:
    # Si al menos uno es par, muestra
este mensaje
    print("Al menos uno es par")
else:
    # Si ninguno es par, muestra este
mensaje
    print("Ninguno es par")
```

Lógicos

NOT

1. Diagrama de flujo

```
a < 0
|
v
NOT(a < 0)?
|
v
Verdadero -> Mostrar "El
número NO es negativo"
Falso -> Mostrar "El número
es negativo"
```

2. Pseudocódigo

```
Inicio
  Leer a
  Si NO(a < 0) Entonces
    Escribir "El número NO es
    negativo"
  SiNo
    Escribir "El número es
    negativo"
Fin
```

3. Sintaxis Básica

```
#Pide al usuario que ingrese Do
números
a = int(input("Ingresa un número: "))

# Verifica si el número NO es
negativo usando el operador NO
if not (a < 0):
    # Si el número NO es negativo (es
    decir, es cero o positivo), muestra
    este mensaje
    print("El número NO es negativo")
else:
    # Si el número es negativo
    muestra este mensaje
    print("El número es negativo")
```

Programas Relacionales



Relacionales

(= a que a =)

1. Diagrama de flujo

```
Inicio
|
v
Leer a
|
v
Leer b
|
v
¿a == b?
|---Sí--> Mostrar "a es igual a b"
|---No--> Mostrar "a no es igual a b"
|
v
¿a != b?
|---Sí--> Mostrar "a es diferente de b"
|---No--> Mostrar "a no es diferente de b"
|
v
¿a > b?
|---Sí--> Mostrar "a es mayor que b"
|---No--> Mostrar "a no es mayor que b"
|
v
¿a < b?
|---Sí--> Mostrar "a es menor que b"
|---No--> Mostrar "a no es menor que b"
|
v
¿a >= b?
|---Sí--> Mostrar "a es mayor o igual que b"
|---No--> Mostrar "a no es mayor o igual que b"
|
v
¿a <= b?
|---Sí--> Mostrar "a es menor o igual que b"
|---No--> Mostrar "a no es menor o igual que b"
|
v
Fin
```

2. Pseudocódigo

```
Inicio
  Escribir "Ingresa el valor de a: "
  Leer a
  Escribir "Ingresa el valor de b: "
  Leer b

  Si a == b Entonces
    Escribir a, " es igual a ", b
  Sino
    Escribir a, " no es igual a ", b

  Si a != b Entonces
    Escribir a, " es diferente de ", b
  Sino
    Escribir a, " no es diferente de ", b

  Si a > b Entonces
    Escribir a, " es mayor que ", b
  Sino
    Escribir a, " no es mayor que ", b

  Si a < b Entonces
    Escribir a, " es menor que ", b
  Sino
    Escribir a, " no es menor que ", b

  Si a >= b Entonces
    Escribir a, " es mayor o igual que ", b
  Sino
    Escribir a, " no es mayor o igual que ", b

  Si a <= b Entonces
    Escribir a, " es menor o igual que ", b
  Sino
    Escribir a, " no es menor o igual que ", b

Fin
```

3. Sintaxis Básica

```
a = int(input("Ingresa el valor de a: "))
b = int(input("Ingresa el valor de b: "))
```

```
# Operador de igualdad (==)
if a == b:
    print(f"{a} es igual a {b}")
else:
    print(f"{a} no es igual a {b}")
```

```
# Operador de desigualdad (!=)
if a != b:
    print(f"{a} es diferente de {b}")
else:
    print(f"{a} no es diferente de {b}")
```

```
# Operador mayor que (>)
if a > b:
    print(f"{a} es mayor que {b}")
else:
    print(f"{a} no es mayor que {b}")
```

```
# Operador menor que (<)
if a < b:
    print(f"{a} es menor que {b}")
else:
    print(f"{a} no es menor que {b}")
```

```
# Operador mayor o igual que (>=)
if a >= b:
    print(f"{a} es mayor o igual que {b}")
else:
    print(f"{a} no es mayor o igual que {b}")
```

```
# Operador menor o igual que (<=)
if a <= b:
    print(f"{a} es menor o igual que {b}")
else:
    print(f"{a} no es menor o igual que {b}")
```

Programas

Geométricos
(Área, Radio)



Valores geométricos : círculo

1. Diagrama de flujo

Diagrama de flujo (descripción paso a paso)
Inicio
Solicitar al usuario que ingrese el radio del círculo y guardar ese valor en r.
Asignar el valor de pi como 3.14.
Calcular el cuadrado del radio ($\text{cuadrado} = r^2$).
Calcular el área ($\text{area} = \pi * \text{cuadrado}$).
Mostrar el área del círculo.
Calcular el diámetro ($\text{diametro} = r * 2$).
Mostrar el diámetro.
Calcular el perímetro ($\text{perimetro} = 2 * \pi * r$).
Mostrar el perímetro.
Fin

2. Pseudocódigo

Inicio
 Escribir "Ingresa el radio del círculo:"
 Leer r
 $\pi \leftarrow 3.14$
 $\text{cuadrado} \leftarrow r^2$
 $\text{area} \leftarrow \pi * \text{cuadrado}$
 Escribir "El área del círculo es: ", area, "cm"
 $\text{diametro} \leftarrow r * 2$
 Escribir "El diámetro del círculo es: ", diametro, "cm"
 $\text{perimetro} \leftarrow 2 * \pi * r$
 Escribir "El perímetro del círculo es: ", perimetro, "cm"
Fin

3. Sintaxis Básica

```
r=int(input("ingresa el radio del círculo : "))
pi=3.14
cuadrado=pow(r,2)
area=pi*cuadrado
print(f"El area del círculo es: {area}cm")
diametro=r*2
print(f"El diametro del radio es: {diametro}cm")
perimetro =2*pi*r
print(f"El perimetro del radio es: {perimetro}cm²")
```

Valores geométricos : CUADRADO

1. Diagrama de flujo

1. Inicio
2. Solicitar al usuario ingresar la medida del lado del cuadrado.
3. Leer el valor ingresado y guardarlo en la variable lado1.
4. Calcular el área del cuadrado como: $\text{área} = \text{lado1} * \text{lado1}$.
5. Calcular el perímetro del cuadrado como: $\text{perímetro} = 4 * \text{lado1}$.
6. Mostrar el área del cuadrado.
7. Mostrar el perímetro del cuadrado.
8. Fin

2. Pseudocódigo

Inicio

Escribir "Ingresa la medida del lado del cuadrado:"

Leer lado1

$\text{área} \leftarrow \text{lado1} * \text{lado1}$

$\text{perímetro} \leftarrow 4 * \text{lado1}$

Escribir "El área del cuadrado es: ", area

Escribir "El perímetro del cuadrado es: ", perimetro

Fin

3. Sintaxis Básica

```
lado1=float(input("ingresa la  
medida del lado de cuadrado  
: "))  
area= lado1*lado1  
perimetro=4*lado1  
print(f"el area del cuadrado  
es: ",area)  
print(f"El perimetro del  
cuadrado es: ",perimetro)
```

Valores geométricos : Triángulo

1. Diagrama de flujo

1: pide al usuario ingresar un valor de altura
2: pide al usuario ingresar un valor de base
3 guarda el valor de base y altura
4 ejecuta la operación y la guarda en área
5 imprime el resultado

2. Pseudocódigo

```
inicio
    pide al usuario ingresar un valor
    se guarda en variable a de altura
    pide al usuario ingresar otro valor
    se guarda en la variable b base
    se crea una variable que ejecuta la operación
     $b*a/2$ 

    imprime el resultado de área y se muestra en
    la terminal
```

3. Sintaxis Básica

```
h=int(input("ingresa el valor
del área del triangulo: "))
b=int(input("ingresa el valor
de la base del triángulo: "))
área=b*h/2
print (f"El area del
triangulo es de : {area}")
```

Valores geométricos : Volumen Pirámide

1. Diagrama de flujo

1. Solicitar al usuario que ingrese el valor de la base de la pirámide (b).
2. Guardar el valor ingresado como un número decimal (float).
3. Solicitar al usuario que ingrese el valor de la altura de la pirámide (H).
4. Guardar el valor ingresado como un número decimal (float).
5. Calcular el área de la base (AB) usando la fórmula: $AB = b * b$.
6. Mostrar el área de la base (AB) al usuario.
7. Calcular el apotema lateral (a) usando la fórmula: $a = \sqrt{(b / 2)^2 + H^2}$.
8. Mostrar el apotema lateral (a) al usuario.
9. Calcular el área lateral (al) usando la fórmula: $al = 2 * b * a$.
10. Mostrar el área lateral (al) al usuario.
11. Calcular el volumen (v) usando la fórmula: $v = (1 / 3) * AB * H$.
12. Mostrar el volumen (v) al usuario.
13. Calcular el área total (at) usando la fórmula: $at = AB + al$.
14. Mostrar el área total (at) al usuario.
15. Fin del algoritmo.
- 16.

2. Pseudocódigo

Inicio

```
// Leer la base
Escribir "Ingresa el valor de la base de tu pirámide:"
Leer b
```

```
// Leer la altura
Escribir "Ingresa el valor de la altura de tu pirámide:"
Leer H
```

```
// Calcular área de la base
AB ← b * b
Escribir "El AB (Área de la base) es igual a: ", AB
```

```
// Calcular apotema lateral
a ← raíz cuadrada((b / 2) ^ 2 + H ^ 2)
Escribir "El apotema lateral es: ", a
```

```
// Calcular área lateral
al ← 2 * b * a
Escribir "El área lateral es: ", al
```

```
// Calcular volumen
v ← (1 / 3) * AB * H
Escribir "El volumen es: ", v
```

```
// Calcular área total
at ← AB + al
Escribir "El área total es: ", at
```

Fin

3. Sintaxis Básica

```
# Base
b = float(input("Ingresa el valor de la base de tu
pirámide: "))

# Altura
H = float(input("Ingresa el valor de la altura de tu
pirámide: "))

# Área de la base
AB = b * b
print(f"El AB (Área de la base) es igual a:{AB}")

# Apotema lateral
a = ((b / 2) ** 2 + H ** 2) ** 0.5
print(f"El apotema lateral es:{a}")

# Área lateral
al = 2 * b * a
print(f"El área lateral es:{al}")

# Volumen
v = (1 / 3) * AB * H
print(f"El volumen es:{v}")

# Área total
at = AB + al
print(f"El área total es:{at}")
```

Valores geométricos : **VOLUMEN CUBO**

1. Diagrama de flujo

1: pide ingresar el valor de la longitud de un lado del cubo
2: ejecuta la operación L^3
3: Devuelve el resultado del valor del volumen

2. Pseudocódigo

inicio
 pide ingresar un valor de longitud
 ejecuta la función elevando al cubo el valor ingresado
 devuelve el resultado de la operación (Volumen)

3. Sintaxis Básica

```
L=int(input(f"Ingresa la longitud de un lado del cubo: "))  
V=pow(L,3)  
print(f"El volumen del cubo es de: {V}")
```

Valores geométricos : **VOLUMEN Cilindro**

1. Diagrama de flujo

1 pide al usuario ingresar el valor de radio
2 pide al usuario ingresar el valor de altura
3 se declara una constante de pi
4 se hace la operación de radio al cubo
5 se hace la operación de $\pi \times \text{radio}^3 \times \text{altura}$

2. Pseudocódigo

inicio
se declara una constante de pi
se pide al usuario ingresar un valor
se pide al usuario ingresar otro valor
se calcula el cubo el radio
se calcula el volumen con la fórmula $\pi \times r^3 \times h$
muestra el resultado
fin

3. Sintaxis Básica

```
pi=3.14  
r=int(input("Ingresa el valor  
de radio: "))  
h=int(input(f"Ingresa el  
valor de la altura de  
cilindro: "))  
radio3=pow(r,2)  
V=pi*radio3*h  
print(f"El volumen del  
cilindro es de: {V}")
```

Programas Conversión



Conversiones

Decimal a binario

1. Diagrama de flujo

```
graph TD
    Inicio([Inicio]) --> Leer[Leer número decimal]
    Leer --> Cond1{¿El número es 0?}
    Cond1 -- Sí --> Imprimir0[Imprimir "0"]
    Imprimir0 --> Fin([Fin])
    Cond1 -- No --> Mientras[Mientras numero > 0]
    Mientras --> Residuo[Obtener residuo de número / 2]
    Residuo --> Agregar[Agregar residuo al inicio de la cadena binaria]
    Agregar --> Dividir[Dividir número entre 2 (entero)]
    Dividir --> Repetir[Repetir ciclo]
    Repetir --> ImprimirCadena[Imprimir cadena binaria]
    ImprimirCadena --> Fin
```

2. Pseudocódigo

```
Inicio
    Escribir "Introduce un número decimal:"

    Leer numero

    Si numero = 0 Entonces
        Escribir "El número binario es: 0"
    Sino
        binario <- cadena vacía

        Mientras numero > 0 Hacer
            residuo <- numero MOD 2

            binario <- convertir residuo a cadena + binario

            numero <- numero DIV 2

        Fin Mientras

        Escribir "El número binario es: " + binario
    FinSi
```

3. Sintaxis Básica

```
# Pide al usuario que ingrese un número entero
numero = int(input("Introduce un número entero: "))

# Guarda el número original para mostrarlo al final
numero_ingresado = numero

# Si el número es cero, su binario es simplemente '0'
if numero == 0:
    binario = "0"
else:
    binario = ""
    # Convertir el número a binario usando divisiones sucesivas por 2
    while numero > 0:
        residuo = numero % 2
        binario = str(residuo) + binario
        numero = numero // 2

# Mostrar el número original y su binario
print(f"El número {numero_ingresado} en binario es {binario}")
```


Conversiones

Binario a decimal

1. Diagrama de flujo

inicio

Solicitar al usuario que ingrese un número en binario (como cadena de texto).

Leer el número binario (guardar como cadena).

Inicializar una variable decimal en 0 y una variable potencia en 0.

Recorrer la cadena binaria de derecha a izquierda:

a. Por cada dígito, convertirlo a entero y multiplicarlo por 2 elevado a la posición (potencia).

b. Sumar el resultado a la variable decimal.

c. Incrementar la variable potencia en 1.

Mostrar el resultado en decimal.

Fin

2. Pseudocódigo

INICIO

Escribir "Ingresa un número en binario:"

Leer binario // como cadena

decimal \leftarrow 0

potencia \leftarrow 0

Para i desde longitud(binario) - 1 hasta 0
con paso -1 hacer

 dígito \leftarrow entero(binario[i])

 decimal \leftarrow decimal + (dígito * (2 ^

potencia))

 potencia \leftarrow potencia + 1

Fin Para

Escribir "El número decimal es: ", decimal

Fin

3. Sintaxis Básica

```
binario = input("Ingresa un  
número en binario: ")
```

```
decimal = 0
```

```
potencia = 0
```

```
# Recorrer la cadena de derecha  
a izquierda
```

```
for digito in binario[::-1]:
```

```
    decimal += int(digito) * (2
```

```
** potencia)
```

```
    potencia += 1
```

```
print("El número decimal es:",  
decimal)
```

Conversiones

Decimal a hexadecimal

1. Diagrama de flujo

Inicio
Solicitar al usuario que ingrese un número decimal.
Leer el número decimal y guardarlo en una variable (decimal).
Inicializar una variable hexadecimal como cadena vacía.
Si el número decimal es 0, asignar "0" a hexadecimal.
Si no es 0, repetir mientras decimal sea mayor a 0:
a. Obtener el residuo de dividir decimal entre 16.
b. Convertir el residuo a su equivalente hexadecimal (0-9, A-F).
c. Agregar el carácter correspondiente al principio de la cadena hexadecimal.
d. Dividir decimal entre 16 (división entera) y actualizar su valor.
Mostrar el resultado en hexadecimal.
Fin

2. Pseudocódigo

```
Inicio

Escribir "Ingresa un número decimal:"
Leer decimal

hexadecimal ← ""

Sí decimal = 0 Entonces
    hexadecimal ← "0"
Sino
    Mientras decimal > 0 Hacer
        residuo ← decimal MOD 16
        Si residuo < 10 Entonces
            carácter ← residuo como texto
        Sino
            carácter ← carácter ASCII correspondiente
        (A-F) FinSi
        hexadecimal ← carácter + hexadecimal
        decimal ← decimal DIV 16
    Fin Mientras
    FinSi

Escribir "El número en hexadecimal es:", hexadecimal

Fin
```

3. Sintaxis Básica

```
decimal = int(input("Ingresa un número decimal: "))

if decimal == 0:
    hexadecimal = "0"
else:
    hexadecimal = ""
    num = decimal
    while num > 0:
        residuo = num % 16
        if residuo < 10:
            carácter = str(residuo)
        else:
            carácter = chr(ord('A') + residuo - 10)
        hexadecimal = carácter + hexadecimal
        num //= 16

print("El número en hexadecimal es:", hexadecimal)
```

Conversiones

Hexadecimal a decimal

1. Diagrama de flujo

Inicio

Solicitar al usuario que ingrese un número en hexadecimal (como cadena de texto).

Leer el número hexadecimal (guardar como cadena).

Inicializar una variable decimal en 0 y una variable potencia en 0.

Recorrer la cadena hexadecimal de derecha a izquierda:

a. Obtener el valor numérico del dígito hexadecimal (0-9 → 0-9, A-F/a-f → 10-15).

b. Multiplicar el valor por 16 elevado a la posición (potencia).

c. Sumar el resultado a la variable decimal.

d. Incrementar la variable potencia en 1.

Mostrar el resultado en decimal.

Fin

2. Pseudocódigo

Inicio

Escribir "Ingresa un número en hexadecimal:"

Leer hexadecimal // como cadena

decimal ← 0

potencia ← 0

Para i desde longitud(hexadecimal) - 1 hasta 0 con paso -1 hacer

carácter ← hexadecimal[i]

Si carácter es un número (0-9) Entonces

valor ← entero(carácter)

Sino

valor ← valor correspondiente de la letra

(A/a=10, B/b=11, ..., F/f=15)

FinSi

decimal ← decimal + (valor * (16 ^ potencia))

potencia ← potencia + 1

Fin Para

Escribir "El número decimal es: ", decimal

Fin

3. Sintaxis Básica

```
hexadecimal = input("Ingresa un número en hexadecimal: ")
```

```
decimal = 0
```

```
potencia = 0
```

```
# Recorrer la cadena de derecha a izquierda
```

```
for caracter in hexadecimal[::-1]:
```

```
    if caracter.isdigit():
```

```
        valor = int(caracter)
```

```
    else:
```

```
        valor = ord(caracter.upper()) -
```

```
ord('A') + 10
```

```
        decimal += valor * (16 ** potencia)
```

```
        potencia += 1
```

```
print("El número decimal es:", decimal)
```

Conversiones

Decimal a octal

1. Diagrama de flujo

inicio

Solicitar al usuario que ingrese un número decimal.

Leer el número decimal y guardarlo en una variable (decimal).

Inicializar una variable octal como cadena vacía.

Si el número decimal es 0, asignar "0" a octal.

Si no es 0, repetir mientras decimal sea mayor a 0:

a. Obtener el residuo de dividir decimal entre 8.

b. Convertir el residuo a texto y agregarlo al principio de la cadena octal.

c. Dividir decimal entre 8 (división entera) y actualizar su valor.

Mostrar el resultado en octal.

Fin

2. Pseudocódigo

Inicio

Escribir "Ingresa un número decimal:"

Leer decimal

octal ← ""

Sí decimal = 0 Entonces

 octal ← "0"

Sino

 Mientras decimal > 0 Hacer

 residuo ← decimal MOD 8

 carácter ← residuo como texto

 octal ← carácter + octal

 decimal ← decimal DIV 8

 Fin Mientras

FinSi

Escribir "El número en octal es: ", octal

Fin

3. Sintaxis Básica

```
decimal = int(input("Ingresa un  
número decimal: "))
```

```
if decimal == 0:
```

```
    octal = "0"
```

```
else:
```

```
    octal = ""
```

```
    num = decimal
```

```
    while num > 0:
```

```
        residuo = num % 8
```

```
        octal = str(residuo) +
```

```
octal
```

```
        num //= 8
```

```
print("El número en octal es:",  
octal)
```

Conversiones

Octal a decimal

1. Diagrama de flujo

Inicio

Solicitar al usuario que ingrese un número en octal (como cadena de texto).

Leer el número octal (guardar como cadena).

Inicializar una variable decimal en 0 y una variable potencia en 0.

Recorrer la cadena octal de derecha a izquierda:

a. Convertir el dígito a entero.

b. Multiplicar el dígito por 8 elevado a la posición (potencia).

c. Sumar el resultado a la variable decimal.

d. Incrementar la variable potencia en 1.

Mostrar el resultado en decimal.

Fin

2. Pseudocódigo

Inicio

Escribir "Ingresa un número en octal:"

Leer octal // como cadena

decimal \leftarrow 0

potencia \leftarrow 0

Para i desde longitud(octal) - 1 hasta 0 con paso -1
hacer

 dígito \leftarrow entero(octal[i])

 decimal \leftarrow decimal + (dígito * (8 ^ potencia))

 potencia \leftarrow potencia + 1

Fin Para

Escribir "El número decimal es: ", decimal

Fin

3. Sintaxis Básica

```
octal = input("Ingresa un número  
en octal: ")
```

```
decimal = 0
```

```
potencia = 0
```

```
# Recorrer la cadena de derecha  
a izquierda
```

```
for dígito in octal[::-1]:
```

```
    decimal += int(dígito) * (8  
** potencia)
```

```
    potencia += 1
```

```
print("El número decimal es:",  
decimal)
```