

UADE

Introducción a la Algoritmia

Clase 3

Temas de la clase 2

- Estructuras de control: Estructura alternativa.
- Operadores relacionales.
- Operadores lógicos.

Objetivos de la Clase

Comprender las estructuras de control, específicamente las estructuras alternativas.

Familiarizarse con los operadores relacionales y lógicos.

Aplicar estructuras de control y operadores en la resolución de problemas.

Estructura Alternativa

Definición:

Permite tomar decisiones en el flujo del programa.

Se ejecuta una u otra instrucción dependiendo de una condición.

Pseudocódigo y diagrama de flujo de datos

Inicio

Ingrese su edad

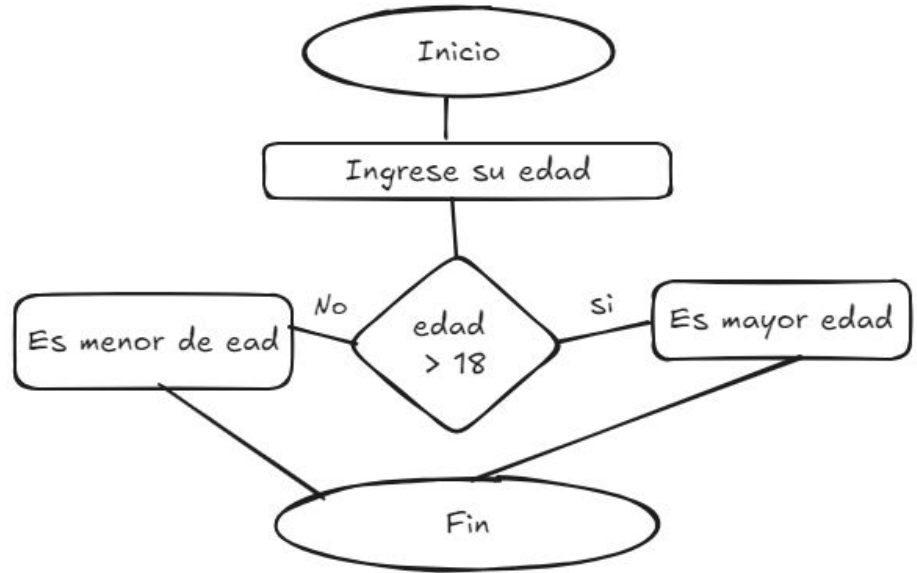
Si edad > 18 entonces

 imprimir “Es mayor de edad”

sino

 imprimir “Es menor de edad”

Fin



Ejemplo en python

```
edad = int(input("ingrese su edad"))
```

```
if edad >= 18:
```

```
    print("Es mayor de edad")
```

```
else:
```

```
    print("Es menor de edad")
```

A tener en cuenta

Sangría de cuatro espacios

Al finalizar la condición se escriben dos puntos:

Se ejecutan todas las instrucciones que se encuentran en el mismo nivel.

Operadores Relacionales

Definición:

Comparan dos valores y devuelven un valor booleano (True o False).

Operadores Comunes:

Igual a (==)

Distinto de (!=)

Mayor que (>)

Menor que (<)

Mayor o igual que (>=)

Menor o igual que (<=)

Ejemplo 1

Realizar un programa que pida un importe de venta. Si el importe de venta es mayor o igual a \$30000, imprimir el importe menos el 5%, sino imprimir el importe menos el 2%.

Nota: se puede usar una variable auxiliar.

Se pide realizar pseudocódigo, diagrama de flujo de datos e implementación en python.

Implementación en python

```
importe = int(input("Ingrese precio de venta"))
```

```
if importe >= 30000:
```

```
    descuento = 5
```

```
else:
```

```
    descuento = 2
```

```
importeConDescuento = importe - ( importe * descuento / 100 )
```

```
print(importeConDescuento)
```

Operadores Lógicos

Definición:

Combinan valores booleanos y devuelven un valor booleano.

Operadores Comunes:

and: Verdadero si ambos operandos son verdaderos.

or: Verdadero si al menos uno de los operandos es verdadero.

not: Invierte el valor del operando.

Tabla de verdad, operador and (operador y)

and: Verdadero si ambos operandos son verdaderos.

Condición 1	Condición 2	Condición 1 and Condición 2
V	V	Verdadero
V	F	Falso
F	V	Falso
F	F	Falso

Ejemplo 2

Realizar un programa que pida un importe de venta y el scoring de un cliente.

Si el importe de venta es mayor o igual a \$30000 y el cliente tiene un scoring mayor a 20, imprimir el importe menos el 5%, sino imprimir el importe menos el 2%.

Nota: se puede usar una variable auxiliar.

Se pide realizar pseudocódigo, diagrama de flujo de datos e implementación en python.

Implementación en python

```
importe = int(input("Ingrese precio de venta"))
scoring = int(input("Ingrese el scoring del cliente"))
if importe >= 30000 and scoring > 20:
    descuento = 5
else:
    descuento = 2
importeConDescuento = importe - ( importe * descuento / 100 )
print(importeConDescuento)
```

Tabla de verdad, operador or (operador o)

or: Verdadero si al menos uno de los operandos es verdadero.

Condición 1	Condición 2	Condición 1 or Condición 2
V	V	Verdadero
V	F	Verdadero
F	V	Verdadero
F	F	Falso

Ejemplo 3

Realizar un programa que pida un importe de venta y el scoring de un cliente.

Si el importe de venta es mayor o igual a \$30000 y el cliente tiene un scoring mayor a 20, imprimir el importe menos el 5%, si el importe de venta es mayor a \$10000 o el scoring es mayor a 5 imprimir el importe menos el 2%, en cualquier otro caso, no hay descuento.

Nota: se puede usar una variable auxiliar.

Se pide realizar pseudocódigo, diagrama de flujo de datos e implementación en python.

Implementación en python

```
importe = int(input("Ingrese precio de venta"))
scoring = int(input("Ingrese el scoring del cliente"))

if importe >= 30000 and scoring > 20:
    descuento = 5
elif importe >= 10000 or scoring > 5:
    descuento = 2
else:
    descuento = 0

importeConDescuento = importe - ( importe * descuento / 100 )

print(importeConDescuento)
```

Ejercicios

1. Escribe un programa que pida un número al usuario y luego imprima si ese número es mayor que 10.
2. Escribe un programa que pida un número al usuario y luego imprima si ese número es positivo o negativo. Si el número es 0, imprime que es neutro.
3. Escribe un programa que pida una calificación al usuario (entre 0 y 100) y luego imprima la nota en letras: "A" para 90 o más, "B" para 80-89, "C" para 70-79, "D" para 60-69, y "F" para menos de 60.
4. Escribe un programa que pida la edad y el peso de una persona, y luego imprima si es apta para donar sangre. Para ser apto, la persona debe tener entre 18 y 65 años y pesar al menos 50 kg.

Ejercicios

5. Escribe un programa que pida la temperatura exterior y la velocidad del viento, y luego imprima si es peligroso salir. Es peligroso si la temperatura es menor a 0°C o la velocidad del viento es mayor a 60 km/h.

6. Escribe un programa que pida un número al usuario y luego imprima si ese número no es divisible por 5.

7. Escribe un programa que pida tres números al usuario y luego imprima si al menos uno de ellos es mayor a 10 y, al mismo tiempo, todos son positivos.

Ejercicios

8. Escribe un programa que pida el puntaje de un examen (entre 0 y 100) y luego imprima una recomendación: "Excelente" para puntajes de 90 o más, "Muy bien" para 70-89, "Bien" para 50-69, y "Necesita mejorar" si es menor de 50, siempre que no haya reprobado con una calificación menor a 40.

9. Escribe un programa que pida la cantidad de horas trabajadas y la paga por hora. Si la cantidad de horas trabajadas es más de 40 y la paga por hora es mayor o igual a \$20, imprime "Recibe pago extra". Si la paga por hora es menor a \$10 y no trabajó más de 40 horas, imprime "Pago insuficiente". En cualquier otro caso, imprime "Pago adecuado".