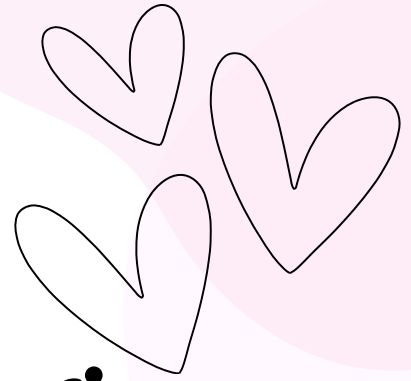




Ingeniería en
**Sistemas
Computacionales**



Instituto tecnológico superior de Martínez de la torre

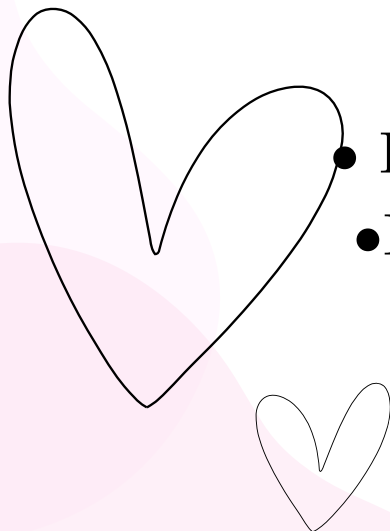
Fundamentos de programación

José lameiras pleites

Tema 3: Control de flujo

Actividades de aprendizaje

- cuadro comparativo
- Ejemplificar un programa
- Desarrollar un programa



SeleneHerrera Hernandez
Angel Gabriel Gonzáles Vázquez
Alondra yoselin lino perez



Cuadro comparativo

- Investigar sobre la sintaxis, funcionamiento y aplicación de las estructuras de selección y de repetición realizar un cuadro comparativo sobre las estructuras de repetición.

Estructura de repetición	Descripcion	sintaxis en puthon	aplicaciones comunes
Ciclo "for"	Itera sobre una secuencia de elementos (listas, cadenas, rangos, etc.). Sabe de antemano cuántas veces va a repetirse.	for variable in secuencia: # Código	Procesar listas, aplicar operaciones en serie, recorrer rangos específicos.
Ciclo "while"	Ejecuta un bloque de código mientras una condición booleana sea verdadera. No se sabe de antemano cuántas veces se repetirá.	while condición: # Código	Repetir hasta que se cumpla una condición, juegos interactivos, entradas de usuarios.
Ciclo d-while (simulado en Python)	Se asegura de ejecutar el bloque de código al menos una vez antes de evaluar la condición (en Python, esto se logra con un while True y un break	while True: # Código if condición: break	Ejecución garantizada al menos una vez, como en menús o validaciones.

Ejemplificar un programa

- Ejemplificar en programas que resuelvan problemas reales, discutirlo en grupo.

Ejemplos de programas que resuelvan problemas reales

1. Aplicación con Ciclo for: Calculadora de promedio de calificaciones

Este programa solicita un número de calificaciones, luego las procesa y calcula el promedio:

```
python Copiar código

# Declaración de variables
precio_total = 0
productos_comprados = int(input("¿Cuántos productos vas a comprar? "))

# Ciclo para procesar cada producto
for i in range(productos_comprados):
    precio_producto = float(input(f"\nIntroduce el precio del producto {i + 1}: $"))
    descuento_producto = float(input("Introduce el porcentaje de descuento: "))

    # Calcular el descuento del producto
    descuento_aplicado = (descuento_producto / 100) * precio_producto
    precio_final_producto = precio_producto - descuento_aplicado
    precio_total += precio_final_producto

    # Mostrar el precio final de cada producto
    print(f"El precio final del producto {i + 1} después del descuento es: ${precio_final_producto}")

# Aplicar descuento adicional si hay más de 4 productos
if productos_comprados > 4:
    descuento_adicional = 15
    descuento_extra = (descuento_adicional / 100) * precio_total
    precio_total -= descuento_extra # Restar el descuento adicional al total
    print(f"\n¡Has comprado más de 4 productos! Se te aplica un descuento adicional del 15")
    print(f"Descuento adicional: ${descuento_extra:.2f}")

# Mostrar el total a pagar
print(f"\nEl total a pagar por los {productos_comprados} productos es: ${precio_total:.2f}")
```

Aplicación con Ciclo while: Cajero automático

Este programa simula un cajero automático que sigue solicitando acciones hasta que el usuario decida salir

```
python Copiar código

saldo = 1000 # Saldo inicial

while True:
    print("\nOpciones: ")
    print("1. Ver saldo")
    print("2. Retirar dinero")
    print("3. Salir")

    opcion = int(input("Elige una opción: "))

    if opcion == 1:
        print(f"Tu saldo es: ${saldo:.2f}")
    elif opcion == 2:
        retiro = float(input("¿Cuánto dinero deseas retirar? "))
        if retiro <= saldo:
            saldo -= retiro
            print(f"Has retirado ${retiro:.2f}. Tu saldo restante es ${saldo:.2f}")
        else:
            print("Saldo insuficiente.")
    elif opcion == 3:
        print("Gracias por usar el cajero.")
        break
    else:
        print("Opción inválida, intenta de nuevo.")
```

Práctica de ejercicios con estructuras secuencial, selectiva e iterativa

1. Estructura Secuencial: Las instrucciones se ejecutan una después de la otra de manera lineal, sin
 - Ejemplo: Declaración de variables y cálculos básicos.

*


```
python Copiar código

a = 10
b = 5
resultado = a + b
print(f"El resultado de sumar {a} y {b} es: {resultado}")
```

2. Estructura Selectiva (if-else): Las instrucciones se ejecutan dependiendo de una condición.

- ejemplo: comparar dos números.

python

 Copiar código


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

if numero % 2 == 0:
    print(f"El número {numero} es par.")
else:
    print(f"El número {numero} es impar.")
```

3. Estructura Iterativa (for): Las instrucciones se repiten un número determinado de veces.

- Ejemplo. Contar los números del 1 al 10

python

 Copiar código

```
for i in range(1, 11):
    print(i)
```



Desarrollar un programa

- Realizar práctica de ejercicios, donde se utilicen las estructuras secuencial, selectiva e iterativa desarrollando algoritmo y programa, presentarlo y discutirlo en grupo

Problema. Crear un problema que permita calcular el total de la compra de varios productos, aplicar descuentos y mostrar el resultado final

Algoritmo:

1. **Pedir al usuario cuántos productos va a comprar.**
2. **Para cada producto, pedir el precio y porcentaje de descuento.**
3. **Calcular el precio final con descuento**
- 4 **Sumar los precios con descuento**
5. **Si el numero de productos subiera 4, aplicar un descuento adicional del 15%**
- 6.-**mostrar el total a pagar.**

```
python Copiar código

saldo = 1000 # Saldo inicial

while True:
    print("\nOpciones: ")
    print("1. Ver saldo")
    print("2. Retirar dinero")
    print("3. Salir")

    opcion = int(input("Elige una opción: "))

    if opcion == 1:
        print(f"Tu saldo es: ${saldo:.2f}")
    elif opcion == 2:
        retiro = float(input("¿Cuánto dinero deseas retirar? "))
        if retiro <= saldo:
            saldo -= retiro
            print(f"Has retirado ${retiro:.2f}. Tu saldo restante es ${saldo:.2f}")
        else:
            print("Saldo insuficiente.")
    elif opcion == 3:
        print("Gracias por usar el cajero.")
        break
    else:
        print("Opción inválida, intenta de nuevo.")
```

Discusión

Este programa combina las tres estructuras básicas de control en Python para resolver un problema práctico: la compra de productos y la aplicación de descuentos. Cada una de estas estructuras es fundamental para resolver problemas de diferentes tipos en programación y asegura un flujo de ejecución lógico y controlado.