

# Fundamentos de Programación Moderna

## Variables, control de flujo y lógica

Probabilidad y Estadística

Aprendiendo con diversión 

25 de noviembre de 2025

# ¿Qué vamos a aprender?

## Nota importante

¡Hoy empezamos a programar de verdad! 

Objetivo 1: Entender qué son las variables y tipos de datos

Objetivo 2: Escribir expresiones y sentencias

Objetivo 3: Controlar el flujo del programa con decisiones

Objetivo 4: Usar ciclos para repetir tareas

## Ejemplo

Como organizar cajas con diferentes objetos y decidir qué hacer con cada una

# Variables: Las Cajas de Memoria

## 💡 Nota importante

Una variable es como una caja etiquetada donde guardamos datos 📁

Cada variable tiene un nombre (etiqueta)

Cada variable almacena un valor

## 📝 Ejemplo

```
nombre_empresa = "Mi Empresa S.A."  
saldo = 1500000  
activo = True
```

✓ La computadora reserva memoria para guardar estos valores

# Tipos de Datos Primitivos

Tipo	Uso	Ejemplo
Enteros (int) 	IDs, cantidades	<code>id_usuario = 12345</code>
Flotantes (float) 	Dinero, precios	<code>precio = 19.99</code>
Textos (string) 	Nombres, emails	<code>nombre = "Juan"</code>
Booleanos (bool) 	Estados sí/no	<code>activo = True</code>

## 💡 Nota importante

Cada tipo de dato se guarda diferente en memoria

# Operadores Aritméticos

Permiten hacer cálculos matemáticos

Operador	Operación	Ejemplo
+	Suma	$10 + 5 = 15$
-	Resta	$20 - 8 = 12$
*	Multiplicación	$6 * 7 = 42$
/	División	$15 / 3 = 5$

## Ejemplo

`total = precio * cantidad + iva`

# Operadores Lógicos

Permiten comparar y combinar condiciones

Operador	Significado	Ejemplo
<code>==</code>	Igual a	<code>edad == 18</code>
<code>!=</code>	Diferente de	<code>saldo != 0</code>
<code>&gt;</code>	Mayor que	<code>precio &gt; 100</code>
<code>&gt;=</code>	Mayor o igual	<code>edad &gt;= 18</code>
<code>&lt;</code>	Menor que	<code>stock &lt; 10</code>
<code>&lt;=</code>	Menor o igual	<code>nota &lt;= 7</code>

## Ejemplo

Si saldo `>=` precio entonces aprobar compra

# Expresiones vs. Sentencias

## 💡 Nota importante

¿Cuál es la diferencia? 🤔

Expresión	Sentencia
Tiene un valor	Ejecuta una acción
$5 + 3$	$x = 5 + 3$
precio * 1.19	print(mensaje)
edad $i = 18$	if edad $i = 18$ :

## 📝 Ejemplo

Expresión: precio \* 1.19 (calcula IVA) → Sentencia: total = precio \* 1.19 (asigna resultado)

## Ejemplo Práctico: Cálculo de IVA

### ➊ Problema

Calcular el precio total de un producto con IVA (19%

$$\text{precio}_n\text{eto} = 100000$$

$$\text{iva} = \text{precio}_n\text{eto} * 0,19 = 19000$$

$$\text{precio}_t\text{otal} = \text{precio}_n\text{eto} + \text{iva} = 119000$$

### Ejemplo

# Control de Flujo: IF/ELSE

## 💡 Nota importante

¡La computadora puede tomar decisiones! 🖥️

IF = Si se cumple una condición, ejecuta un bloque de código

ELSE = Si no, ejecuta otro bloque

## 📝 Ejemplo

```
if saldo >= costo:  
    aprobar_transaccion()  
    restar_saldo(costo)  
else:  
    rechazar_transaccion()  
    mostrar_error("Saldo insuficiente")
```

# Ejemplo: Validación de Transacción

## ① Problema

Aprobar o rechazar una compra según el saldo disponible

## 💡 Ejemplo

```
saldo = 50000
precio_producto = 30000

if saldo >= precio_producto:
    print("Compra aprobada")
    saldo = saldo - precio_producto
    print("Nuevo saldo:", saldo)
else:
    print("Saldo insuficiente")
```

Resultado : Compra aprobada Nuevo saldo : 20000

# Operadores Lógicos: AND, OR, NOT

Operador	Uso	Ejemplo
AND	Ambas condiciones verdaderas	edad $\geq 18$ AND tiene_icencia
OR	Al menos una verdadera	pago_efectivo OR pago_tarjeta
NOT	Niega la condición	NOT usuario_bloqueado

## Ejemplo

```
if saldo >= precio AND usuario_activo:  
    procesar_compra()
```

## Nota importante

Las validaciones de seguridad usan muchos AND para verificar múltiples condiciones

# Estructuras SWITCH/MATCH

Para elegir entre múltiples opciones

## Ejemplo

Menú de opciones en un ERP:

```
match opcion: case 1: crearfactura() case2 : consultar saldo() case3 :  
generar_reporte() case print(" Opcion invalida" )
```

## Nota importante

Más limpio que múltiples IF/ELIF cuando hay muchas opciones

# Ciclos: WHILE (Mientras)

## 💡 Nota importante

¡Repetir código mientras se cumpla una condición! ↗

Se usa cuando no sabemos cuántas veces repetir

## 💡 Ejemplo

```
intentos = 0
conectado = False

while not conectado and intentos < 5:
    conectado = intentarConexion()
    intentos = intentos + 1

if conectado:
    print("Conexion exitosa")
else:
    print("Error de conexion")
```

# Ciclos: FOR (Para Cada)

Se usa cuando sabemos cuántas veces repetir

Ideal para recorrer listas de datos

## Ejemplo

Pagar nómina a todos los empleados:

```
for empleado in empleados: procesarpago(empleado) enviarcomprobante(empleado) print(" Pag
```

# Ejemplo Práctico: Calcular Total de Ventas

## ➊ Problema

Sumar todas las ventas del día

## ➋ Ejemplo

```
ventas = [15000, 23000, 8500, 12000, 31000]
total = 0

for venta in ventas:
    total = total + venta

print("Total del da:", total)
```

*Resultado : Total del día : 89500*

# Prevención de Bucles Infinitos

 Nota importante

 ¡Cuidado con los bucles que nunca terminan!

 Ejemplo

Bucle infinito (ERROR):

 Ejemplo

Bucle correcto:

# ¡Tu Turno de Practicar!

## Problema

Crear un programa que valide un descuento especial

Condiciones:

- Si la compra es mayor a 100,000 → 10
- Si además es cliente VIP → 15
- Calcular el precio final

## Nota importante

Usa IF/ELSE y operadores AND para resolverlo

# ¡Resumen de lo Aprendido!

- ✓ Variables guardan datos en memoria (como cajas etiquetadas)
- ✓ Tipos de datos: Enteros, Flotantes, Textos, Booleanos
- ✓ Operadores aritméticos para cálculos (+, -, \*, /,
- ✓ Operadores lógicos para comparaciones (==, !=, *i*, *i*, *i*=, *i*=)
- ✓ IF/ELSE permite tomar decisiones en el código
- ✓ WHILE repite mientras se cumpla una condición
- ✓ FOR recorre listas y repite un número fijo de veces

## Nota importante

¡Con estos fundamentos ya puedes crear programas reales! 

¡Sigue aprendiendo! 