

# Laboratorio 2 - Condicionales

#### Objetivos

#### **Parte 1: Operadores**

Actividad 1: Comprendiendo los Operadores de Comparación

Actividad 2: Comprendiendo los Operadores Lógicos

Actividad 3: Explorando Combinaciones de operadores

Parte 2: Condicionales

Condicional if

Condicional if-else

Condicional if-elif-else

Actividad 1: Experimentando con Condicionales

Actividad 2: Usando Condicionales

Actividad 3: Condicionales Anidados

Parte 3: Desafíos

Desafío 1: League of Python

Desafío 2: Cachipún

Desafío 3: PyGic

# **Objetivos**

1. En el contexto de un problema, expresar una condición lógica que modele una situación

particular, a partir de comparaciones y operadores lógicos.

 Organizar la lógica de decisiones de un programa en Python en torno a decisiones

simples (un camino), dobles (if-else), en cascada (if-elif) o anidadas, según sea

necesario.

3. Resolver problemas en Python a través de algoritmos que deban tomar decisiones,

expresando adecuadamente las condiciones que permiten decidir el flujo de ejecución.

# **Parte 1: Operadores**

En este laboratorio, exploraremos las estructuras condicionales en Python. Estas permiten que un programa tome decisiones en función de ciertas condiciones.

Pero, antes de utilizar condicionales vamos a entender qué significa que una expresión sea verdadera o falsa a través de los operadores de comparación y los operadores lógicos.

# Actividad 1: Comprendiendo los Operadores de Comparación

Los operadores de comparación permiten comparar valores numéricos o strings entre sí. Devuelven un valor True (Verdadero) o False (Falso) dependiendo del resultado de la comparación.

En la siguiente tabla encontrarás los operadores de comparación que utilizaremos en Python:

Operador	Descripción	Ejemplo
==	Igual que	5 == 5 → True
!=	Diferente qie	4 != 2 → True
<	Menor que	3 < 8 → True
<=	Menor o igual que	6 <= 6 → True
>	Mayor que	10 > 5 → True
>=	Mayor o igual que	7 >= 8 → False



En programación, True y False representan los dos posibles valores de una expresión lógica o condicional. Son llamados valores booleanos y se usan para indicar si una condición es verdadera o falsa

#### Por ejemplo:

```
print(5 > 3) # Esto imprimirá True, porque 5 es mayor que 3 print(2 == 8) # Esto imprimirá False, porque 2 no es igual a 8
```

Aunque en este laboratorio no trabajaremos directamente con variables booleanas, es importante conocer que los operadores de comparación y lógicos generan resultados True o False, lo que nos permitirá tomar decisiones en los programas usando condicionales. Más adelante en el curso veremos cómo usar estos valores de forma más avanzada.

### 1. Analiza los siguientes casos ¿Qué valor tendrá cada expresión?

- a. 5 == 5
- **b.** 10 != 7
- C. 3 < 8
- d. 7 >= 7
- **e.** 12 > 15
- f. 4 <= 3

Utiliza la consola de Python para corroborar tus resultados.

# ▼ ? ¿No recuerdas cómo usar la consola?

Recuerda que la **consola** es la ventana principal que aparece al abrir IDLE. Permite escribir y ejecutar código Python línea por línea de manera interactiva, por lo que es útil para probar pequeñas porciones de código rápidamente.

Por ejemplo, si queremos ejecutar el primer caso, podemos escribirlo directamente en la consola y presionar *enter*:

```
File Edit Shell Debug Options Window Help

Python 3.13.1 (tags/v3.13.1:0671451, Dec 3 2024, 19:06:28) [MSC v.1942 64 bit ( AMD64)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

>>> 5 == 5

True

True

Ln:5 Col: 0
```

2. Completa la tabla evaluando las expresiones, indicando si el resultado es True (verdadero) o False (falso):



- a. ¿Qué ocurre si en la primera expresión 5 == 10 dejamos un solo signo igual (5 = 10)? Describe con tus palabras antes de probar en la consola.
- b. Ahora prueba ejecutar la expresión 5 = 10 en la consola. ¿Qué ocurrió?
   ¿Era lo que esperabas?
- c. ¿Cuándo utilizamos = y cuándo ==?



Cuando queremos asignar un valor a una variable utilizaremos un solo signo :

```
variable = 10
```

Si queremos comparar dos valores utilizaremos dos signos iguales ==:

```
"hola" == "chao"
```

- 3. ¿Qué ocurre si comparamos números de diferentes tipos? Evalúa estas expresiones y explica los resultados:
  - **a.** 5 == 5.0
  - **b.** 10 != 10.0
  - **C.** 3 < 3.5
- ¿Y si comparamos diferentes tipos de datos? Evalúa estas expresiones y explica los resultados::
  - a. 5 == "5"
  - **b.** 10 != "10"
  - **C.** True == 1

# Actividad 2: Comprendiendo los Operadores Lógicos

Cuando necesitamos comparar dos valores o expresiones booleanas entre sí podemos utilizar los Operadores Lógicos and ("y"), or ("o") y not ("no").

Operador	Descripción	Ejemplo	Explicación
and	Verdadero si ambas condiciones son verdaderas	$(5 > 3)$ and $(2 < 4) \rightarrow$ True	Se puede leer como "se comple la condición 1 <u>y</u> se cumple la condición 2".
or	Verdadero si al menos una	(5 > 3) or (2 > 4) → True	Se puede leer como "se comple la condición 1 <u>o</u> se

	condición es verdadera		cumple la <i>condición</i> 2".	
not	Invierte el valor de la condición	not (5 > 3) → False	Se puede leer como " <u>no</u> se cumple la condición".	

### 1. Analizando el operador and:

- a. ¿Qué valor tendrá la expresión True and True? Responde y explica por qué.
- b. Completa la tabla de verdad para el operador and:

Α	В	A and B
True	True	True
True	False	
False	True	
False	False	

c. Prueba en Python: utiliza la consola de Python para verificar los resultados de la tabla de verdad. Puedes usar print, por ejemplo:

print(True and True)

### 2. Analizando el operador or:

- a. ¿Qué valor tendrá la expresión False or True? Responde y explica por qué.
- b. Completa la tabla de verdad para el operador or:

Α	В	A or B
True	True	
True	False	
False	True	
False	False	

Prueba en la consola de Python y verifica si tus resultados son correctos.

### 3. Analizando el operador not

- a. ¿Qué valor tendrá la expresión not True ? Responde y explica por qué.
- b. Completa la tabla de verdad para el operador not:

А	not(A)
True	
False	

Usa la consola de Pyhton para verificar los resultados de la tabla de verdad.

#### 4. Casos "Extraños"

- a. ¿Qué ocurre si usamos números en lugar de valores booleanos? Evalúa estas expresiones:
  - 0 and 1
  - 1 or 0
  - onot 0
  - • Pista: En Python, el o se considera False, y cualquier número distinto de o se considera True.
- b. ¿Qué pasa si usamos *strings* en lugar de valores booleanos? Evalúa estas expresiones:
  - "Hola" and ""
  - "" or "Python"
  - not("Mundo")
- c. Verifica con la consola de Python si tus respuestas acertaron el resultado en los casos anteriores.

# **Actividad 3: Explorando Combinaciones de operadores**

Los operadores de comparación y los operadores lógicos se pueden **combinar** para **generar condiciones más complejas**. Esto podría ser necesario dependiendo del contexto del problema que tenemos que resolver.

1. Dado los siguientes códigos, ¿qué valores (True o False) devolverán las expresiones?

```
edad = 25
tiene_licencia = False
print(edad >= 18 and tiene_licencia)

temperatura = 30
print(temperatura < 0 or temperatura > 25)

a = 10
b = 5
print(not (a < b or b == 5))

x = 12
y = 7
print(x % 2 == 0 and y % 2 != 0)
```

# Parte 2: Condicionales

Los condicionales en Python nos permiten tomar **decisiones** en un programa. Con ellos, el código puede ejecutar diferentes instrucciones dependiendo de si ciertas **condiciones** se cumplen o no.

Los condicionales funcionan evaluando **expresiones lógicas**, que pueden ser **verdaderas** ( True ) o **falsas** ( False ). Dependiendo del resultado, se ejecuta una parte del código u otra.

En esta sección veremos los diferentes tipos de condicionales que utilizaremos en el curso, y ejercitaremos cómo se utilizan.

# Condicional if

Se usa cuando queremos ejecutar un código solo si una condición se cumple (es True). Si la condición es False, el código dentro del if no se ejecuta.

Ejemplo:

```
edad = int(input("Ingresa tu edad: "))
if edad >= 18:
  print("Eres mayor de edad") # Se ejecuta si edad es 18 o más
```

Podemos leer el condicional if del ejemplo como: <u>si</u> la variable edad es mayor o igual que 18, **entonces** se imprime el mensaje "Eres mayor de edad".



! Algo muy importante a tener en cuenta al momento de escribir condicionales es la **identación**: la forma correcta es que todo el código que queremos que se ejecute dentro del ir vaya corrido hacia la derecha **un tab (o 4 espacios)**.

```
if condicion1:
¡Identación! ← ← <sentencia 1>
```

Si es que no se respeta la identación, Python lanzará un error al momento de ejecutar el código:

```
numero = 1
if numero != 0:
print("No es cero")

SyntaxError

Expected an indented block after 'if' statement on line 2

Aceptar
```

# **Condicional** if-else

Se usa cuando hay dos caminos posibles:

- Uno cuando la condición es True, es decir, cuando se cumple la condición.
- Y otro cuando la condición es False, es decir, cuando no se cumple.

### Ejemplo:

```
temperatura = int(input("Ingresa la temperatura actual: "))
if temperatura > 20:
    print("Hace calor") # Se ejecuta si temperatura es mayor a 20
else:
    print("Hace frío") # Se ejecuta si temperatura es 20 o menor
```

Podemos leer el condicional if-else del ejemplo como: <u>si</u> la temperatura es mayor a 20, **entonces** se imprime "Hace calor", **si no** se imprime "Hace Frío".

# Condicional if-elif-else

Si tenemos **más de dos condiciones**, usamos elif (abreviatura de **else if**). En este caso, el programa evalúa las condiciones **en orden** y ejecuta la primera que sea True.

### Ejemplo:

```
nota = int(input("Ingresa tu nota: "))
if nota >= 90:
    print("Excelente")
elif nota >= 70:
    print("Bueno")
elif nota >= 55:
    print("Regular")
else:
    print("Malo")
```

En el caso del ejemplo, el condicional se ejecutará de la siguiente forma:

- 1. Si la variable nota es mayor o igual a 90, se imprimirá "Excelente", si no pasamos a la siguiente condición.
- 2. Si la nota es mayor o igual a 70 entonces se imprime "



Cuando tenemos un condicional del tipo if-elif-else Python evalúa las condiciones una por una de arriba hacia abajo, y se detiene en la primera que sea True, sin revisar las siguientes. Ese comportamiento se llama "cortocircuito" (short-circuiting) en la evaluación de condiciones.

Para entender esto analicemos el ejemplo de arriba:

- Si el usuario ingresa, por ejemplo, la nota 95, sí se cumple que es mayor o igual que 90, pero también se cumple que es mayor o igual a 70, y mayor o igual a 55.
- Sin embargo, al ejecutar el programa Python irá a la primera condición if nota >= 90, y como esta se cumple, no evaluará ninguna de las siguientes, imprimirá el mensaje y seguirá ejecutando el programa con lo que sigue después de todo el condicional.

# **Actividad 1: Experimentando con Condicionales**

1. Completa las siguientes condiciones rellenando el espacio en blanco (\_\_\_\_\_) según se requiera en cada expresión condicional:

```
edad = int(input("Ingresa la edad "))
if edad ____:
  print("Puedes votar")
numero= int(input("Ingresa un número "))
if numero ____:
  print("El número es positivo")
else:
  print("El número es negativo")
```

```
edad = int(input("Ingresa la edad "))
if edad ______ 18:
    print("Eres menor de edad")
elif edad ______ 18 and edad _____ 60:
    print("Eres un adulto")
else:
    print("Eres un adulto mayor")
```

2. ¿Qué imprimirá este código si a = 7 y b = 10?

```
if a > b:
    print("A es mayor que B")
elif a < b:
    print("A es menor que B")
else:
    print("A es igual a B")</pre>
```

3. Encuentra y corrige el error lógico en este código, para que los mensajes de los print se cumplan correctamente:

```
if x >= 10 and x < 20:
    print("Entre 10 y 20")
elif x < 10 or x >= 20:
    print("Fuera de rango")
else:
    print("Exactamente 10 o 20")
```

### **Actividad 2: Usando Condicionales**

1. Escribe un programa que pida dos números e imprima cuál es mayor.

#### ▼ • • Pista

Primero piensa en las condiciones que debería tener el programa, y luego usa un condicional para imrpimir los mensajes correspondientes.

Analiza también si es que existe algún caso borde (¿qué ocurre si el usuario ingresa dos números iguales?, ¿qué podría hacer el programa en ese caso?)

2. Escribe un programa que genere un número aleatorio entre 1 y 100. Si es par, debe imrpimir "Es par", y si es impar, "Es impar".

## ▼ 🔢 ¿No recuerdas cómo generar un número aleatorio?

La biblioteca random que posee la función randint(x, y) que genera un número al azar entero entre el rango x e y:

import random

aleatorio = random.randint(1,10) #genera un entero aleatorio entre el 1

## 3. Juego de adivinar un número

Escribe un programa que:

- Genere un número aleatorio entre 1 y 10 y permita que el usuario intente adivinarlo ingresando un número.
- Si el usuario adivina, debe imprimir "¡Correcto!", y si no, "Intenta de nuevo".

#### ▼ • • Pista

Sería útil guardar el número aleatorio en una variable, el número que ingrese el usuario en otra, y luego compararlas e imprimir los mensajes usando condicionales.

#### 4. Calculadora de descuentos

Escribe un programa que pida el precio de un producto en pesos, y luego calcule su descuento según las siguientes reglas:

- Si cuesta más de **\$100.000**, aplica 20% de descuento.
- Si cuesta entre \$50.000 y \$100.000, aplica 10%.
- Si cuesta menos de \$50.000, no hay descuento.

Al final, debe mostrar el monto total de descontado, y el precio final del producto. Ambos valores deben estar **redondeados**.

Guíate con estos ejemplos de ejecución del programa final:

Ingresa el precio del producto: 500000

Descuento aplicado: 100000

Precio final: 400000

Ingresa el precio del producto: 98990

Descuento aplicado: 9899

Precio final: 89091

Ingresa el precio del producto: 23590

Descuento aplicado: 0 Precio final: 23590

#### 5. Clasificador de Sismos

Luego de titularte comenzaste a trabajar en el Centro Sismológico Nacional, cuyo sitio web <u>sismologia.cl</u> ofrece informes sobre los últimos sismos.

Como es frecuente en Chile, hubo un temblor hace poco y tu jefe debe dar el reporte de su magnitud. Ya que tu jefe no entiende muy bien cómo se clasifican los sismos (nadie sabe cómo consiguió su puesto), te delega la gran tarea de implementar un clasificador de sismos según la escala de Richter:

Magnitud (M <sub>W</sub> =Mayores de 6,9 M <sub>L</sub> =De 2,0 a 6,9)	Descripción	
Menos de 2,0	Micro	
2,0-2,9	Menor	
3,0-3,9	Melloi	
4,0-4,9	Ligero	
5,0-5,9	Moderado	
6,0-6,9	Fuerte	
7,0-7,9	Mayor	
8,0-8,9	Énico o Cataclismo	
9,0-9,9	Épico o Cataclismo	
10,0+	Legendario o apocalíptico	

Figura 1: Escala de Richter

Escribe un programa que permita clasificar un sismo ingresado por tu jefe según la escala de Richter, para salvar tu puesto de trabajo. Guíate con los siguientes ejemplos de ejecución:

```
Ingrese el grado: 2.7
Menor

Ingrese el grado: 8.6
Epico

Ingrese el grado: -10
Grado invalido
```

### **Actividad 3: Condicionales Anidados**

Un **condicional anidado** es cuando dentro de un if, elif o else hay otro if. Se usan cuando la decisión depende de **más de un factor**.

### Ejemplo:

```
dia = "Lunes"
hora = 10

if dia == "Lunes" or dia == "Martes":
    if hora < 13:
        print("Es horario de clases")
    else:
        print("Es hora de almuerzo")
else:
    print("No es lunes ni martes")</pre>
```

En este ejemplo, si es que el día es "Lunes" o "Martes", se ingresará a un segundo condicional para ver si es horario de clases o almuerzo.

### ¿Por qué usar condicionales anidados?

- Permiten evaluar múltiples condiciones con mayor claridad.
- Ayudan cuando una decisión depende de otra.

#### Veamos cómo se aplica:

- 1. Escribe un programa que le permita al usuario ingresar su edad, y si tiene licencia de conducir, y vea si tiene o no permisos para manejar siguiendo estas reglas:
  - Si la edad es menor a 13, imprimir "Eres un niño".
  - Si está entre 13 y 17, imprimir "Eres un adolescente".
  - Si la edad es 18 o más:
    - Si tiene licencia, imprimir "Eres adulto y puedes conducir".
    - Si no tiene licencia, imprimir "Eres adulto, pero no puedes conducir".

# Parte 3: Desafíos

# **Desafío 1: League of Python**

Eres un ávido jugador del juego más popular llamado League of Python. Tu sabes que el verdadero nivel de un jugador se muestra en competiciones a gran escala. Y League of Python cuenta con un sistema de Divisiones digno de los mejores eSports.

El sistema de Divisiones funciona de la siguiente manera: para clasificar a sus jugadores el sistema revisa el puntaje obtenido en las dos últimas partidas y, en base a la suma de estos puntajes, los jugadores se clasifican como:

- Bronce: si la suma es menor o igual que 300 pts.
- Plata: si la suma es mayor que 300 y menor o igual que 500 pts.
- Oro: si la suma es mayor que 500 y menor o igual que 700 pts.
- Platino: si la suma es mayor que 700 y menor o igual que 1000 pts.
- Diamante: si la suma es mayor que 1000 pts.



Figura 2: Las ligas de LOP.

Como verdadero amante de LOP, decides escribir un programa para saber en qué liga te quedas esta temporada. Tu programa debe **recibir** dos números (los puntajes de tus 2 últimas partidas) y debe asignarte la División en la que quedarás según su suma.

Guíate con los siguientes ejemplos de ejecución:

Ingrese el puntaje de la primera partida: 100 Ingrese el punatje de la segunda partida: 150

Tu division es Bronce

Ingrese el puntaje de la primera partida: 900 Ingrese el punatje de la segunda partida: 800

Tu division es Diamante

# Desafío 2: Cachipún

Estás en pleno campo de batalla con tu mejor amigo porque no se pueden poner de acuerdo en quién es el mejor villano de películas. Mientras que tu amigo dice que es el despiadado y enigmático **Voldemort** de la Saga Harry Potter, tú sostienes que el poderoso **Thanos** es quien debe llevar ese título. Luego de una acalorada discusión, deciden que la mejor forma de resolver el conflicto y conservar su amistad, es someter la decisión a una ronda de cachipún que defina la opción ganadora.



Figura 3: Un ejemplo de su amistad.

Escribe un programa que simule la ronda de cachipún, en donde tú y tu amigo elijan una de las siguientes tres opciones: Piedra, Papel o Tijera. El programa debe almacenar ambas opciones y determinar quién es el ganador teniendo en cuenta las siguientes reglas:

- · Papel le gana a Piedra
- Piedra le gana a Tijera
- Tijera le gana a Papel

Primero, debes mostrar por pantalla un mensaje que le explique al usuario como jugar. Después, debes pedir dos números para saber que movimiento hizo cada jugador. En base a estos dos valores debes decidir quien gano (¡o si hubo un empate!). Guíate con los siguientes ejemplos:

### Bienvenid@ a Cachipun:

- Presiona 1 para jugar piedra.
- Presiona 2 para jugar papel.
- Presiona 3 para jugar tijera.

Jugador 1 ingresa tu opción: 1 Jugador 2 ingresa tu opción: 1

Ha sido un empate!

### Bienvenid@ a Cachipun:

- Presiona 1 para jugar piedra.

- Presiona 2 para jugar papel.
- Presiona 3 para jugar tijera.

Jugador 1 ingresa tu opción: 2 Jugador 2 ingresa tu opción: 1

El jugador 1 ha ganado. Voldemort es el mejor villano.

### ▼ • • Pista

Piensa primero en cuántas combinaciones posibles existen. Si es que son más de dos, lo más conveniente es usar elifs. También podría ser útil usar condicionales anidados.

# Desafío 3: PyGic

Manejas una tienda de cartas *PyGic: The Programering* y quieres desarrollar un programa que determine el precio de una carta que un cliente te quiere vender.



Una carta promedio de PyGic.

El programa debe recibir el nombre de la carta, su rareza (comun o rara), su ataque y su defensa. Para obtener el precio en dólares de las cartas se utiliza el siguiente método:

• Si la carta es común:

- Si la suma de su ataque y defensa es mayor a 15, entonces su precio es: 2 \* ataque + defensa
- Si no, su precio es: ataque + defensa / 2
- Si la carta es rara:
  - Si la multiplicación de su ataque con su defensa es mayor a 30, entonces su precio es: 2 \* (ataque + defensa) + 1
  - Si no, su precio es: ataque + 2 \* defensa + 1

Finalmente, si el precio de la carta es menor a 20 dólares, entonces el programa debe imprimir "Por ese precio me compro un mejor posavasos". Si el precio de la carta es mayor o igual a 20, debe imprimir "Su carta [ nombre ] vale [ precio ], y me estoy arriesgando".

Guíate por los siguientes ejemplos de ejecución:

Ingrese nombre de la carta: Volundr

Ingrese rareza: rara Ingrese ataque: 21 Ingrese defensa: 13

Su carta Volundr vale 69 y me estoy arriesgando.

Ingrese nombre de la carta: El Hacker

Ingrese rareza: comun

Ingrese ataque: 5 Ingrese defensa: 8

Por este precio mejor me compro un posavasos.

#### ▼ • • Pista

Usa condicionales anidados (if-else) para organizar la lógica del cálculo del precio: primero revisa la rareza, y luego evalúa el precio.

Una vez tengas calculado el precio, puedes usar otro condicional al final para impimir los mensajes según corresponda.