DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

CONTROL DE FLUJO: ESTRUCTURAS DE DECISIÓN

10145 - FUNDAMENTOS DE PROGRAMACIÓN PARA INGENIERÍA

RESUMEN DE CONTENIDOS

OPERADORES DE COMPARACIÓN

- Además de los operadores aritméticos, en Python existen los siguientes **operadores de comparación**

Mayor	`>`	`4.3 > 3.2`
Mayor o igual	`>=`	`4.0 >= 4.1`
Menor	`<`	`-2 < 0`
Menor o igual	`<=`	`-3.14 <= -3.2`
Igual	`==`	`2.0 == 2`
Distinto	`!=`	`-2 != -2.1`

OPERADORES LÓGICOS

- Existen además **operadores lógicos**:
  - La **negación** (`not`) para **invertir** el valor de verdad

```
not 50 > 4 → not True → False
```

  - La **conjunción** o "*y lógico*" (**and**), que resulta verdadero si y solo si **todas** las sub-expresiones son verdaderas

```
x > y and y <= z
```

  - La **disyunción** u "*o lógico*" (**or**), que resulta verdadero si **al menos una** de las sub-expresiones lo es

```
x != y or x <= z or y < z
```

PRECEDENCIA DE OPERADORES

Operación	Operador	Aridad	Asociatividad	Precedencia
Exponenciación	**	Binario	Por la derecha	1
Identidad	+	Unario	—	2
Cambio de signo	-	Unario	—	2
Multiplicación	*	Binario	Por la izquierda	3
División	/	Binario	Por la izquierda	3
Módulo (o resto)	%	Binario	Por la izquierda	3
Suma	+	Binario	Por la izquierda	4
Resta	-	Binario	Por la izquierda	4
Igual que	==	Binario	—	5
Distinto de	!=	Binario	—	5
Menor que	<	Binario	—	5
Menor o igual que	<=	Binario	—	5
Mayor que	>	Binario	—	5
Mayor o Igual que	>=	Binario	—	5
Negación	not	Unario	—	6
Conjunción	and	Binario	Por la izquierda	7
Disyunción	or	Binario	Por la izquierda	8

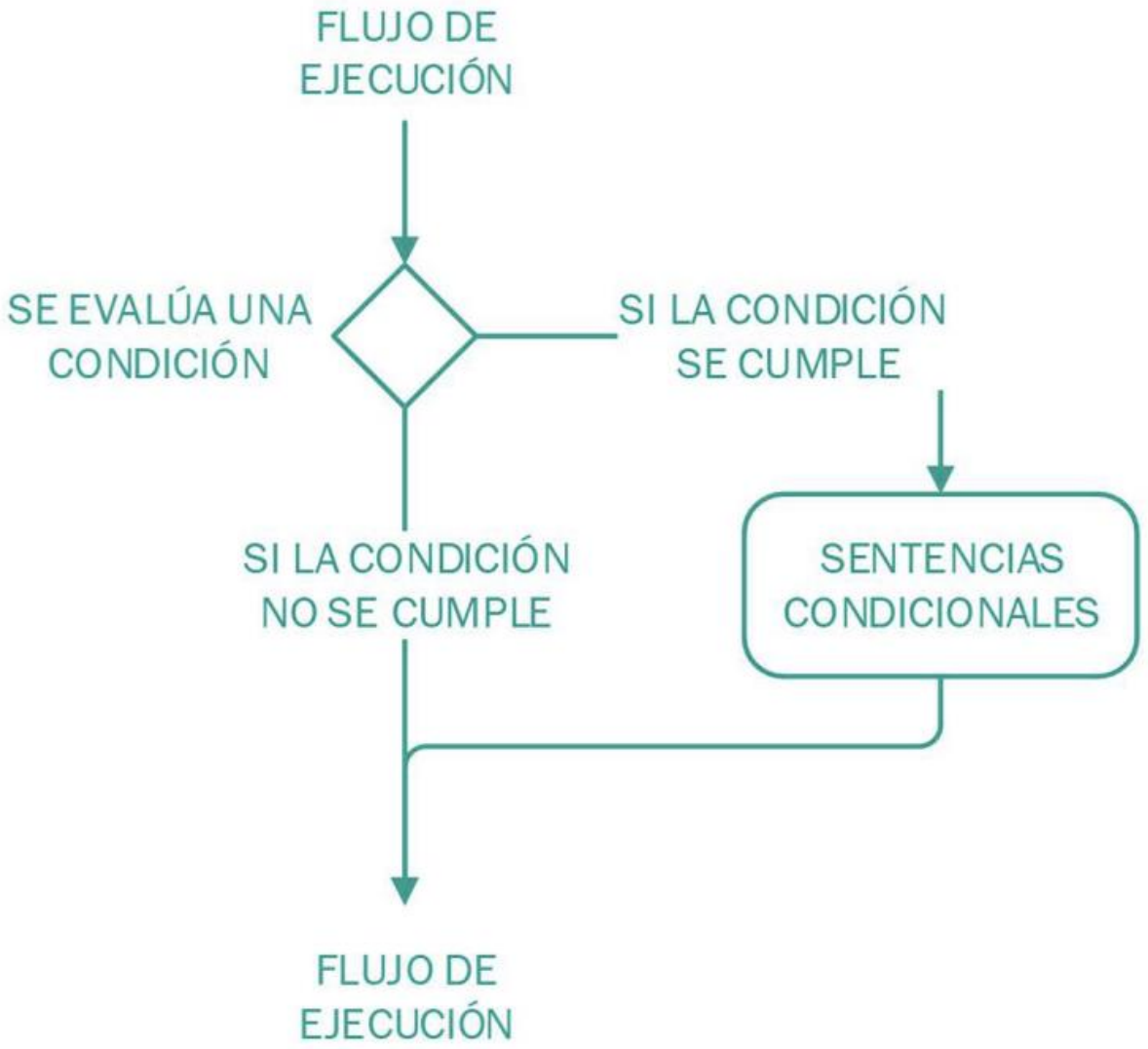
SENTENCIA `if`

- Permite **condicionar la ejecución** de un bloque de sentencias a que una expresión booleana retorne `True`
  - Sintaxis:

```
if <booleano>:
        # Se ejecuta si la condición se cumple
        <Bloque de sentencias condicionales>
<Bloque de sentencias que sigue>
```

- El bloque condicionado debe estar **indentado**. Para volver al flujo no condicionado es necesario volver al nivel previo de **indentación**

SENTENCIA `if`

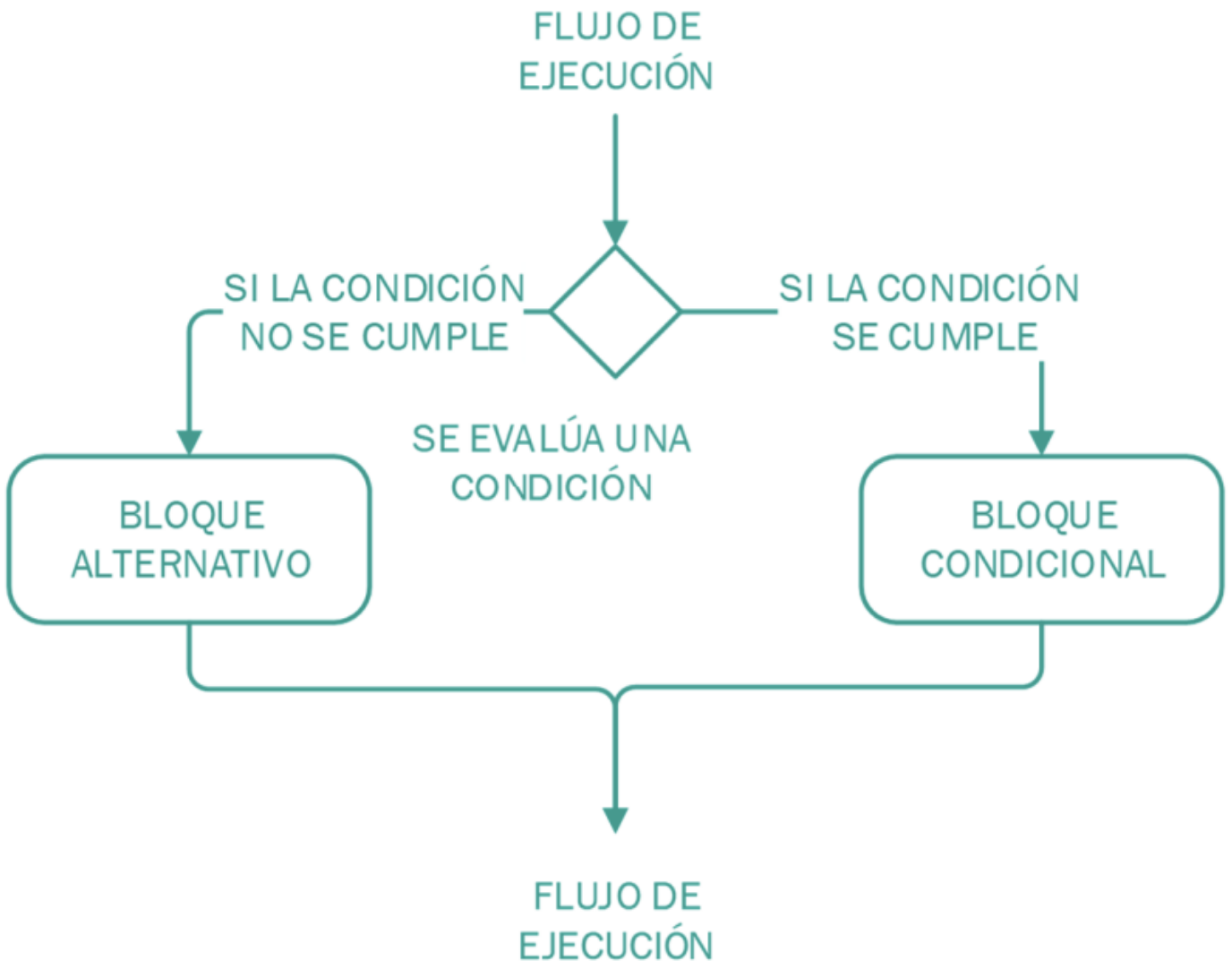


SENTENCIA `if-else`

- Permite **dividir el flujo de la ejecución de dos bloques** a que una expresión booleana retorne `True`
  - Sintaxis:

```
if <booleano>:
    # Se ejecuta si la condición se cumple
    <Bloque de sentencias condicionales>
else:
    # Se ejecuta si la condición no se cumple
    <Bloque de sentencias alternativo>
# Se ejecuta independiente de si la condición se cumplió o no
<Bloque de sentencias que sigue>
```

SENTENCIA `if-else`



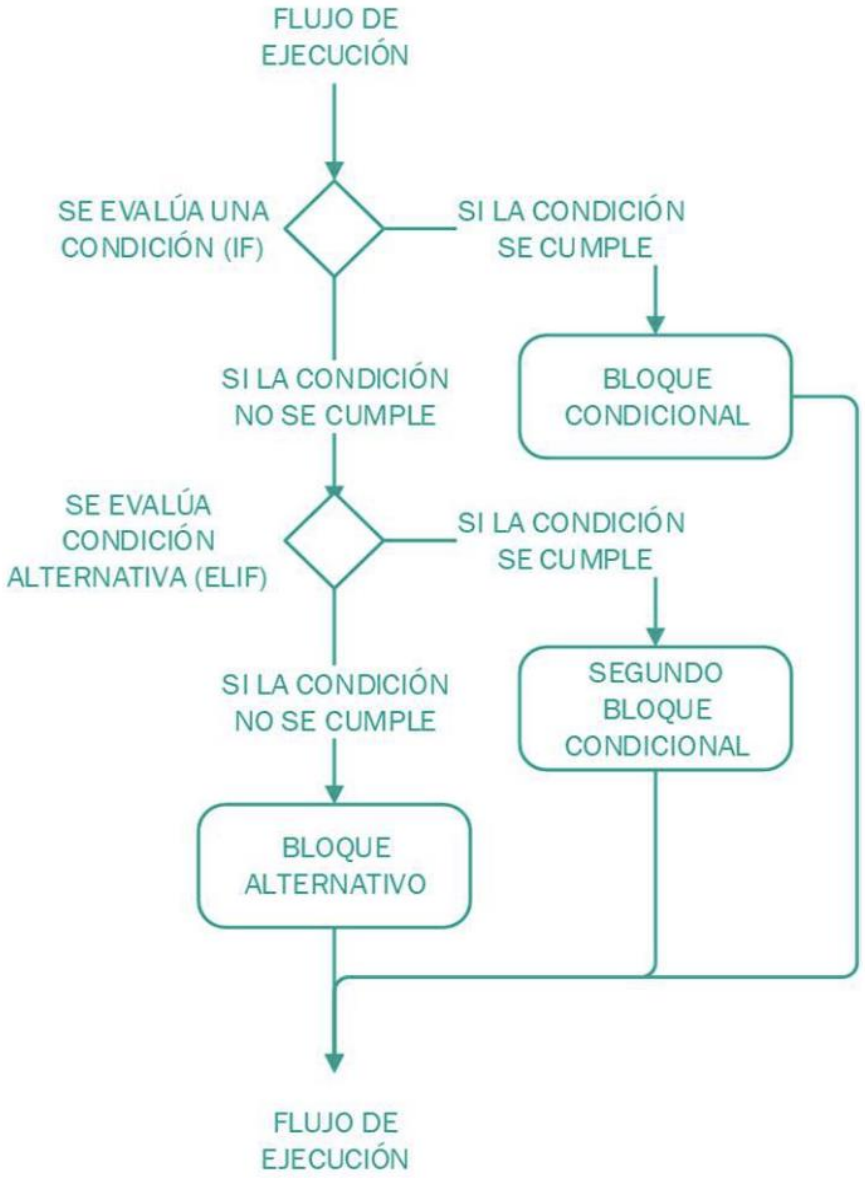
SENTENCIA `if-elif-else`

- Sintaxis:

```
if <booleano>:
    # Se ejecuta si la condición 1 se cumple
    <Bloque de sentencias condicionales 1>
elif <booleano>:
    # Se ejecuta si la condición 1 no se cumple y sí se cumple la
condición 2 (Se puede crear tantos bloques elif como sea necesario,
cada uno con una condición distinta)
    <Bloque de sentencias condicionales 2>
else:
    # Se ejecuta si las condiciones 1 y 2 no se cumplen
    <Bloque de sentencias alternativo>
# Se ejecuta independiente de si la condición se cumplió o no
<Bloque de sentencias que sigue>
```

SENTENCIA `if-elif-else`

DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE



BLOQUES if DISTINTOS

- Es posible definir dos o más bloques **if** distintos, uno a continuación del otro
- En dichos casos la restricción de cumplir las condiciones en orden de aparición **if-elif-elif**... se reinicia al momento de pasar al siguiente bloque de decisiones
- El cumplimiento de las condiciones dentro de un mismo bloque de decisiones es excluyente (solo se considera la primera que se cumpla dentro de un mismo bloque), mientras que en el caso de dos o más bloques de decisiones se verificarán las condiciones de manera independiente (entre bloques) y dependiente (dentro del mismo bloque)

EJERCICIOS

EJERCICIO PROPUESTO 1

- Implemente un programa en Python que determine si un número ingresado es par o impar.

EJERCICIO PROPUESTO 1

- DATOS ENTRADA:
  - Número entero
- DATOS SALIDA:
  - Mensaje indicando paridad del número
- ALGORITMO:
  1. Solicitar el número
  2. Calcular paridad. Si el número es par ir a 3, en caso contrario ir a 4.
  3. Mostrar por consola que el número es par e ir a 5
  4. Mostrar por consola que el número es impar
  5. Fin

EJERCICIO PROPUESTO 1


```
# DATOS ENTRADA
# Solicita un número
numero = int(input("Ingrese un número: "))

# PROCESAMIENTO
# Determina si el número ingresado es par
if (numero % 2) == 0:
        # Salida informa que el número es par
        print("El número", numero, "es par")
# Caso para el número impar
else:
        # Salida informa que el número es impar
        print("El número", numero, "es impar")
```

EJERCICIO PROPUESTO 2

- Construya un programa para determinar cuánto se debe pagar por equis cantidad de lápices, considerando que si son 1000 o más el costo es de 850, de lo contrario, el precio es de 900.

EJERCICIO PROPUESTO 2

- DATOS ENTRADA:
  - Número de lápices
- DATOS SALIDA:
  - Costo total de los lápices
- ALGORITMO:
  1. Solicitar el número de lápices
  2. Si el número de lápices es mayor o igual a mil ir a 3, en caso contrario ir a 4
  3. Calcular costo total con precio 850 e ir a 5
  4. Calcular costo total con precio 900
  5. Mostrar por consola que el costo total de los lápices

EJERCICIO PROPUESTO 2


```
# ENTRADAS
# Solicita el número de lápices
numero_lapices = int(input("Ingrese el número de lápices: "))
 # PROCESAMIENTO
# Determina si el número ingresado es mayor o igual a 1000
 if numero_lapices >= 1000:
            # Calcula el costo total con precio de 850
             costo_total = numero_lapices * 850
else:
            # Calcula el costo total con precio de 900
             costo_total = numero_lapices * 900
# SALIDAS
print("Se debe pagar un total de ", costo_total)
```

EJERCICIO PROPUESTO 3

- Construya un programa en Python que determine si un número ingresado es el doble de un número entero impar.

EJERCICIO PROPUESTO 3

- DATOS ENTRADA:
  - Un número entero
- DATOS SALIDA:
  - Mensaje indicando si el número cumple con la condición
- ALGORITMO:
  1. Solicitar el número
  2. Si el número es par y el número//2 es impar, entonces mostrar por consola que el número es el doble de un impar e ir a 4. En caso contrario ir a 3
  3. Mostrar por consola que el número no es el doble de un impar
  4. Fin

EJERCICIO PROPUESTO 3


```
# ENTRADA
# Solicita un número
numero = int(input("Ingrese un número: "))
# PROCESAMIENTO
# Determina si es el doble de un número impar
if (numero % 2 == 0) and (((numero // 2) % 2) == 1):
        # Informa que el número es el doble de un impar
        print(numero, "es el doble de un número impar")
# Caso contrario, el número no es el doble de un impar
else:
        # Informa que el número no es el doble de un impar
                print(numero, "no es el doble de un número impar")
```

EJERCICIO PROPUESTO 4

- Realice un programa en Python para resolver cualquier ecuación de primer grado

$$ax + b = 0$$

- Con solución:

$$x = \frac{-b}{a}$$

EJERCICIO PROPUESTO 4

- DATOS ENTRADA:
  - Coeficientes a y b
- DATOS SALIDA:
  - Mensaje con la solución (tres casos)
- ALGORITMO:
  1. Solicitar los coeficientes a y b
  2. Si a ≠ 0 mostrar la solución –b/a e ir a 5, en caso contrario ir a 3
  3. Si b = 0 mostrar por consola que la ecuación tiene infinitas soluciones e ir a 5, en caso contrario ir a 4
  4. Mostrar por consola que la ecuación no tiene solución
  5. Fin

EJERCICIO PROPUESTO 4

DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

```
# ENTRADAS
a = float(input("Ingrese el valor del coeficiente a: "))
b = float(input("Ingrese el valor del coeficiente b: "))
# PROCESAMIENTO
# Determina si coeficiente a es distinto a cero
if a != 0:
        solucion = -b / a
        #Salida
        print("La solución de la ecuación es: ", solucion)
# Determina si coeficiente b es igual a cero
 elif b == 0:
        #Salida
        print("La ecuación tiene infinitas soluciones")
else:
        #Salida
         print("La ecuación no tiene solución")
```

EJERCICIO PROPUESTO

- Construye un programa que solicite el valor de dos ángulos interiores de un triángulo e indique si un triángulo es equilátero, isósceles o escaleno

  - Recuerda que los ángulos interiores siempre sumarán 180°.
  - Las entradas siempre serán números enteros positivos con valores de ángulos válidos.

EJERCICIO PROPUESTO

- Construye un programa que reciba una cantidad de dinero y entregue el desglose de este en billetes y monedas válidos en circulación en Chile (20.000, 10.000, 5.000, 2.000, 1.000, 500, 100, 50, 10, 5, 1)

- Por ejemplo, si la entrada fuese $20.250, el programa debería entregar:
  - 1 billete(s) de $20.000
  - 2 moneda(s) de $100
  - 1 moneda(s) de $50

EJERCICIO PROPUESTO

- Construya un programa en Python que solicite una fecha, es decir, el valor del año, mes y día, e indique si dicha fecha existe.

- Considere para su cálculo que el programa debería reconocer y diferenciar los meses que tienen 28, 30 y 31 días, además de determinar cuándo un año es bisiesto y cuándo no.

PREGUNTA TIPO QUIZ

- ¿Qué imprime el programa presentado a continuación?

```
x = 8
y = 12
z = 1

res = x * (y ** z)
if x % 2 == 0 and y % 2 == 0 and z % 2 == 0 :
        res = res - x
if x % 3 == 0 and y % 3 == 0 and z % 3 == 0 :
        res = res - y
if x % 5 == 0 and y % 5 == 0 and z % 5 == 0 :
        res = res - z
if x % 2 == 0 or y % 2 == 0 or z % 2 == 0 :
        res = res + x
if x % 3 == 0 or y % 3 == 0 or z % 3 == 0 :
        res = res - y
if x % 5 == 0 or y % 5 == 0 or z % 5 == 0 :
        res = res + z

print('res: ',res)
```

a) res: 72

b) res: 92

c) res: 96

d) res: 104

TAREAS PARA TRABAJO AUTÓNOMO

DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

1. (Después de la clase del viernes) Revisar el apunte:
   - Iteraciones en Google Colab (Disponible en:https://github.com/PROGRA-FING-USACH/Material/blob/main/Lecturas/04%20_Iteracion.ipynb)

2. Revisar las guías 1, 2 y 3

3. Responder el cuestionario (OPCIONAL): https://forms.gle/7LUeAoeFa4Gmq8fZA

¿CONSULTAS?