

## *Bifurcación*

León, R. (2021). *Bifurcación* [apunte].  
Chile. UNAB

## BIFURCACIÓN

### 1. Introducción

Un programa se constituye por una sucesión de *sentencias* que son ejecutadas de forma secuencial. Por ejemplo, en la Figura 1 podemos observar un programa que contiene cuatro sentencias.

**Figura 1**

Programa con cuatro sentencias.

Programa
<pre>n = int(input("ingrese n: ")) m = int(input("ingrese m: ")) s = n + m print(f"La suma de {n} + {m} es {s}")</pre>
Salida del programa
ingrese n: 3 ingrese m: 6 La suma de 3 + 6 es 9

1

Estas son sentencias simples, las primeras dos permiten obtener un valor desde el usuario, la tercera línea calcula la suma de los enteros ingresados, y finalmente se imprime el resultado.

Además de estas sentencias, existen las *sentencias de control* que permiten realizar otros tipos de flujos dentro del programa. Estas instrucciones permiten construir *sentencias condicionales (bifurcación)* y *sentencias de ciclos (repetición)*.

Un *condicional* es un conjunto de sentencias que pueden o no ejecutarse, dependiendo del resultado de una condición. Un *ciclo* es un conjunto de sentencias que son ejecutadas varias veces, hasta que una condición de término es satisfecha. Tanto los condicionales como los ciclos contienen a otras sentencias. Para indicar esta relación se utiliza la *indentación*: las sentencias contenidas no se escriben en la misma columna que la sentencia de control, sino con una sangría hacia la derecha.

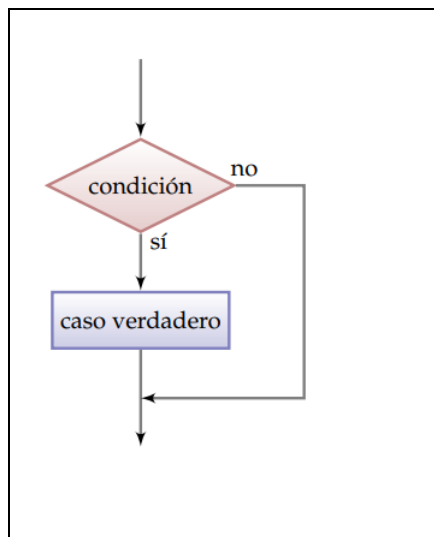
## 2. Bifurcación

Como se mencionó anteriormente un *condicional* es un conjunto de sentencias que pueden o no ejecutarse, dependiendo del resultado de una condición.

La sentencia *if* (*si*) ejecuta un bloque de instrucciones solo si se cumple una condición. Si la condición es falsa, el programa sigue su flujo.

**Figura 2**

Funcionamiento de la sentencia **if**.



La sintaxis de la sentencia *if* es la siguiente:

**if** condicion:  
sentencias

Por ejemplo, en la Figura 3 se muestra un programa que felicita a alguien que ha aprobado un ramo. Como se puede apreciar, cuando la condición es falsa, el mensaje no se imprime.

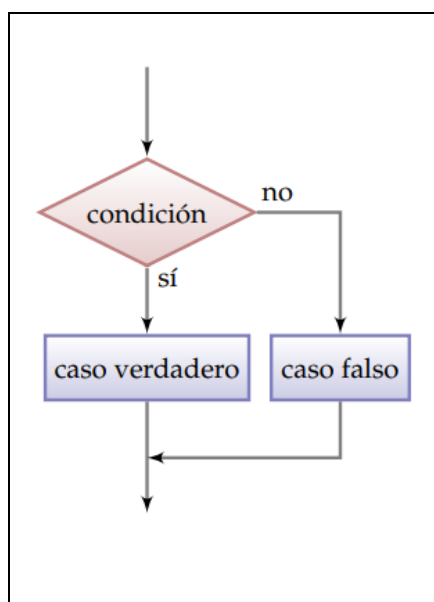
**Figura 3**  
Ejemplo sentencia **if**.

Programa
<pre>nota = float(input("ingrese su nota: "))  if nota &gt;= 3.95:     print(f"Felicitaciones! Ud. ha aprobado con nota {nota}")</pre>
Salida 1 del programa
ingrese su nota: 5.5 Felicitaciones! Ud. ha aprobado con nota 5.5
Salida 2 del programa
ingrese su nota: 3.2

La sentencia *if-else* (*si-o-si-no*) decide que instrucciones ejecutar dependiendo si una condición es verdadera o falsa.

3

**Figura 4**  
Funcionamiento de la sentencia **if-else**.



La sintaxis de la sentencia *if-else* es la siguiente:

**if** condicion:

    sentencias para la condición verdadera

**else:**

    sentencias para la condición falsa

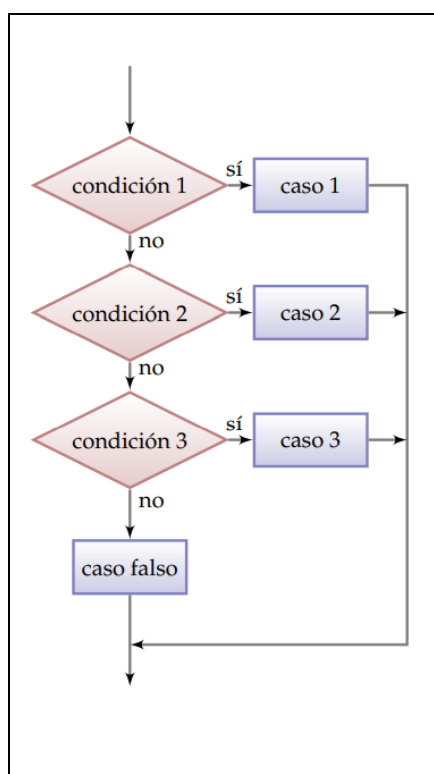
Por ejemplo, en la Figura 5 se muestra un programa que realiza acciones distintas dependiendo si el número es par o impar.

**Figura 5**  
Ejemplo sentencia **if-else**.

Programa
<pre>numero = int(input("ingrese numero: ")) resto = numero % 2  if resto == 0:     print(f"El numero es par")     mitad = numero // 2     print(f"La mitad de {numero} es {mitad}") else:     print(f"El numero es impar")     sucesor = numero + 1     print(f"El sucesor de {numero} es {sucesor}")</pre>
Salida 1 del programa
ingrese numero: 24 El numero es par La mitad de 24 es 12
Salida 2 del programa
ingrese numero: 57 El numero es impar El sucesor de 57 es 58

La sentencia *if-elif-else* (*si-o-si-no*) depende de dos o más condiciones, que son evaluadas en orden. La primera condición que sea verdadera es el bloque de instrucciones que se ejecuta. La sentencia *else* al final es opcional, y el bloque de instrucciones de esta se ejecuta cuando ninguna de las condiciones anteriores es verdadera.

**Figura 6**  
Funcionamiento de la sentencia **if-elif-else**.



La sintaxis de la sentencia *if-elif-else* es la siguiente:

```
if condicion1:  
    sentencias para la condición 1 verdadera  
elif condicion2:  
    sentencias para la condición 2 verdadera  
...  
else:  
    sentencias si ninguna condición es verdadera
```

Por ejemplo, en la Figura 7 se muestra un programa que dependiendo de la figura geométrica que elige el usuario, se calcula el área de la misma.

**Figura 7**  
Ejemplo sentencia **if-elif-else**.

Programa
<pre> figura = input("Ingrese figura (T:triángulo, C:cuadrado, R:rectángulo): ")  if figura == "T":     b = float(input("Ingrese base: "))     h = float(input("Ingrese altura: "))     area = (b * h) / 2     print(f"El área del triángulo es {area}") elif figura == "C":     a = float(input("Ingrese lado: "))     area = a ** 2     print(f"El área del cuadrado es {area}") elif figura == "R":     w = float(input("Ingrese ancho: "))     h = float(input("Ingrese altura: "))     area = w * h     print(f"El área del rectángulo es {area}") else:     print("Figura no válida") </pre>
Salida 1 del programa
Ingrese figura (T:triángulo, C:cuadrado, R:rectángulo): C Ingrese lado: 3.5 El área del cuadrado es 12.25
Salida 2 del programa
Ingrese figura (T:triángulo, C:cuadrado, R:rectángulo): T Ingrese base: 2.5 Ingrese altura: 20 El área del triángulo es 25.0
Salida 3 del programa
Ingrese figura (T:triángulo, C:cuadrado, R:rectángulo): S Figura no válida
Salida 4 del programa
Ingrese figura (T:triángulo, C:cuadrado, R:rectángulo): R Ingrese ancho: 6.5 Ingrese altura: 3.2 El área del rectángulo es 20.8

Siempre solo una alternativa será ejecutada. Tan pronto alguna de las condiciones sea verdadera, el resto de las condiciones no son evaluadas.

*"Para complementar la información y contenidos presentados, puedes dirigirte a":*



## Bibliografía

1. **Downey, A. (2013). Think Python. Green Tea Press.** Accesado el 2 de diciembre de 2019 en <https://greenteapress.com/wp/think-python-2e/>
2. **Python software foundation, Python v3 Documentation.** Accesado el 2 de diciembre de 2019 en <https://docs.python.org/3/>
3. Marzal Varó, A., Gracia Luengo, I., & García Sevilla, P. (2014). Introducción a la programación con Python 3. Universitat Jaume I. <https://doi.org/10.6035/sapientia93> (Capítulo 1, secciones 1.1 y 1.2, páginas: 11 - 16)