

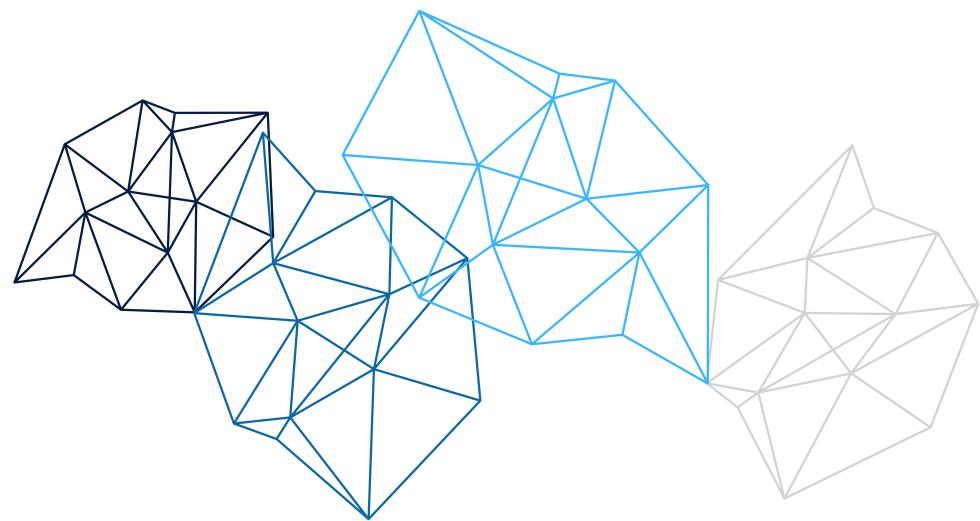
Estructuras de control

Si un programa sólo ejecutara instrucciones planas unas tras otras no servirían de mucho. Pues, el hecho que las acciones se ejecuten secuencialmente implica que nunca se ejecutara más de una acción al mismo tiempo

Por suerte ahí es donde aparecen las estructuras de control, las cuales van a permitir que el flujo del programa se adapte y sepa cómo actuar ante determinadas situaciones e incluso repetir una tarea si es necesario.

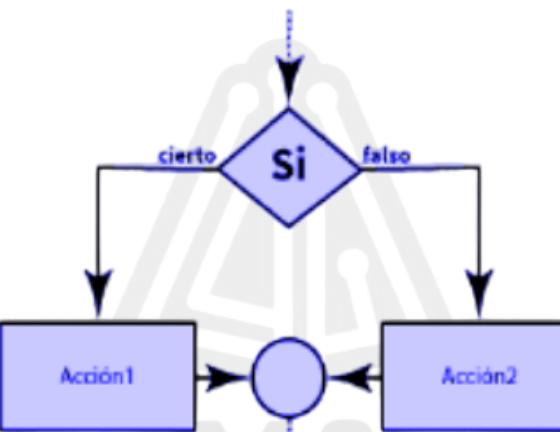
¿Qué tipos de estructuras veremos en el curso?

- Condicionales (simple, alternativo o doble, etc.)
- Repetitivas (for, while, loops, etc.)



Condicionales

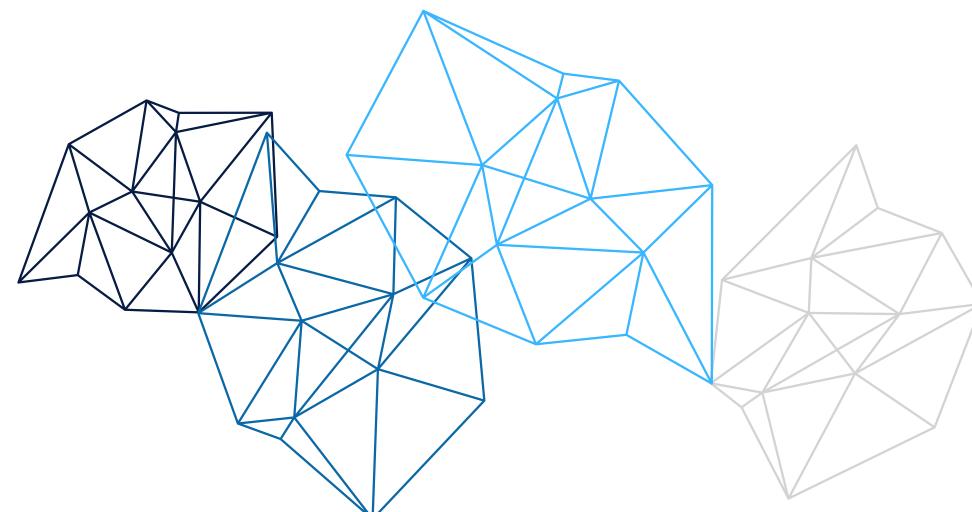
No todos los problemas pueden resolverse empleando estructuras secuenciales. Cuando hay que tomar una decisión aparecen las estructuras condicionales.



Cómo funciona?

En las estructuras condicionales, se realiza una evaluación de una condición y de acuerdo al resultado, el programa realiza una determinada acción. Las condiciones son especificadas utilizando expresiones lógicas.

INFORMATARIO
Hacia una mejor industria informática



Condicional simple: IF (SI)

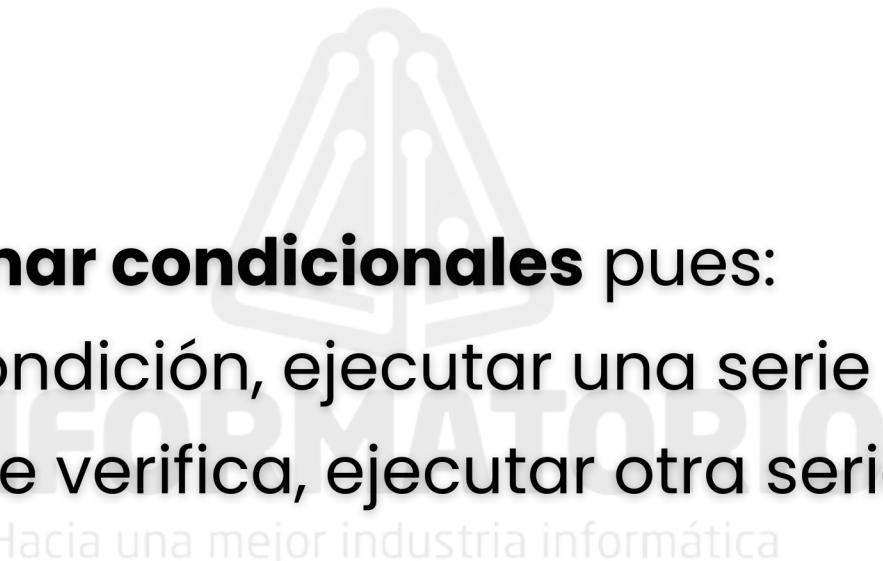
La sentencia if se ejecuta siempre que la expresión que comprueba devuelve True:

```
>>if True: # equivale a if not False  
    print("Se cumple la condición")  
    print("También se muestre este print")  
  
#Se cumple la condición  
  
# También se muestre este print
```

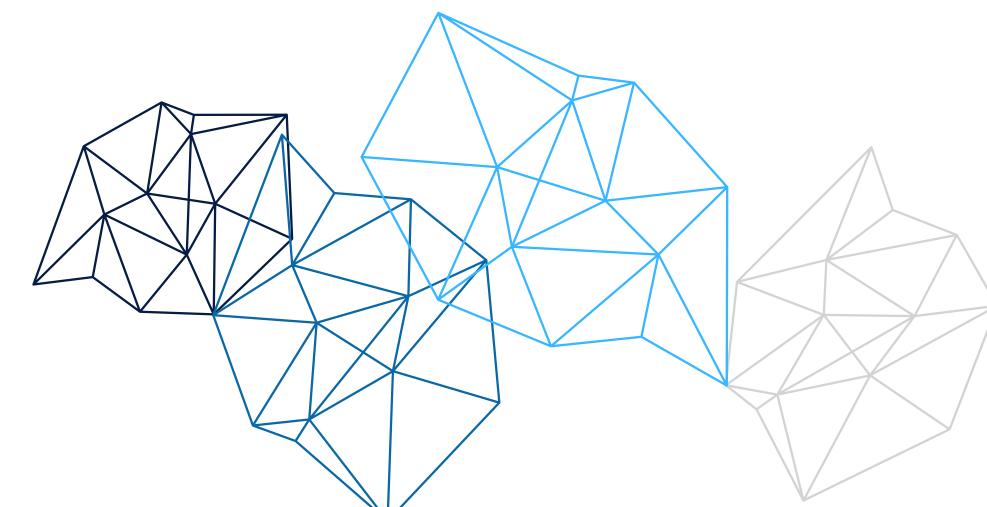
Puede ocurrir que necesites **encadenar condicionales** pues:

- Si se verifica una determinada condición, ejecutar una serie de instrucciones (bloque 1).
- Si no, esto es, si la condición NO se verifica, ejecutar otra serie de instrucciones (bloque 2).

```
>>a = 5  
  
if a == 2:  
    print("a vale 2")  
  
if a == 5:  
    print("a vale 5") #a vale 5
```



Hacia una mejor industria informática



Anidados

O también anidar If dentro de If:

```
>>a = 5  
b = 10  
if a == 5:  
    print("a vale",a)  
    if b == 10:  
        print("y b vale",b)  
#a vale 5  
#y b vale 10
```



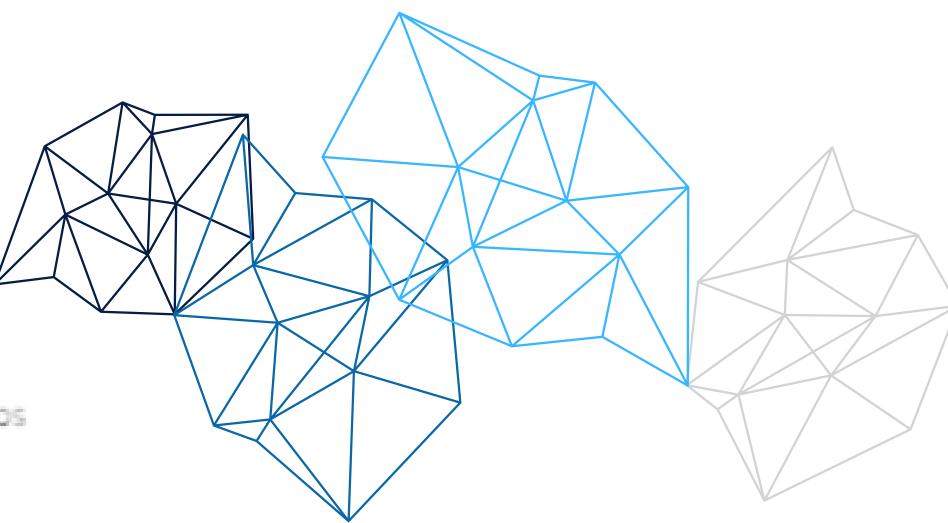
Subsecretaría de
Empleo



Ministerio de
Producción, Industria y Empleo



CHACO
Gobierno de todos



Multiples condiciones

Como condición podemos evaluar múltiples expresiones, siempre que éstas devuelvan True o False:

```
>>if a==5 and b == 10:  
    print("a vale 5 y b vale 10")
```

Solo si los dos son verdaderos devuelve:

```
#a vale 5 y b vale 10  
>>if a == 5 or b == 10:  
    print("a vale 5 o b vale 10")
```

Si al menos un de los dos es True (verdadero) devuelve:

```
#a vale 5 o b vale 10
```



INFORMATARIO

Hacia una mejor industria informática



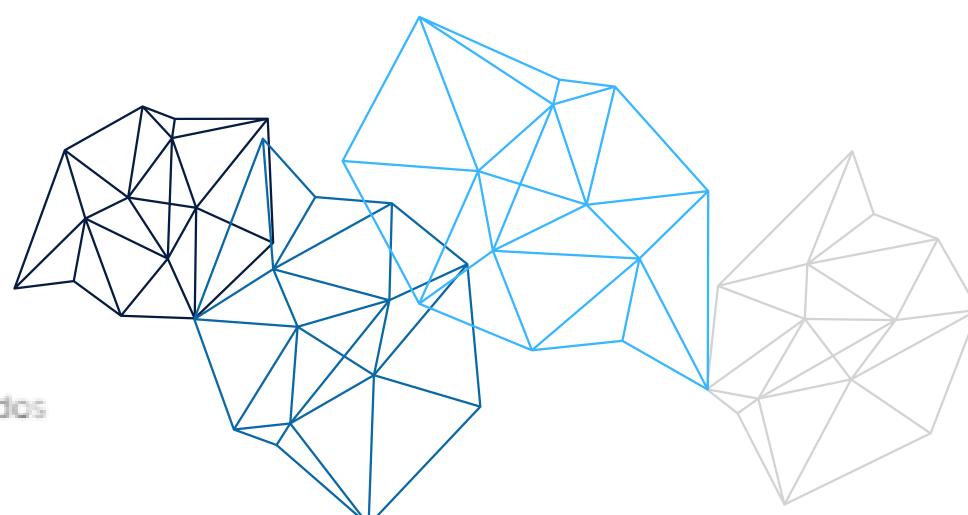
Subsecretaría de
Empleo



Ministerio de
Producción, Industria y Empleo



CHACO
Gobierno de todos



Condicional alternativo o doble: Sentencia else (sino)

Se encadena a un If para comprobar el caso contrario (en el que no se cumple la condición):

```
>>n = 11  
if n % 2 == 0:  
    print(n,"es un número par")  
else:  
    print(n,"es un número impar")  
#11 es un número impar
```



INFORMATORIO
Hacia una mejor industria informática



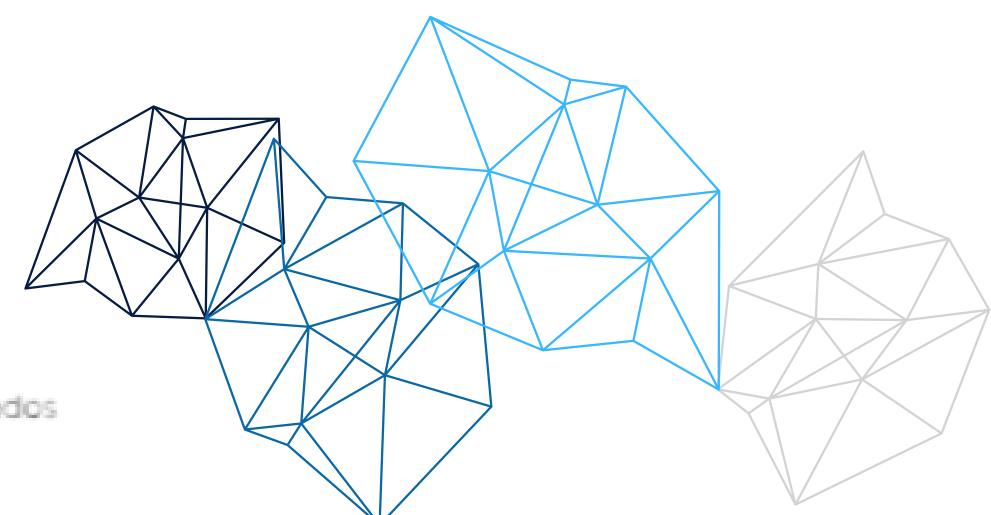
Subsecretaría de
Empleo



Ministerio de
Producción, Industria y Empleo



CHACO
Gobierno de todos

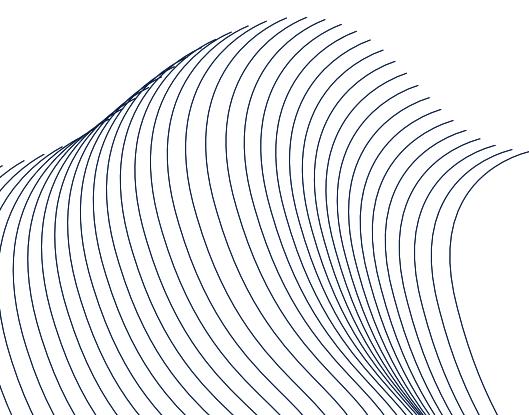


Condicional multiple: Sentencia elif (sino si)

Con frecuencia es necesario que existan más de dos elecciones posibles. Este problema se podría resolver por estructuras selectivas simples o alternativas, encadenadas. Pero si el número de alternativas es grande puede plantear serios problemas de escritura y de legibilidad.

Por este motivo, en estos casos, se recomienda el uso de la **estructura condicional alternativo múltiple** que permite evaluar una variable que puede tomar de 1 a n valores y según ocurra uno de esos valores, se realizará una de las n acciones; es decir, que el programa seguirá sólo un determinado camino entre varios.

INFORMATORIO
Hacia una mejor industria informática



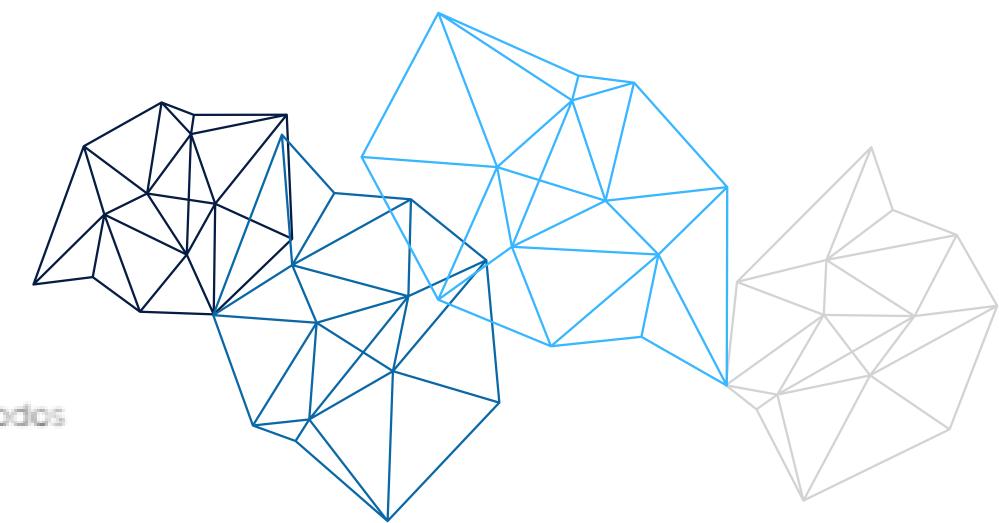
Subsecretaría de
Empleo



Ministerio de
Producción, Industria y Empleo



CHACO
Gobierno de todos



Estructura condicional alternativo múltiple

En Python se encadena a un if u otro elif para comprobar múltiples condiciones, siempre que las anteriores no se ejecuten:

```
>>nota = float(input("Ingresar una nota: "))

if nota >= 9:
    print("Sobresaliente")
elif nota >= 7:
    print("Notable")
elif nota >= 6:
    print("Bien")
elif nota >= 5:
    print("Suficiente")
else:
    print("Insuficiente")
```

