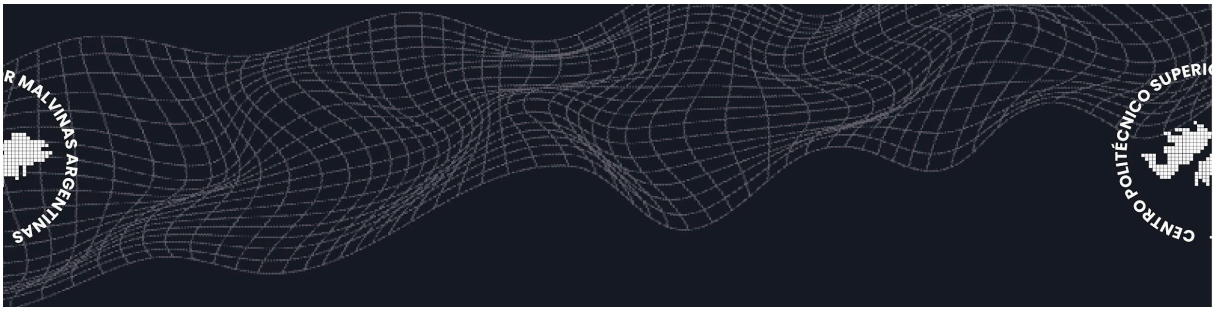


PROGRAMACIÓN 1

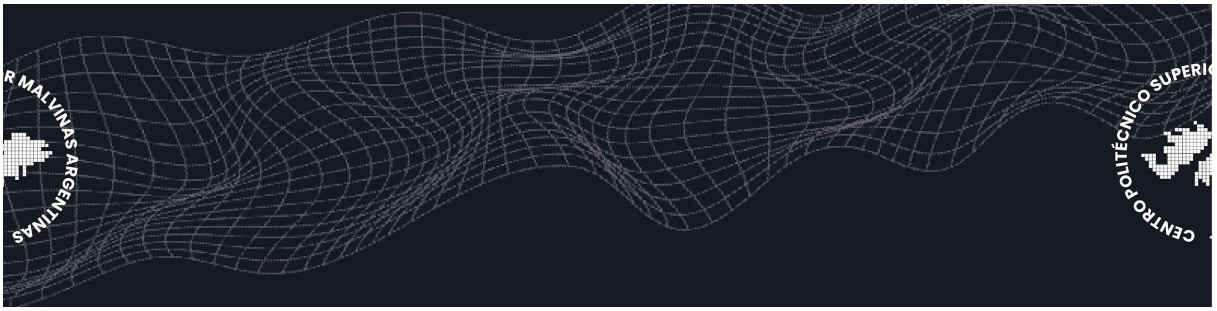
1° AÑO

Clase Nº 4: Flujo de control I - Estructuras selectivas

Contenido

En la clase de hoy trabajaremos los siguientes temas:

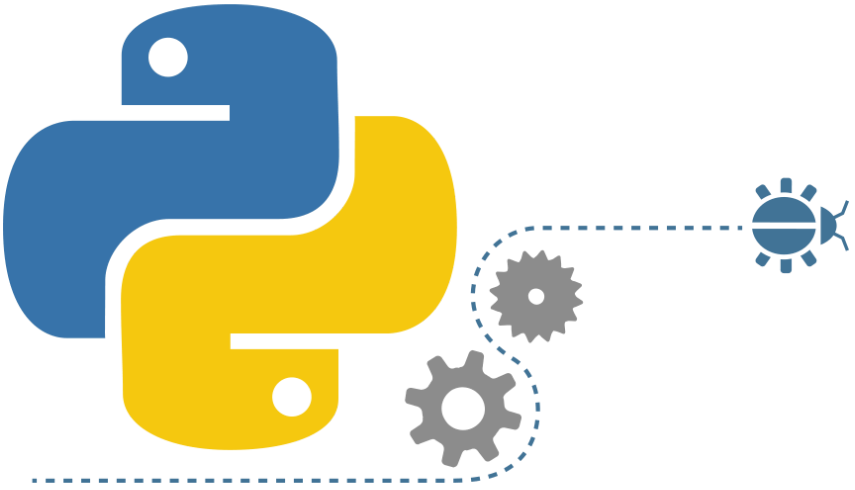
- Flujo de control de un programa.
- Estructura secuencial.
- Sentencia if.
- Alternativa simple.
- Sentencia if y else.
- Alternativa doble.
- Sentencia elif.
- Alternativa múltiple.
- Estructuras de decisión anidadas.
- Condiciones más complejas.

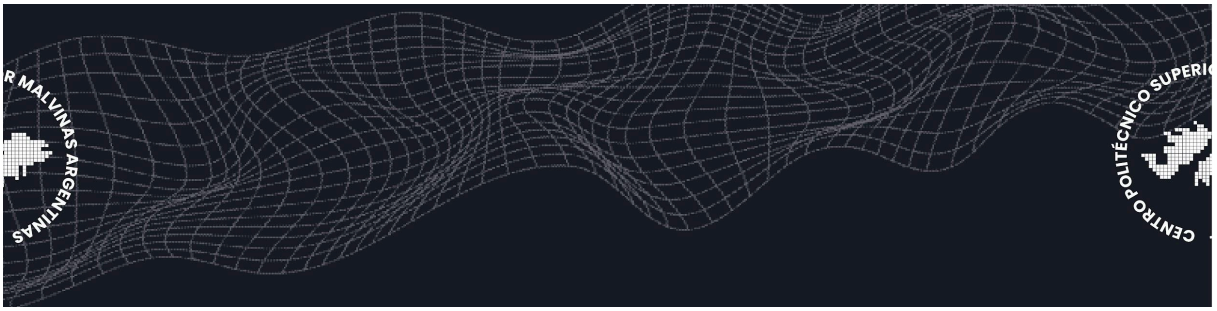


1. Presentación

¡Bienvenidos al emocionante mundo de las estructuras selectivas en Python!

Aquí aprenderás cómo tomar decisiones inteligentes en tus programas usando las declaraciones "if", "else" y "elif". Descubrirás cómo crear lógica condicional, evaluar expresiones y guiar el flujo de tu código según condiciones específicas. ¡Prepárate para potenciar tus habilidades de programación y hacer que tus programas tomen decisiones informadas!

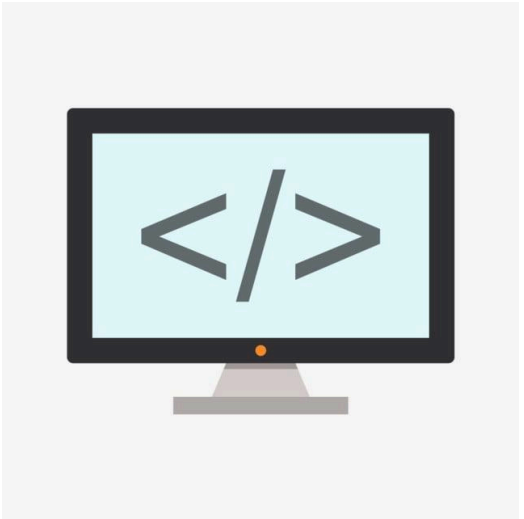


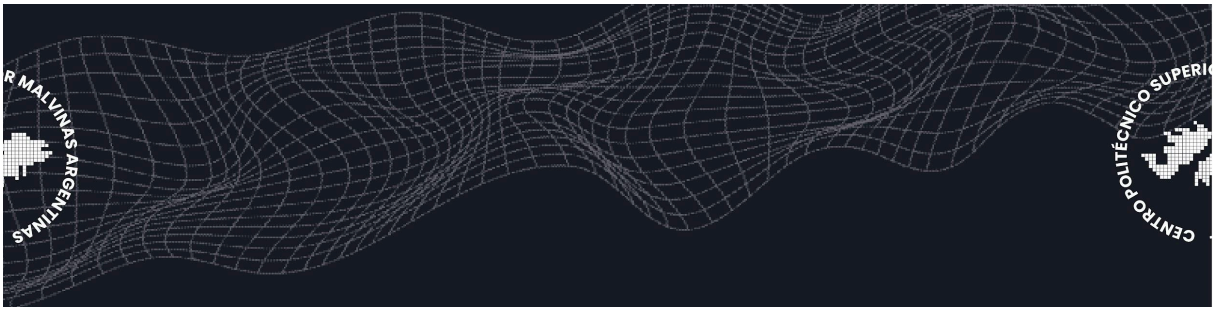


2. Desarrollo

Flujo de control de un programa

El flujo de control se refiere a la secuencia en la que se ejecutan las instrucciones de un programa y cómo se toman decisiones o se repiten ciertas partes del código según ciertas condiciones. En Python, el flujo de control se maneja mediante estructuras como instrucciones condicionales (if, elif, else) y bucles (for, while). Estas estructuras permiten controlar la dirección y la repetición de la ejecución del programa.





Estructura secuencial

La estructura secuencial es un concepto fundamental en programación que se refiere a la ejecución ordenada de instrucciones en un programa, una después de la otra, siguiendo una secuencia lógica. En Python, la estructura secuencial permite ejecutar una serie de declaraciones en el orden en que están escritas, sin realizar saltos ni desvíos.

Estructura secuencial en Python que suma dos números ingresados por el usuario:

```python
# Ingreso de datos por el usuario
numero1 = float(input("Ingrese el primer número: "))
numero2 = float(input("Ingrese el segundo número: "))

# Suma de los números
suma = numero1 + numero2

# Mostrar el resultado
print(f"La suma de {numero1} y {numero2} es: {suma}")
```

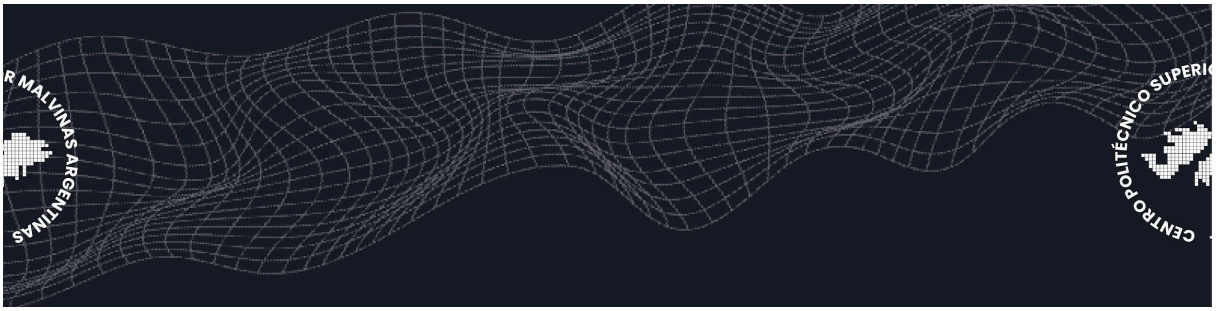
1- En este caso, las instrucciones también se ejecutan en secuencia:

2- El usuario ingresa el primer número.

3- El usuario ingresa el segundo número.

4- Se calcula la suma de los dos números ingresados.

5- Se muestra el resultado de la suma en la pantalla.



Estructuras selectivas

Las estructuras selectivas son bloques de programación que permiten que un programa tome decisiones y realice diferentes acciones en función de ciertas condiciones o situaciones. Estas estructuras son fundamentales para controlar el flujo de ejecución en un programa y permiten que éste se adapte y tome diferentes caminos según las circunstancias.

La estructura secuencial se refiere a la ejecución de instrucciones en un orden lineal, donde cada instrucción se ejecuta una tras otra.

```python
nombre = input("Ingrese su nombre: ")
print("Hola,", nombre)
edad = int(input("Ingrese su edad: "))
print("Usted tiene", edad, "años.")
```

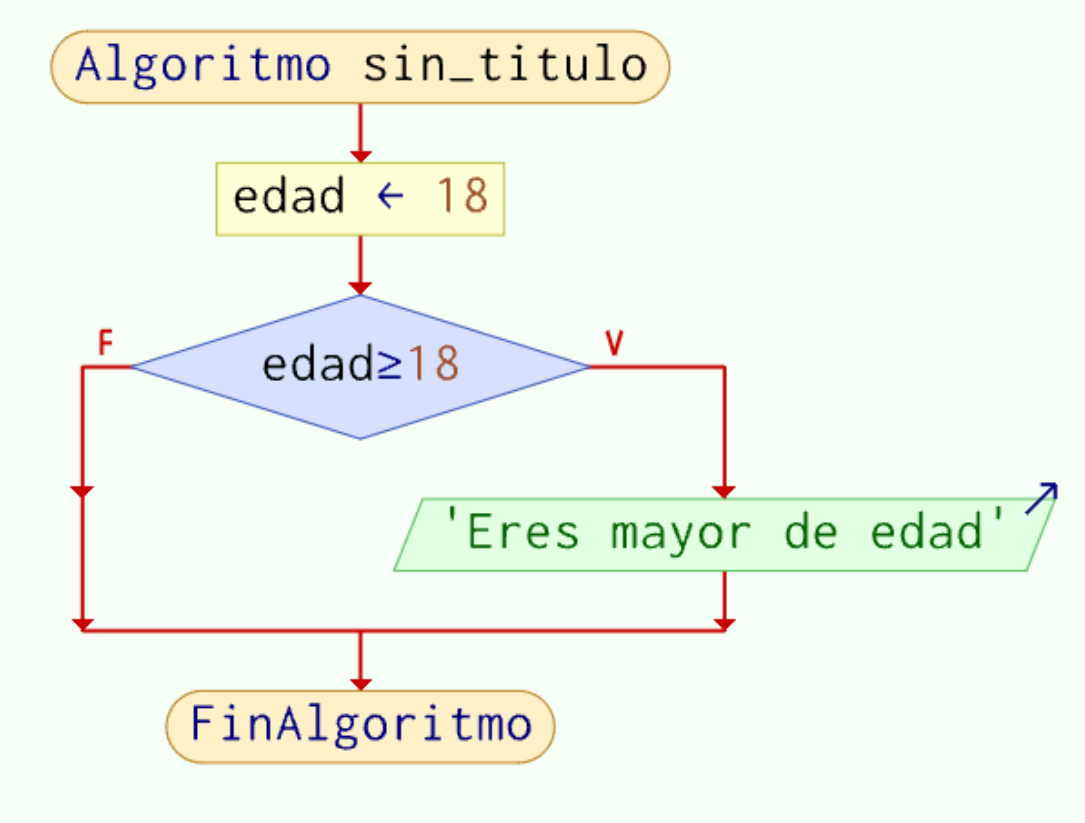
Sentencia if

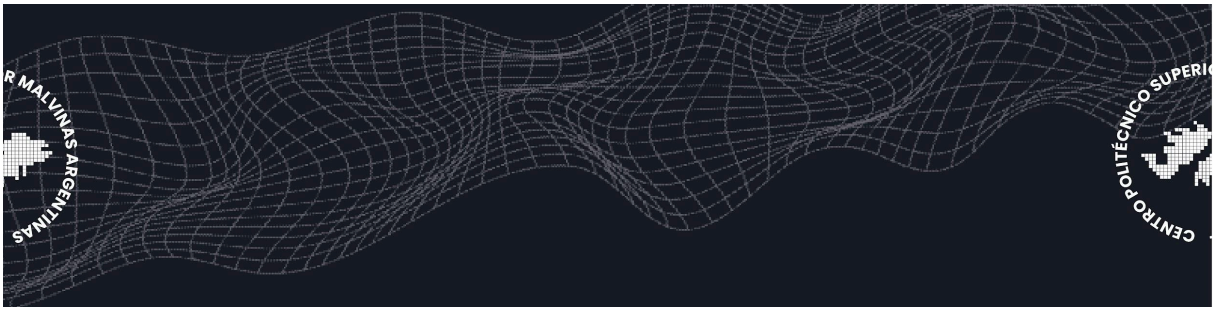
La sentencia "if" (si, en inglés) es una estructura fundamental en programación que permite controlar el flujo de ejecución de un programa en función de una condición. Esta condición puede ser verdadera o falsa, y en función de su evaluación, el programa tomará diferentes caminos y realizará diferentes acciones.

Alternativa simple

```python
edad = 18

if edad >= 18:
    print("Eres mayor de edad")
```





Sentencias if y else

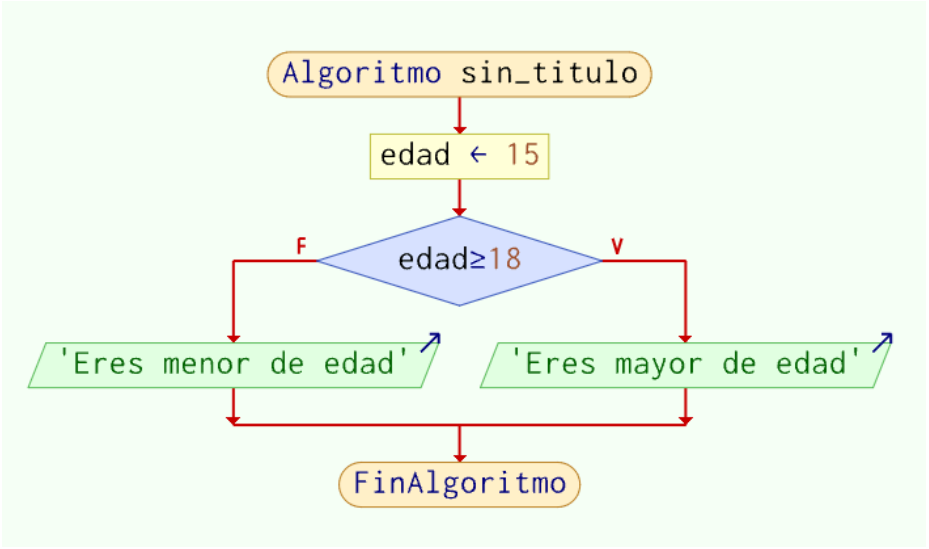
La instrucción "else" se utiliza junto con "if" para ejecutar un bloque de código cuando la condición del "if" no es verdadera.

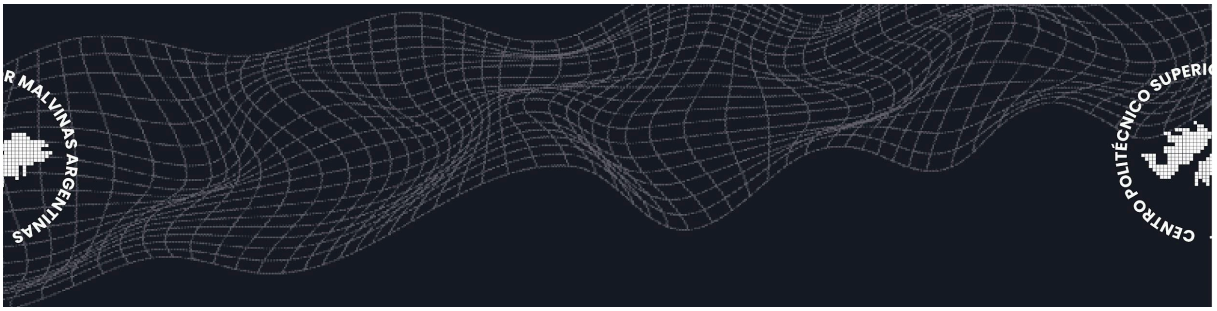
La sentencia "if" y "else" son estructuras fundamentales en programación que permiten tomar decisiones en función de una condición. Estas estructuras controlan el flujo de ejecución del programa al evaluar una expresión booleana y ejecutar diferentes bloques de código según si la condición es verdadera o falsa.

Alternativa doble

```python
edad = 15

if edad >= 18:
    print("Eres mayor de edad")
else:
    print("Eres menor de edad")
```





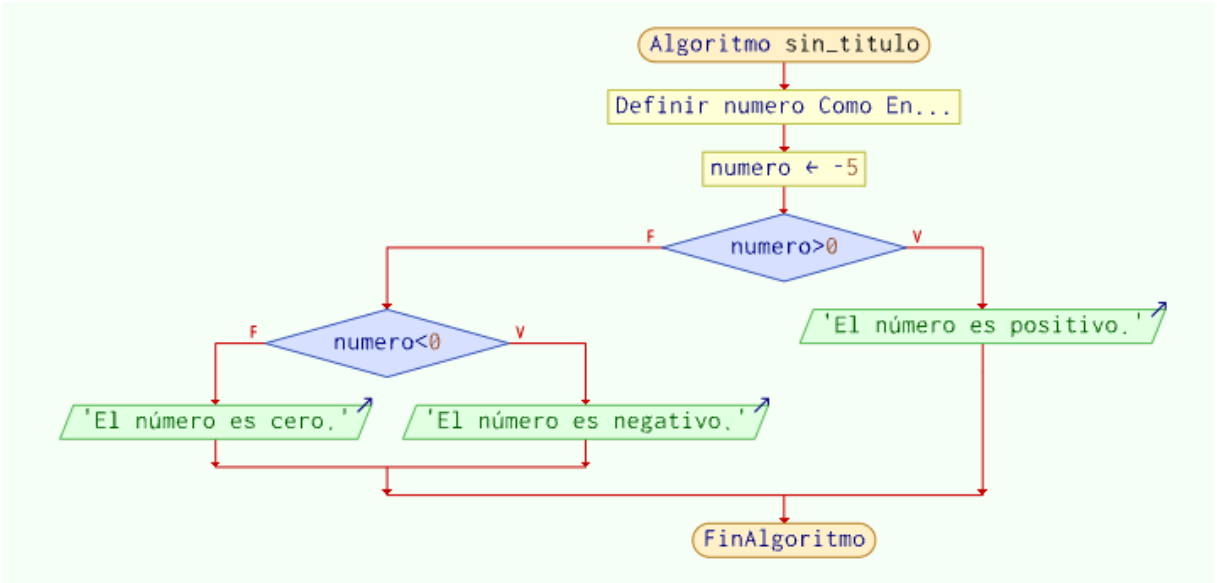
Sentencia elif

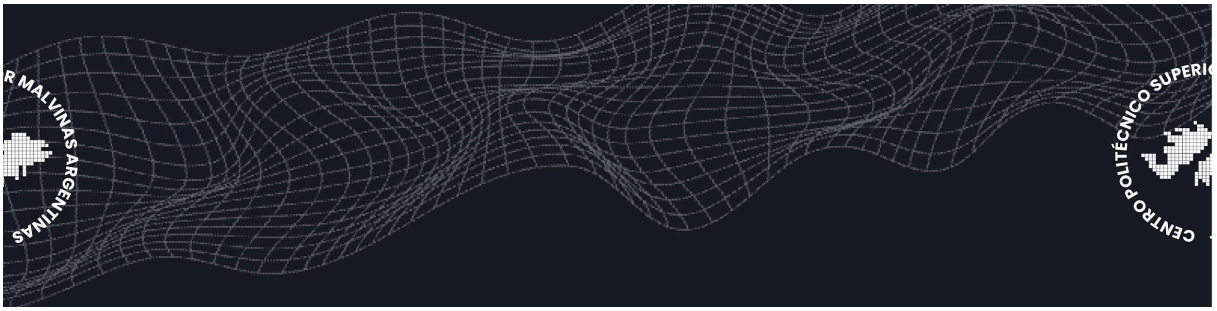
elif es una abreviatura de "else if" en muchos lenguajes de programación, incluido Python. Se utiliza como parte de una estructura condicional para evaluar múltiples condiciones en orden y ejecutar el bloque de código asociado a la primera condición que sea verdadera.

Alternativa múltiple

```python
numero = -5  # Cambia este valor para probar diferentes números

if numero > 0:
    print("El número es positivo.")
elif numero < 0:
    print("El número es negativo.")
else:
    print("El número es cero.")
```





Estructuras de Decisión Anidadas

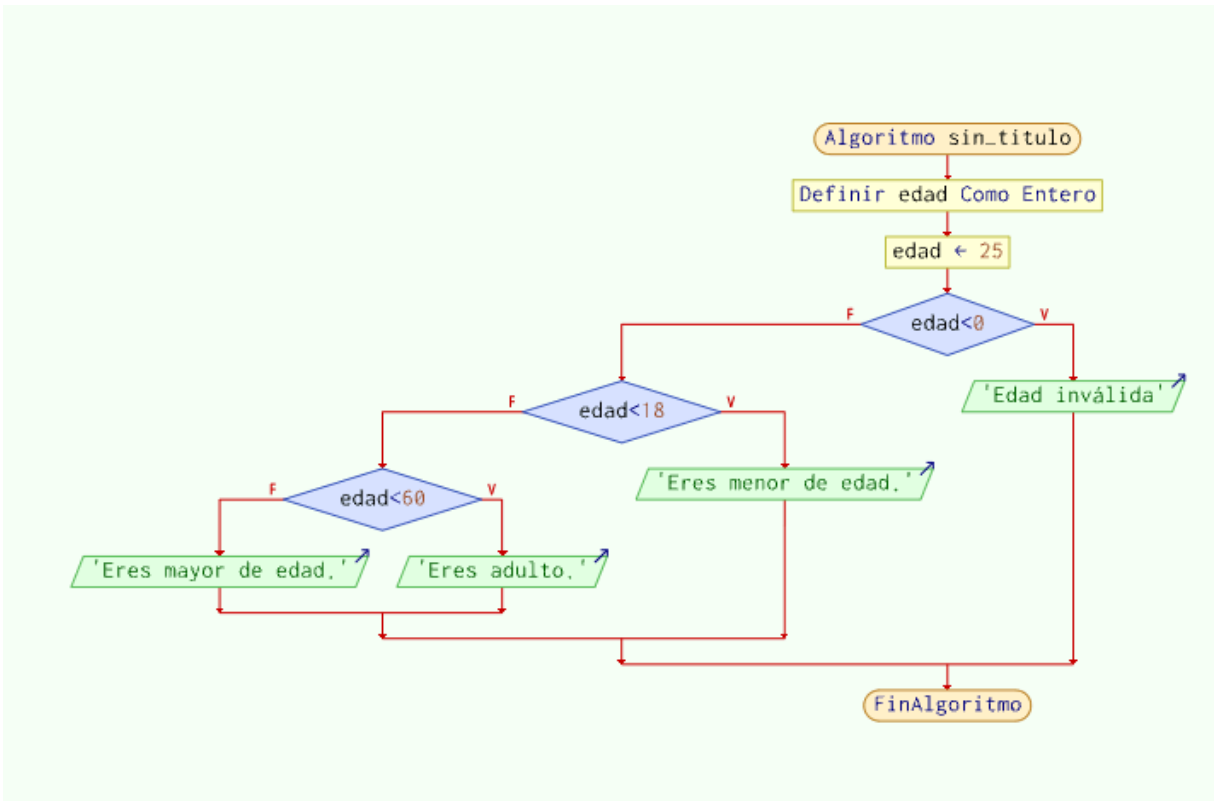
Una estructura de decisión anidada es un concepto de programación que implica la inclusión de múltiples bloques de decisión dentro de otro bloque de decisión. En otras palabras, es la combinación de múltiples estructuras de decisión (como declaraciones "if" o "switch") dentro de una sola estructura, con el propósito de manejar una variedad más compleja de condiciones y escenarios.

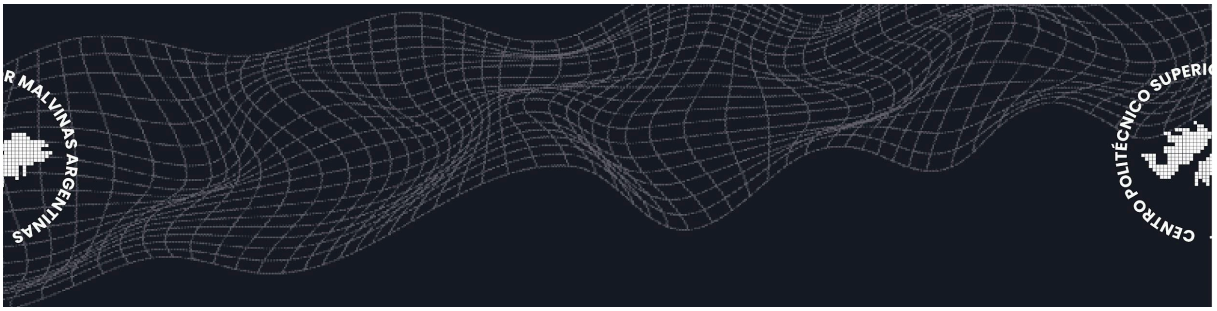
Esta anidación de estructuras de decisión permite que un programa evalúe condiciones en varias etapas y tome diferentes rutas de acción en función de la combinación de condiciones que se cumplen. En esencia, las estructuras de decisión anidadas permiten crear lógica más sofisticada y detallada en el flujo de ejecución de un programa.

```python
# Ejemplo de estructura de decisión anidada en Python

# Definir la variable edad
edad = 25

# Comenzar la estructura de decisión anidada
if edad < 0:
    print("Edad inválida")
else:
    if edad < 18:
        print("Eres menor de edad.")
    else:
        if edad < 60:
            print("Eres adulto.")
        else:
            print("Eres mayor de edad.")
```

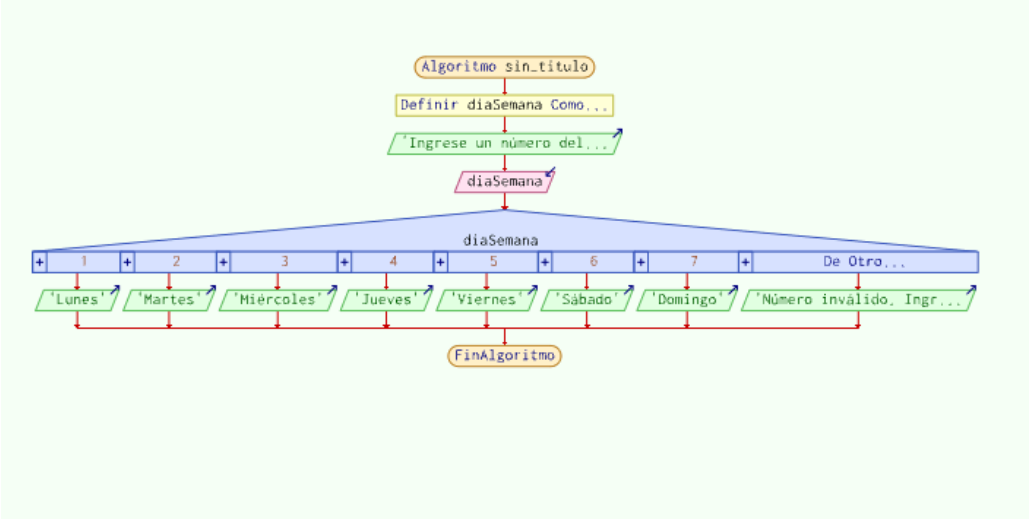

Algoritmo sin_titulo
Definir edad Como Entero
edad ← 25
edad<0
F
V
'Edad inválida'
edad<18
F
V
'Eres menor de edad.'
edad<60
F
V
'Eres mayor de edad.'
'Eres adulto.'
FinAlgoritmo



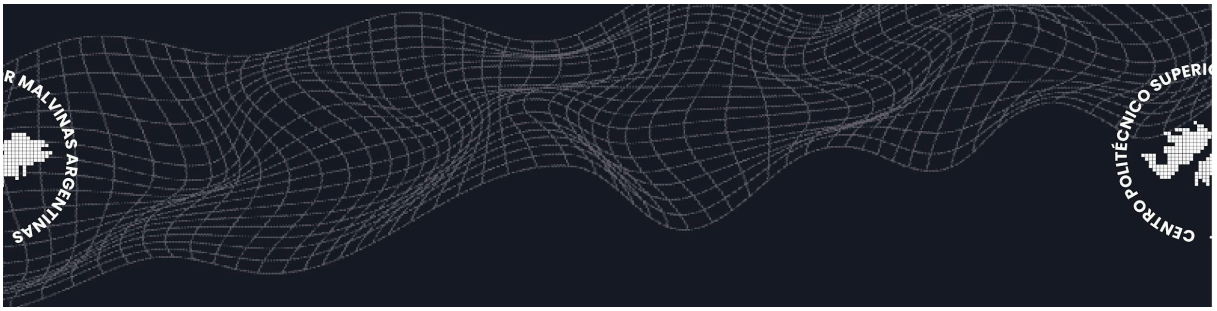
Sentencia switch

El concepto de "switch" en programación se refiere a una estructura de control que se utiliza para evaluar una expresión y tomar diferentes acciones basadas en el valor de esa expresión. En esencia, el "switch" proporciona una forma eficiente de manejar múltiples casos diferentes en un bloque de código.

Un "switch" generalmente consta de una expresión que se evalúa y una serie de casos (ramas) que corresponden a valores específicos de la expresión. Cuando la expresión se evalúa, el programa verifica qué caso coincide con el valor resultante y ejecuta el bloque de código asociado a ese caso.



En Python, no existe una sentencia switch como en algunos otros lenguajes de programación. Sin embargo, puedes lograr un comportamiento similar utilizando una serie de declaraciones if-elif-else.



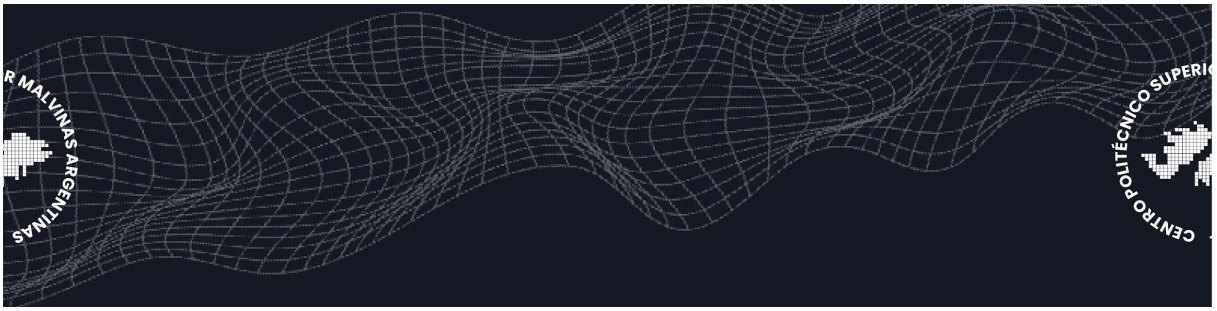
```python
dia_semana = int(input("Ingrese un número del 1 al 7 que repres

if dia_semana == 1:
    print("Lunes")
elif dia_semana == 2:
    print("Martes")
elif dia_semana == 3:
    print("Miércoles")
elif dia_semana == 4:
    print("Jueves")
elif dia_semana == 5:
    print("Viernes")
elif dia_semana == 6:
    print("Sábado")
elif dia_semana == 7:
    print("Domingo")
else:
    print("Número inválido. Ingrese un número del 1 al 7.")
```

En este caso, hemos utilizado la estructura if, elif y else en Python para lograr el mismo comportamiento que la estructura switch.

Condiciones más complejas en Python

En Python, las condiciones pueden volverse más complejas al combinar múltiples expresiones lógicas. Esto se logra utilizando operadores lógicos como "and", "or" y "not" para construir expresiones que evalúen múltiples condiciones simultáneamente. Además, podemos agrupar expresiones lógicas utilizando paréntesis para establecer el orden de evaluación.



Operador "and": Este operador devuelve True si ambas condiciones son verdaderas.

```
x = 5
y = 10
if x > 0 and y > 0:
    print("Ambas variables son positivas")
```

Operador "or": Este operador devuelve True si al menos una de las condiciones es verdadera.

```
edad = 25
if edad < 18 or edad >= 65:
    print("Tienes derecho a un descuento")
```

Operador "not": Este operador invierte el valor de la condición.

```
x = 5
if not x == 0:
    print("x no es igual a cero")
```

Agrupación de expresiones lógicas: Podemos usar paréntesis para agrupar expresiones y controlar el orden de evaluación.

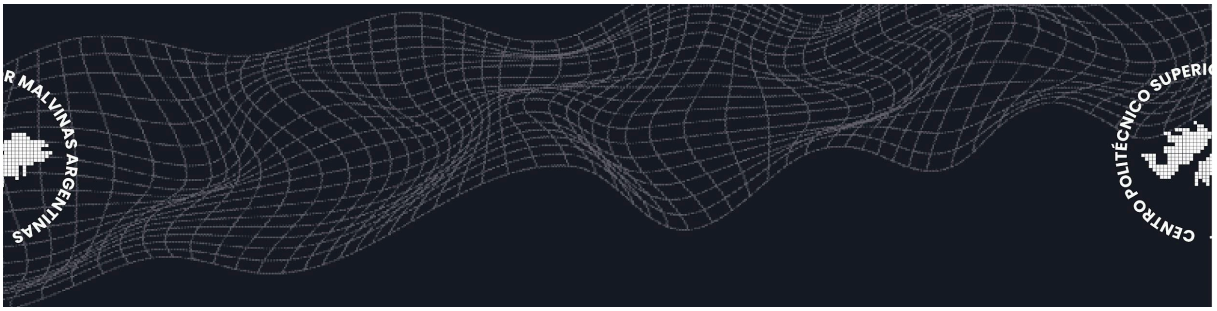
```
edad = 35
if (edad >= 18 and edad < 65) or (edad >= 65 and edad < 70):
    print("Eres elegible para votar")
```

```
numero = 75
if (numero > 0 and numero < 100) or numero < 0:
    print("El número es positivo y menor que 100 o es negativo")
```

3. Actividades

A continuación deberás utilizar el pensamiento computacional y resolver los siguientes ejercicios:

1. Verificación de edad para ver una película: Supongamos que estás desarrollando un programa para un cine y deseas asegurarte de que los espectadores sean lo suficientemente mayores para ver una película clasificada como PG-13. Debes solicitar la edad del espectador y permitir el acceso solo si tienen al menos 13 años.
2. Calificación de un estudiante: Imagina que eres un profesor y deseas calcular las calificaciones finales de tus estudiantes en función de sus puntajes en un examen. La calificación final se asignará de la siguiente manera:
   - ➔ Si el puntaje es mayor o igual a 90, la calificación es "A".
   - ➔ Si el puntaje está entre 80 y 89, la calificación es "B".
   - ➔ Si el puntaje está entre 70 y 79, la calificación es "C".
   - ➔ Si el puntaje está entre 60 y 69, la calificación es "D".
   - ➔ Si el puntaje es menor que 60, la calificación es "F".
3. Calculadora de descuento: Solicita al usuario ingresar el precio original de un producto. Luego, calcula y muestra el precio final después del descuento. Tener en cuenta que si se ingresa un precio de producto mayor o igual a $12.999 entonces se realizará el descuento del 30%, de lo contrario, se realizará el 20%.



Material de ayuda

En el siguiente video tutorial se explica las sentencias if, elif, else "ProgramacionATS":

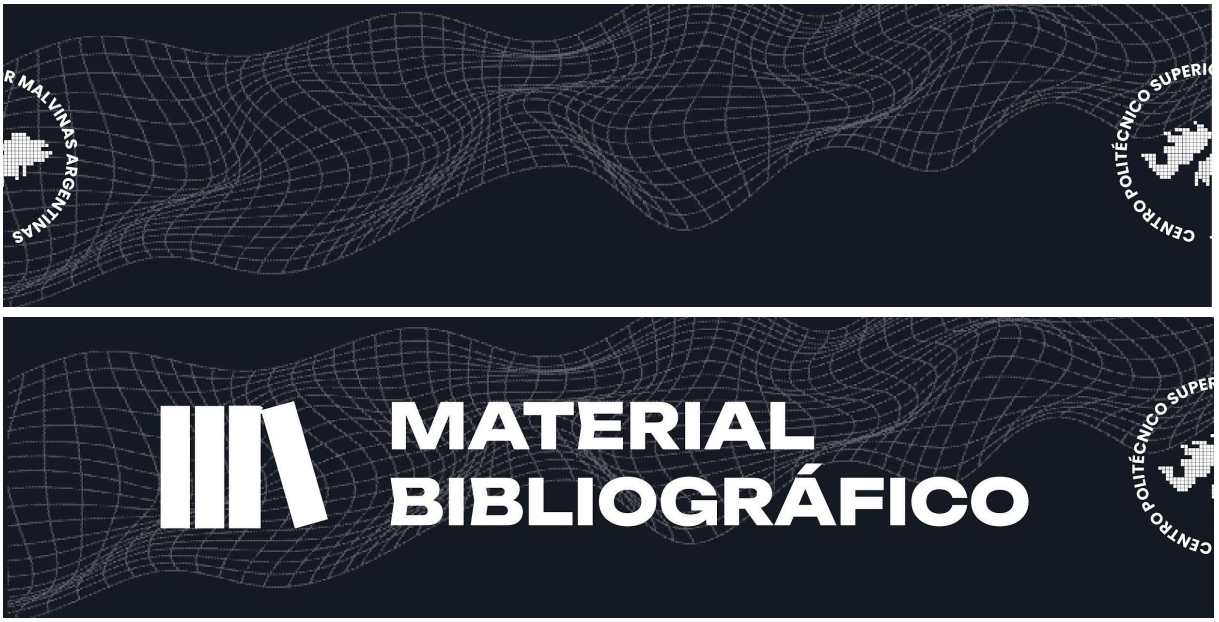
Programación en Python - Condicionales

En el siguiente video tutorial se explica cómo simular un switch con python "Carlos QL":

Match Case en Python

4. Conclusión

En este módulo hemos explorado los pilares fundamentales de la programación, centrándonos en las estructuras selectivas en Python. Desde la sencillez de la alternativa simple hasta la versatilidad de la alternativa doble y la escalabilidad de la alternativa múltiple, hemos aprendido cómo tomar decisiones críticas en nuestros programas. Además, hemos abordado las complejidades de las estructuras de decisión anidadas, que nos permiten resolver problemas más intrincados y desarrollar algoritmos más sofisticados. Al comprender y aplicar estas técnicas, hemos fortalecido nuestra capacidad para crear programas más eficientes y adaptativos. Estos conceptos son fundamentales para construir una base sólida en programación y son aplicables en una amplia variedad de proyectos y situaciones.



Bibliografía obligatoria

- FUNDAMENTOS DE PROGRAMACIÓN. *Algoritmos, estructura de datos y objetos-Cuarta edición - Luis Joyanes Aguilar- McGrawHill - 2008 - ISBN 978-84-481-6111-8*

Bibliografía sugerida de la unidad

- *Python 3 – Los fundamentos del lenguaje - Es un libro completo, bien escrito, claro, aunque un poco extenso, por lo que no es apto para lectores apurados. La intención del autor es brindar absolutamente todo lo que se necesita para que el lector aprenda a programar en Python desde cero.*
- *Python para todos: explorando datos en Python 3 - Este libro es ideal para estudiantes y programadores que buscan especializarse en la exploración y el análisis de datos.*