

Tipificación de Condicionales Clasificación General





Decisiones - Condición

- Escogencia de ***uno entre varios caminos lógicos*** dependientes todos de una condición.
- Algunos autores acotan esta definición en que se escoge ***uno entre dos caminos lógicos*** y dado que todo el proceso del computador es binario podemos decir que esos libros también :enen la razón.





Estructura: Si-entonces-Si no

- Se sugieren las siguientes formas de clasificar los condicionales:
 - Decisiones simples
 - Decisiones en secuencia
 - Decisiones en cascada
 - Decisiones anidadas
- Más condicionales, implica que el algoritmo está abarcando un mayor número de casos posibles del problema que resuelve: mejor comportamiento



Decisiones Simples

Si (Condición)

.

.

Instrucciones a ejecutar

En caso de que la condición sea Verdadera

.

.

Sino

.

.

Instrucciones a ejecutar

En caso de que la condición sea Falsa

.

.

Fin_Si

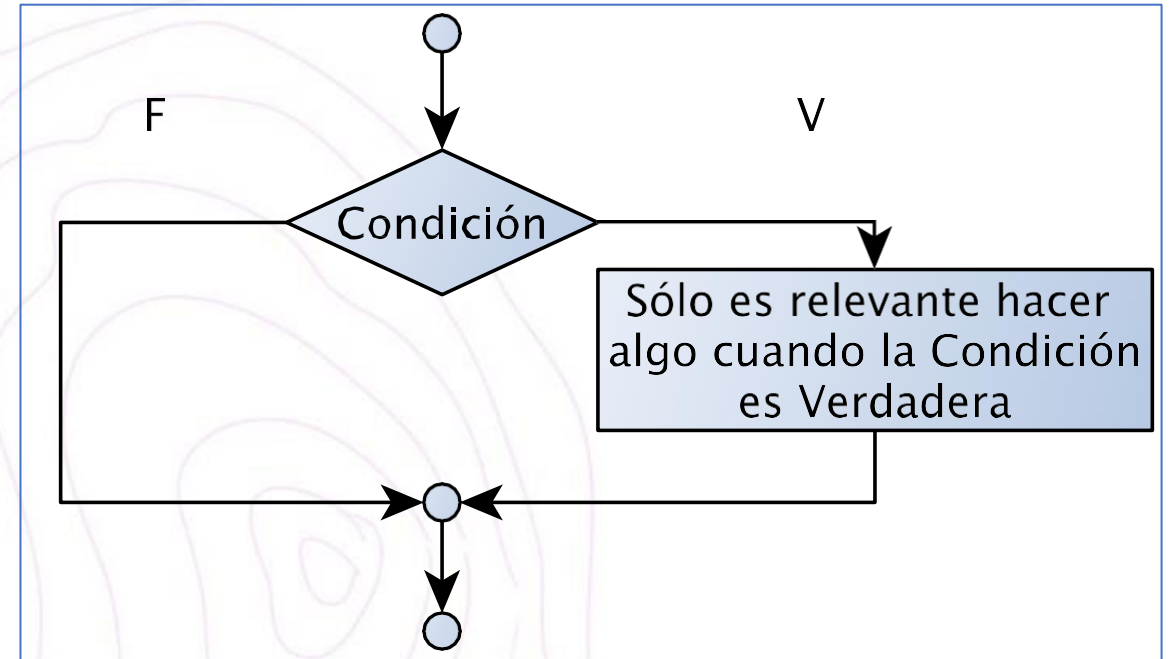
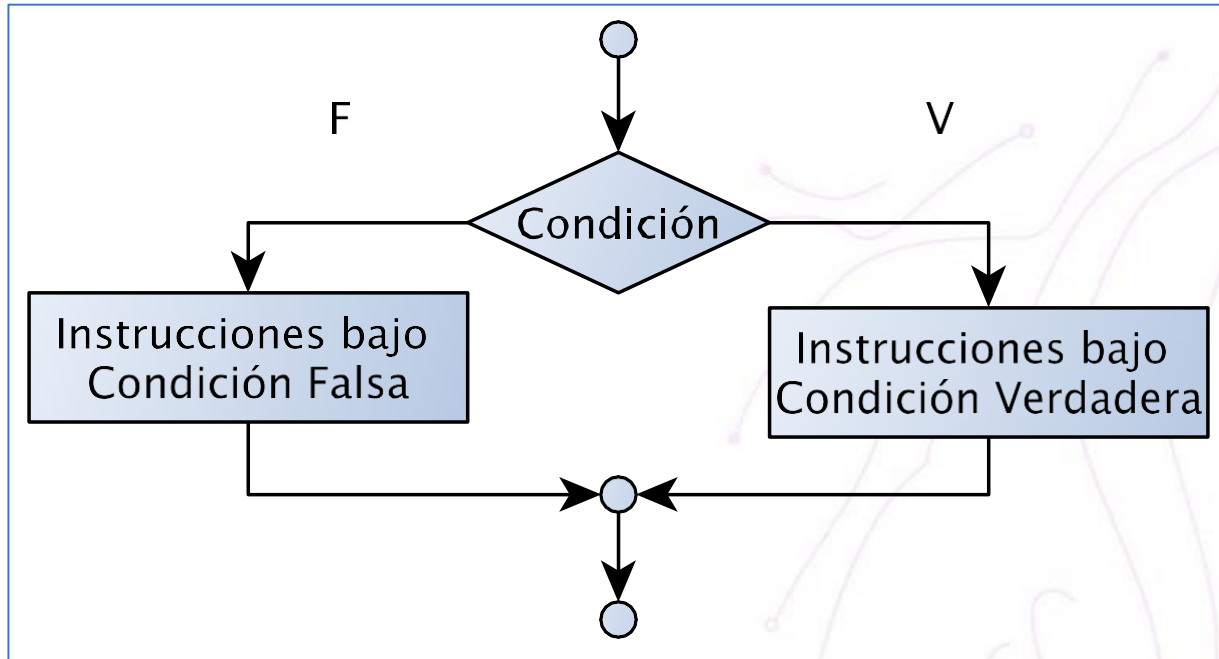


Decisiones Simples

- **No es obligatorio** que un condicional *Si*, tenga que incluir una alterna:va *Si no*, dado que no siempre es importante generar una determinada acción en el caso de que la condición sea *Falsa*.
- Es importante delimitar hasta donde llega toda la estructura de decisión y esa función la cumple el *Fin_Si* que aparece al final de ella.
- *Recordar* que en una estructura de decisión cuando se realizan las instrucciones por la parte *Verdadera* no se hacen las instrucciones por la parte *Falsa* y viceversa.



Decisiones Simples





Decisiones Simples





Decisiones Simples

¿Cuándo utilizar el condicional con sección falsa?

- Se utiliza un Si-entonces-Si no, si las dos condiciones son **mutuamente excluyentes**.
- En otras palabras: si la condición es ***verdadera***, la otra condición ***debe ser falsa***.



Decisiones Simples

Ejemplo: Antes del mediodía (AM) y después del mediodía (PM) son mutuamente excluyentes.

```
Si (horaDelDia >=12) entonces  
    →Escriba "PM";  
Si no // Debe ser AM  
    →Escriba "AM";  
Fin_Si
```



Decisiones Simples

¿Cuándo utilizar el condicional sin sección falsa?

- Cuando ***ambas*** condiciones pueden ser ***ciertas al mismo tiempo***.





Decisiones Simples

Ejemplo: Ambas condiciones son verdaderas, porque, se puede aprobar y al mismo tiempo hacer un buen trabajo.

```
1 Si resultadoDelExamen > 60
2   Escriba "Aprobado";
3
4 Si resultadoDelExamen > 90
5   Escriba "Excelente trabajo!";
```



Decisiones Simples - Secuencia

- Este es el tipo de estructura que se utiliza cuando se deben realizar varias preguntas en donde no es importante (normalmente) el *Si no* de cada decisión.
- Es decir, o no es relevante la parte falsa de la estructura de decisión, o se puede generalizar cuando no se cumple ninguno de los condicionales de la secuencia.



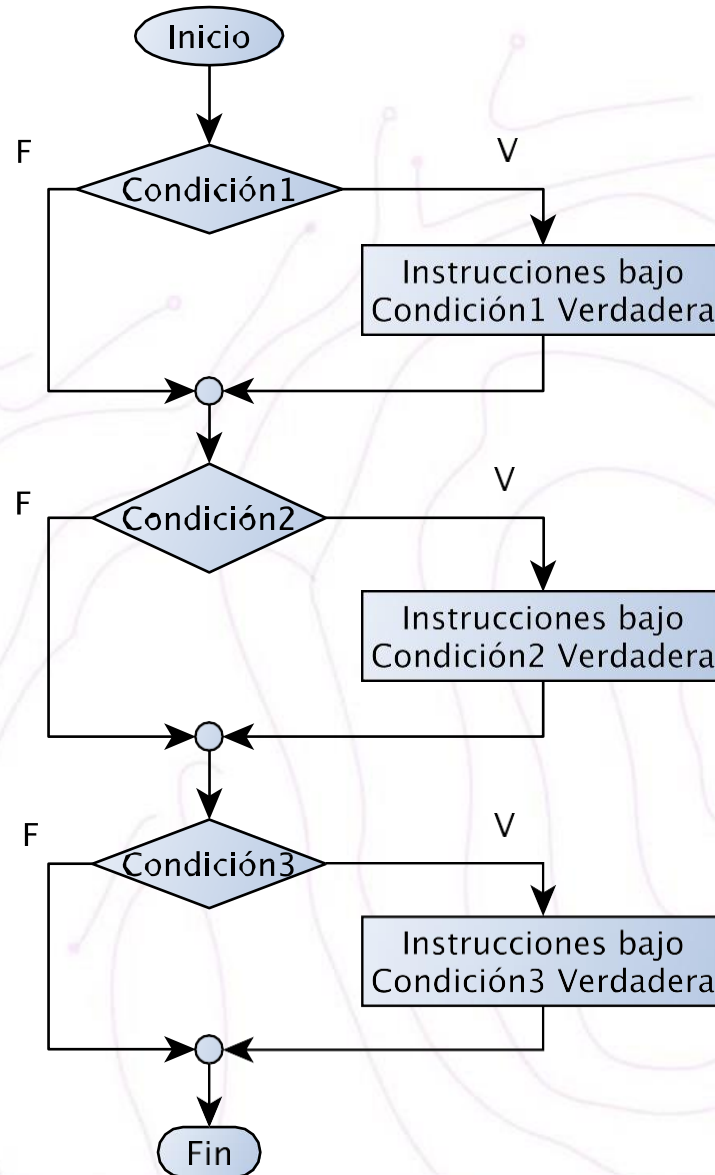


Decisiones Simples - Secuencia

```
1 Si Condición 1
2     Instrucciones a ejecutar en caso de
3     que la Condición 1 sea ///Verdadera
4 Si Condición 2
5     Instrucciones a ejecutar en caso de
6     que la Condición 2 sea ///Verdadera
7 Si Condición 3
8     Instrucciones a ejecutar en caso de
9     que la Condición 3 sea ///Verdadera
10 Si Condición 4
11     Instrucciones a ejecutar en caso de
12     que la Condición 4 sea ///Verdadera
```




Decisiones Simples - Secuencia





Decisiones Simples - Secuencia

Ejemplo:

- Leer un número entero y determinar si es positivo o negativo o si es 0.
- En este, el estado de la variable hace que las posibilidades sean naturalmente excluyentes, se puede enmarcar en una secuencia de condicionales.



Decisiones Simples - Secuencia

Programa Decisiones_en_Secuencia

Variables

Entero : num

Inicio

Escriba "Digite un número entero"

Lea num

Si num < 0

Escriba "El número digitado es negativo"

Si num > 0

Escriba "El número digitado es positivo"

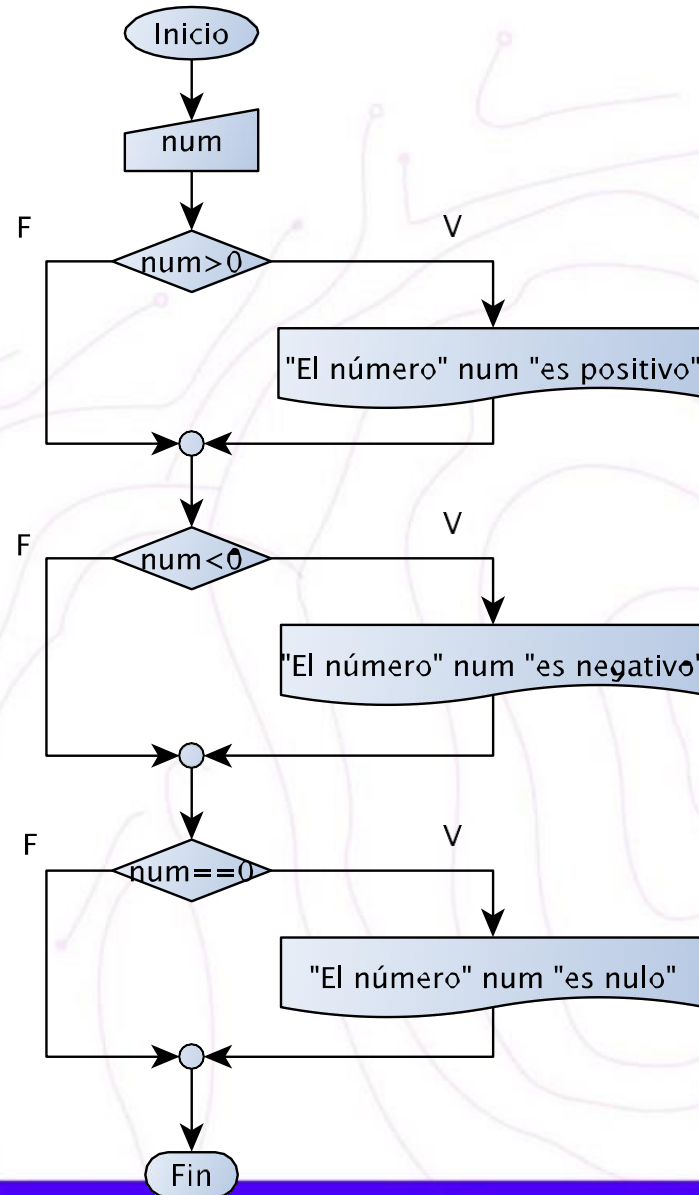
Si num = 0

Escriba "El número digitado es cero"

Fin



Decisiones Simples - Secuencia





Decisiones Simples - Secuencia

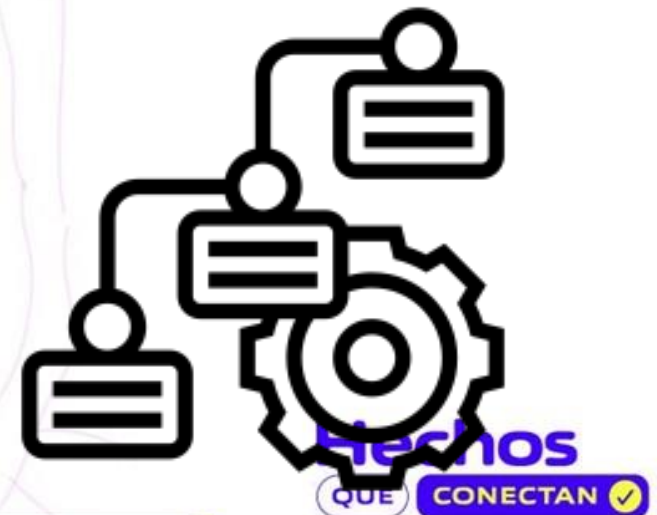


```
1 num = int(input())
2 if num > 0:
3     print("El número ", num, "es positivo")
4 if num < 0:
5     print("El número ", num, "es negativo")
6 if num == 0:
7     print("El número ", num, "es nulo")
```




Decisiones En Cascada

- Este no es mas que un esquema en donde el *Sino* de cada *Si* condicional da inicio a un nuevo *Si* condicional y así sucesivamente.
- Esquema de decisión bastante utilizado para establecer los caminos alternos de una solución.



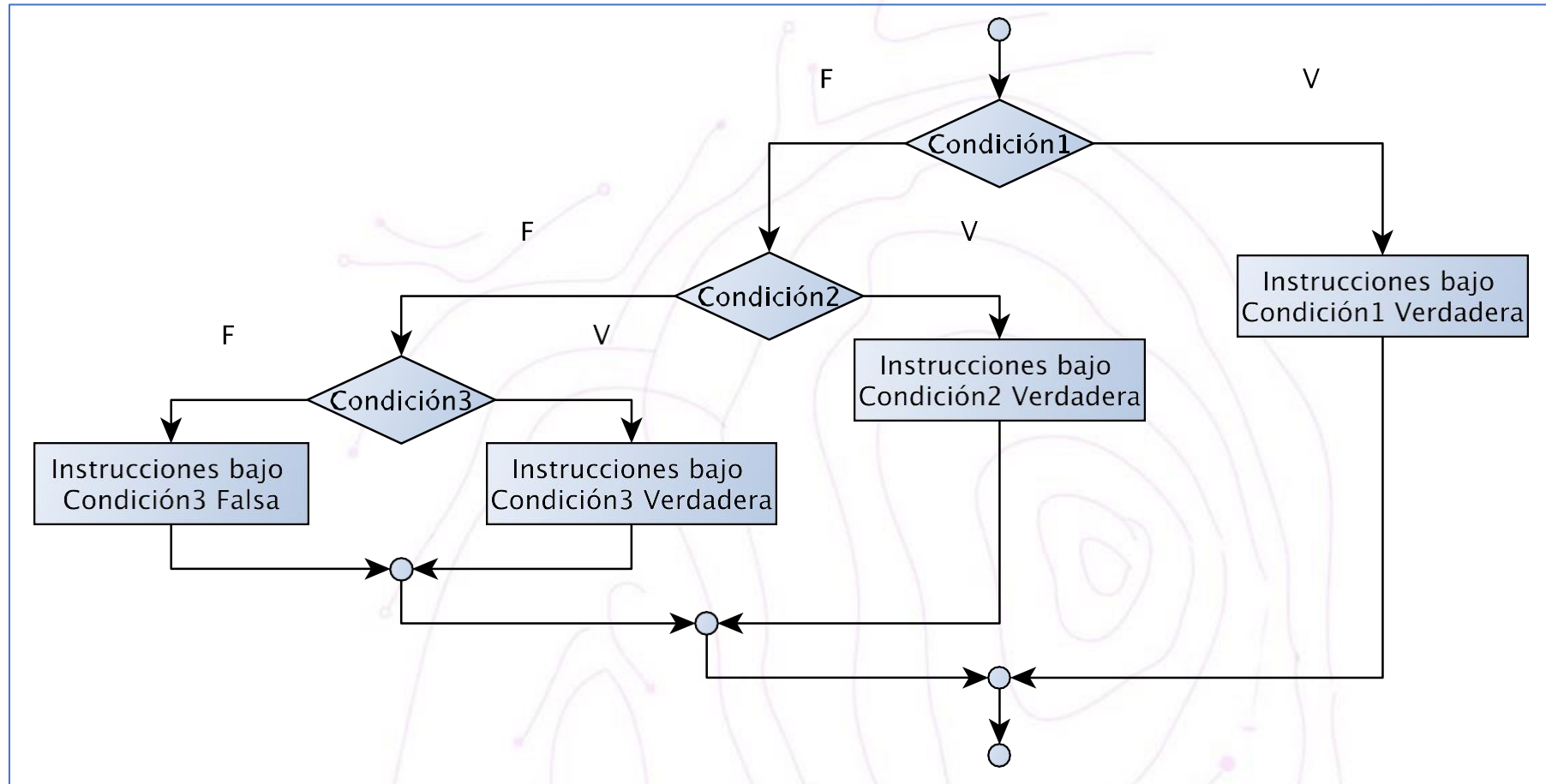


Decisiones En Cascada

```
1 Si Condición 1
2     Instrucciones a ejecutar en caso de
3     que la Condición 1 sea Verdadera
4 Sino
5     Si Condición 2
6         Instrucciones a ejecutar en caso de
7         que la Condición 2 sea Verdadera
8     Sino
9         Si Condición 3
10            Instrucciones a ejecutar en caso de
11            que la Condición 3 sea Verdadera
12        Sino
13            Instrucciones a ejecutar en caso de
14            que la Condición 3 sea Verdadera
15        Fin_Si
16    Fin_Si
17 Fin_Si
```



Decisiones En Cascada





Decisiones En Cascada

Ejemplo: Plantear un algoritmo que alcance los siguientes objetivos:

- Leer un número entero y determinar si es de uno o dos o tres o cuatro dígitos.
- Validar que el número no sea negativo.



Decisiones En Cascada

```
1 //Algoritmo Decisión_en_Cascada
2 //Variables
3 Entero : num
4 Inicio
5 Escriba "Por favor, digite un número entero"
6 Lea num
7 Si num < 0
8     num = num * ( -1 )
9 Fin_Si
10 Si num >= 1 y num <= 9
11     Escriba " El número tiene 1 dígito "
12 Sino
13     Si num >= 10 y num <= 99
14         Escriba "El número tiene 2 dígitos"
15     Sino
16         Si num >= 100 y num <= 999
17             Escriba "El número tiene 3 dígitos"
18         Sino
19             Si num >= 1000 y num <= 9999
20                 Escriba "El número tiene 4 dígitos"
21             Sino
22                 Escriba "El número tiene mas de 4 dígitos"
23             Fin_Si
24         Fin_Si
25     Fin_Si
26 Fin_Si
27 Fin
```




Decisiones En Cascada

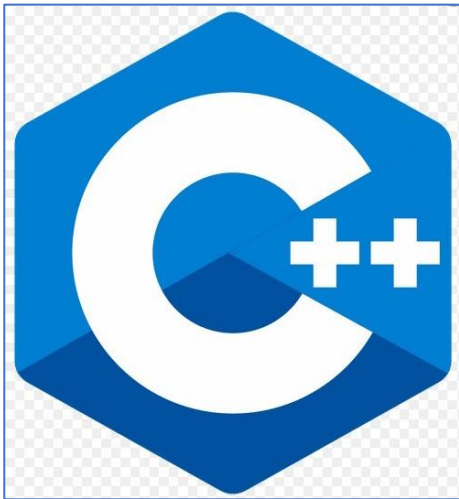
```
1 num = int(input("Por favor, digite un número entero: "))
2 if num < 0:
3     num = num * (-1)
4 if num >= 1 and num <= 9:
5     print("El número tiene 1 dígito")
6 else:
7     if num >= 10 and num <= 99:
8         print("El número tiene 2 dígitos")
9     else:
10        if num >= 100 and num <= 999:
11            print("El número tiene 3 dígitos")
12        else:
13            if num >= 1000 and num <= 9999:
14                print("El número tiene 4 dígitos")
15            else:
16                print("El número tiene más de 4 dígitos")
```





Decisiones En Cascada

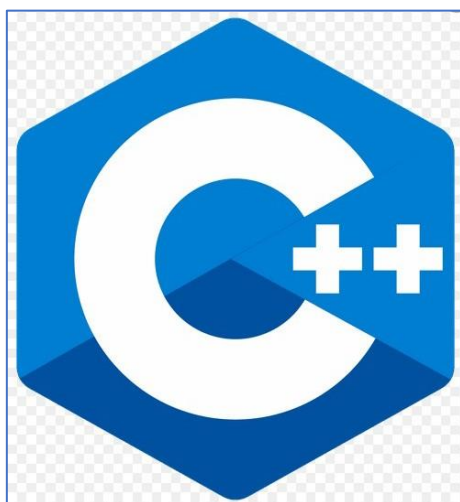
“else if”



```
1  if (condition1)
2  {
3      // do this if condition1 is true
4      // condition 1 statements
5      // then exit if/else if
6  }
7  else if (condition2)
8  {
9      // do this if condition2 is true
10     // condition 2 statements
11     // then exit if/else if
12 }
13 else if (condition3)
14 {
15     // do this if condition3 is true
16     // condition3 statements
17     // then exit if/else if
18 }
19 // continuation point after if/else if is complete
```



Decisiones En Cascada “else if”



```
if (condition1)
{
    // do this if condition is true
    // condition1 statements
    // then exit if/else if
}
else if (condition2)
{
    // do this if condition2 is true
    // condition2 statements
    // then exit if/else if
}
else if (condition3)
{
    // do this if condition3 is true
    // condition3 statements
    // then exit if/else if
}
else // if no if condition was true
{
    // do this if no if conditions were true
    // trailing else statements
}
```

Only executed if condition1,
condition2, and condition3
are all false



Decisiones En Cascada “elif”

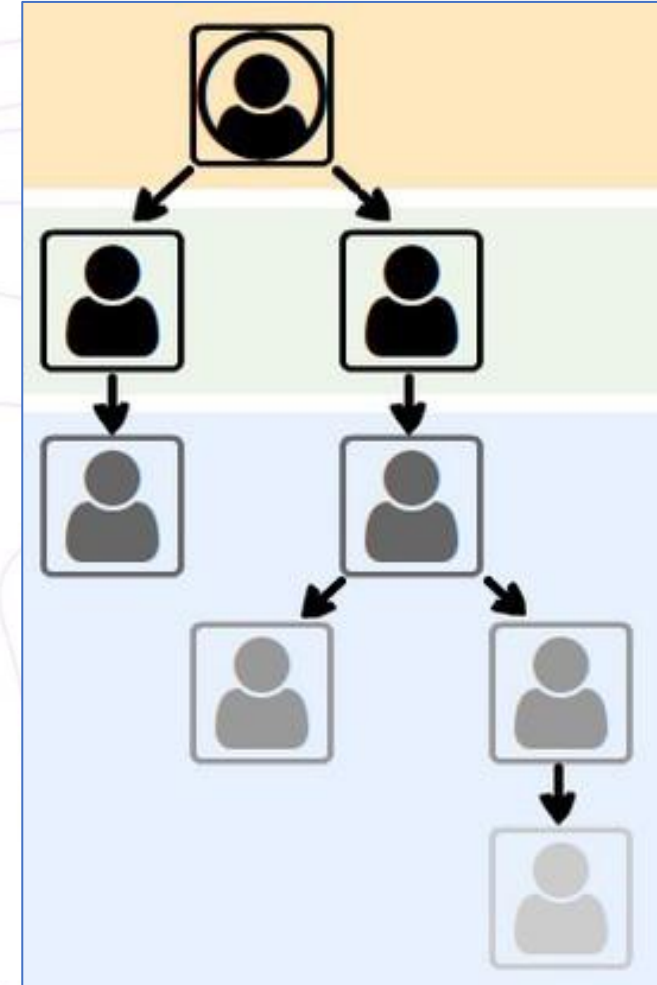
```
1 num = int(input("Por favor, digite un número entero: "))
2 if num < 0:
3     num = num * (-1)
4 if num >= 1 and num <= 9:
5     print("El número tiene 1 dígito")
6 elif num >= 10 and num <= 99:
7     print("El número tiene 2 dígitos")
8 elif num >= 100 and num <= 999:
9     print("El número tiene 3 dígitos")
10 elif num >= 1000 and num <= 9999:
11     print("El número tiene 4 dígitos")
12 else:
13     print("El número tiene más de 4 dígitos")
```





Decisiones Anidadas

- Éstas se presentan en aquellos casos en los cuales una estructura completa de decisión se encuentra dentro de otra.





Decisiones Anidadas

Ejemplo:

- Leer un número entero y determinar si es posi%vo o nega%vo.
- Si es posi%vo determinar si %ene dos dígitos y si es nega%vo determinar si %ene tres dígitos.
- Asumir que no puede entrar el número cero.



Decisiones Anidadas

Programa Decisiones_Anidadas

Variables

Entero : n

Inicio

Escriba "Digite un número entero"

Lea num

Si num > 0

Si num >= 10 y num <= 99

Escriba "El número es positivo y tiene dos dígitos"

Sino

Escriba "El número es positivo y no tiene dos dígitos"

Fin_Si

Sino

Si num >= - 999 y num <= -100

Escriba "El número es negativo y tiene tres dígitos"

Sino

Escriba "El número es negativo y no tiene tres dígitos"

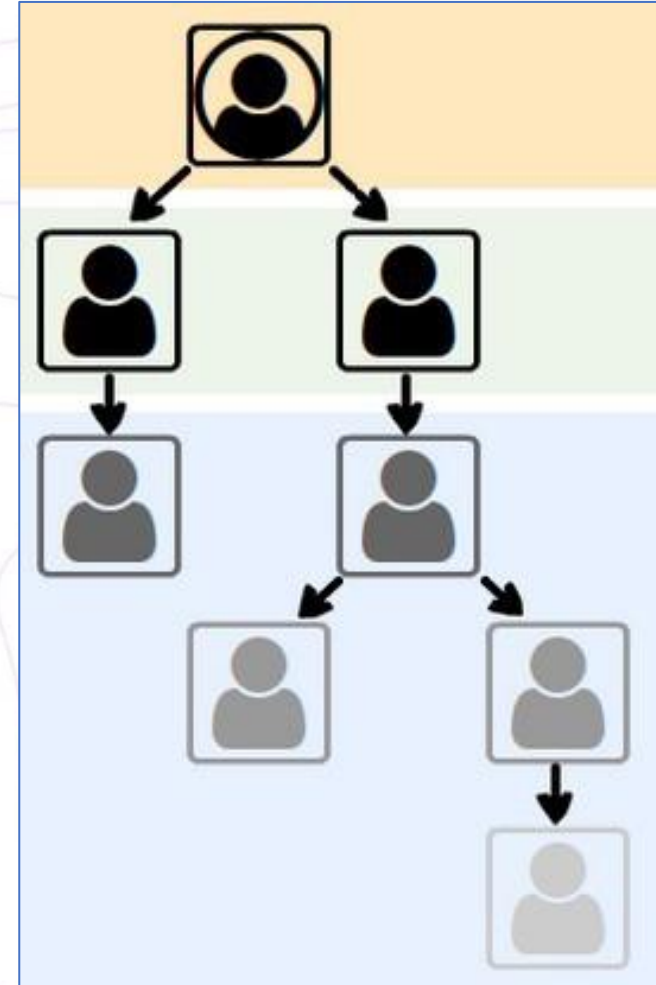
Fin_Si

Fin_Si



Decisiones Anidadas

- **Ejercicio:** Desarrollar el ejemplo en Python.





Estructura de Casos

Actividad:

- Consultar la estructura de casos o “switch” como aparece en algunos lenguajes de programación.
- ¿Estas estructuras se pueden anidar?
- ¿Se pueden mezclar con las estructuras de los condicionales?

