

Combinando condicionales

Ahora que ya conocemos las condiciones y los operadores, podemos empezar a combinarlos!

Swift tiene dos operadores que nos permiten combinar condiciones y verificar su valor de verdad. Estas son "and" y "or"

And:



El conector "and" unifica dos condiciones y solo ejecuta el código si **ambos son verdaderos**.
Por ejemplo, si edadUno es mayor a 18 y edadDos es mayor a 18, entonces ejecuto el código.

Estructura

```
if condicionUno && condicionDos {  
    //Se ejecuta el codigo si ambas son verdaderas  
} else {  
    //Se ejecuta este si alguna de las dos es falsa
```

Veamos un ejemplo:

Como la edadUno es falsa, Swift directamente ni siquiera va a revisar la segunda condición: **Con que la primera sea falsa, el código no se ejecuta.**

```
edadUno = 15  
edadDos = 19  
  
if edadUno >= 18 && edadDos >= 18 {  
    print("Son mayores de edad")  
} else {  
    print("Hay al menos un menor de edad")  
}
```

Or



Nuestro otro conector es "or" el cual permite que se ejecuta el código cuando al menos uno de los dos es verdadero.

Estructura:

```
if condicionUno || condicionDos {  
    //Se ejecuta el codigo si alguna de las dos es verdadera  
} else {  
    // se ejecuta el codigo si las dos son falsas.  
}
```

Ejemplo

```
edadUno = 15  
edadDos = 19  
  
if edadUno >= 18 || edadDos >= 18 {
```

```
    print("Hay al menos un mayor de edad")
}
```

Switch

Okey, pero.. Que pasa cuando tenemos un if que tiene muchos casos?

Bueno, Swift tiene otra herramienta llamada `Switch`.

Veamos como funciona y en que casos podemos intercambiarla por el if!

Switch toma un valor y lo compara con las posibles alternativas que declaramos, denominados "cases".



Swift va a ejecutar el bloque de código correspondiente al caso con el que coincida la información recibida.

Estructura:



Primero utilizamos la palabra reservada `Switch` y entre llaves declaramos los casos con la palabra "case", el nombre del caso y el código que se debe ejecutar si se cumple.

```
switch variable {
    case condicionUno:
        //Se ejecuta este bloque de código
    case condicionDos:
        //Se ejecuta este bloque de código
    case condicionTres:
        //Se ejecuta este bloque de código
```

Algo importante sobre los Switch es que deben ser exhaustivos!

A que me refiero? Tiene que haber una `cantidad limitada de casos` y el Switch los debe cubrir todos.

Es por eso que al final de los casos, agregamos el caso "`Default`"



El default se va a ejecutar en caso de que la información recibida no coincida con ninguno de los casos.

Ejemplo:

```
albumsAvicii = "True"

switch AlbumsAvicii {
    case "True":
        print("Ese es el primer album de avicii")
    case "Stories":
        print("Ese es el segundo album de avicii")
    case "AVICI (01)":
        print("Ese es el tercer album de avicii")
    case "Tim":
        print("Ese es el ultimo album de avicii")
    default:
        print("mmm... ese album no es de avicii")
```

Entonces... Cuando Switch y cuando if?

Destaquemos dos motivos claves:

1. Para usar Switch, tenemos que estar seguros que cubrimos todos los casos posibles. Es muy útil en los casos que necesitamos hacer una comparación entre una cantidad definida de respuestas.
2. El otro motivo es que si tenemos muchas condiciones, por una cuestión de legibilidad probablemente será más claro utilizar un Switch antes que un if.