

# Tipos opcionales

Es común que al abrir una aplicación y presionar algunos botones, la aplicación se cierra súbitamente. ¿Porqué ocurre esto?. Una causa común es que una aplicación intente leer una variable que no tiene un valor en el tiempo de ejecución, y aquí ocurre el error en la aplicación.

La introducción de los opcionales es la forma en que Swift ayuda a los desarrolladores a escribir mejor código y prevenir este tipo de errores.

Puede parecer un concepto extraño pero la idea es bastante simple. Antes de tener acceso a una variable que podría no tener valor, Swift alienta a verificarlo primero. Se debe estar seguro de que la variable tiene un valor antes de cualquier procesamiento, esto puede prevenir un cierre inesperado de la aplicación.

Cuando se declara una variable Swift necesita que tenga un valor, de esta forma una variable no opcional está garantizado que tiene un valor. Si se intenta declarar una variable sin un valor, el compilador marca un error.

```
var variable:String  
  
print(variable)
```

En algunas situaciones se debe declarar una variable sin un valor inicial, por ejemplo en un formulario. No todos los campos son imperativos, algunos campos pueden ser opcionales por lo que podrían no tener un valor.

Técnicamente, opcional es un tipo de dato en Swift e indica que la variable en cuestión podría tener o no un valor. Para declarar una variable como opcional se le debe agregar un signo de interrogación a la declaración.

Este código ya no marcará un error, pero preste atención a la impresión del **print**.

```
var variable:String?  
  
print(variable)
```

En el código, se ha declarado una variable de tipo **String** y que es **opcional**, por esa razón es que el compilador no marca un error. A diferencia de las variables no opcionales, se debe indicar explícitamente el tipo de dato y no será posible que el compilador deduzca el tipo adecuado de acuerdo al primer valor que se almacene en la variable.

En la salida del código se obtiene el mensaje **nil**, lo que significa que la variable no tiene un valor. Se puede asignar un valor a la variable de la manera habitual. Al agregar una asignación al código de ejemplo, la salida imprime el mensaje pero indica algunos mensajes de advertencia.

```
var variable:String?  
  
variable = "Hola mundo"  
  
let mensaje = "El mensaje es: " + variable  
  
print(mensaje)
```

En los mensajes de advertencia swift indica que la variable opcional no tiene un valor por defecto, lo que podría ocasionar un problema en la aplicación.

## Forzar desenvolvimiento

Con las variables opcionales es necesario un mecanismo para verificar y desenvolver el valor, swift ofrece diversas formas.

Se puede utilizar un **if** para verificar si la variable opcional tiene un valor comparandola con **nil**, si se determina que la variable tiene un valor entonces se desenvuelve.

```
var variable:String?  
  
variable = "Hola mundo"  
  
if variable != nil {  
    let mensaje = "El mensaje es: " + variable!  
    print(mensaje)  
}
```

Si la variable no es igual a **nil** entonces debe tener un valor, por lo que se pueden ejecutar sentencias dentro del if que requieran el valor de la variable. Se debe agregar el símbolo **!** a la variable opcional para indicarle a swift que dicha variable si tiene un valor y es seguro utilizarla.

## Enlace opcional

Este es el mecanismo más recomendado para el manejo de las variables opcionales, y no requiere el uso del símbolo **!**.

El código con enlace opcional queda de la siguiente forma:

```
var variable:String?  
  
variable = "Hola mundo"  
  
if let valorVariable = variable {  
    let mensaje = "El mensaje es: " + valorVariable  
    print(mensaje)  
}
```

Se utiliza `if let` para determinar si la variable tiene un valor, y en caso afirmativo se asigna a una constante temporal. Como puede ver, no fue necesario el sufijo `!`.

Es posible utilizar el mismo nombre para la constante temporal y la variable opcional.

```
var variable:String?  
  
variable = "Hola mundo"  
  
if let variable = variable {  
    let mensaje = "El mensaje es: " + variable  
    print(mensaje)  
}
```

**Nota:** A pesar de que la variable opcional y la constante temporal tienen el mismo nombre, son distintas entre sí.