VARIABLES OPCIONALES

Desarrollo de Aplicaciones Móviles

OPCIONALES

- Una de las cosas mas importantes de Swift, es que no admite una variable o constante sin valor.
- En el lenguaje **no existe el valor nulo, vacío, nil o null** que puede existir en otros lenguajes.
- No es un valor válido… de hecho no es un valor.

- ¿Qué hacemos entonces si queremos declarar una variable cuyo valor aún desconocemos?
- ¿Qué pasa si queremos tener una variable que sí nos interese en un momento determinado que pueda estar vacía?

- Swift incorpora un modificador para las variables, esencial para nuestro trabajo: **las opcionales.**

- **Un valor opcional es un contenedor que puede almacenar solo dos posibles cosas: nada o algo.**

- Mientras que una variable o constante en Swift solo puede almacenar algo, pero nunca nada.
- **El opcional es una especie de referencia y no un contenedor en sí. Es un contenedor dentro de otro.**

VARIABLES

- En Swift, una variable que no contiene ningún tipo de puntuación garantiza que contiene un valor de ese tipo.

- Ejemplo:

```
import UIKit

var variable1 = "Hola"
var variable2: String = "Hola"
var variable3: String
variable3
```

```
"Hola"
"Hola"
```

```
Playground execution failed: error: VariablesOpcionales.playground:2:1: error: variable 'variable3' used before being initialized
variable3
^

VariablesOpcionales.playground:1:5: note: variable defined here
var variable3: String
    ^
```

➤ No es posible utilizar el valor **nil** con variables o constantes no opcionales.

```
import UIKit

var variable1 = "Hola"
variable1 = nil
let constante1: String
constante1 = nil
```

OPCIONALES EN SWIFT

- Las variables opcionales en Swift permiten indicar la ausencia de un valor para cualquier tipo de dato.

- El símbolo de interrogación **?** indica que el valor que puede contener la variable/constante es opcional.

- Se puede establecer una variable opcional a un estado sin valor mediante el valor especial **nil**.

EJEMPLO

```swift
import UIKit

var variable1 = "Hola"
var variable2: String = "Hola"
var variable3: String?
variable3
```

```
"Hola"
"Hola"
nil
nil
```

A variable of type
`String?`

will either
contain a `String` value
or be empty (`nil`).

OBTENIENDO EL VALOR DE VARIABLES OPCIONALES

- Para obtener el valor de una variable opcional, se debe realizar un proceso llamado desempaquetamiento (unwrapped), usando el operador **?** o el operador **!** a la derecha de la variable.

USANDO ?

- El operador ? indica al compilador que intente desempaquetar la variable y de no poder no continúe la operación con la variable opcional, asignando nil al resultado requerido.

```
import UIKit

var variable: String?
variable?.endIndex
```

```
nil
nil
```

```
import UIKit

var variable: String?
variable = "hola mundo"
variable?.endIndex
```

```
nil
"hola mundo"
10
```

USANDO !

- El operador ! permite desempaquetar el valor de un opcional, el valor debe existir y en caso de no existir provocará un error, el uso de este operador se debe hacer solo en el caso de estar seguros que el opcional contenga un valor.

```
import UIKit

var variable: String?
variable!.endIndex
```

error: Execution was interrupted, reason: EXC_BAD_INSTRUCTION (code=EXC_I38...

error

```
import UIKit

var variable: String?
variable = "Hola mundo"
variable!.endIndex
```

```
nil
"Hola mundo"
10
```

OTRO EJEMPLO

```
import UIKit

var strValor1:String?
strValor1 = "Hola"
if (strValor1 != nil)
{
    print("Letras \(strValor1!.endIndex)")
    print("Mayusculas \(strValor1!.uppercased())")
}
else
{
    print("La cadena no ha sido inicializada")
}
```

- Swift propone la estructura **if let,** para hacer el código un poco mas legible.

```
var strValor1:String?
strValor1 = "Hola"


if let valorDesempaquetado = strValor1
{
    print("Letras \(strValor1!.endIndex)")
    print("Mayusculas \(strValor1!.uppercased())")
}
else
{
    print("La cadena no ha sido inicializada")
}
```

OPCIONALES IMPLÍCITAMENTE DESEMPAQUETADOS

- Las variables pueden ser nulas o tener un valor, pero no necesitan ser desempaquetadas en ningún caso, se usan sin los operadores **?** o **!**, pero si se intenta usar una variable con un valor nulo se genera un error.

```
import UIKit

var srtValor:String!
srtValor="Hola Mundo"
srtValor.endIndex

srtValor = nil
srtValor.endIndex                                                    error
error: Execution was interrupted, reason: EXC_BAD_INSTRUCTION (code=EXC_I386_INVOP, subcode=0x0).
```

PRÁCTICA 3

- Prueba el funcionamiento de los siguientes códigos, explica el funcionamiento de cada código y envíalo por correo en formato .pdf.

Ejercicio 1:

```
import UIKit

var cajaCebollas: Int?

if cajaCebollas != nil{
    var cantidadCebollas = cajaCebollas!
    print("Tengo \(cantidadCebollas) cebollas en la caja")
}
else{
    print("No hay cebollas dentro de la caja")
}
```

Ejercicio 2:

```
import UIKit

var cajaCebollas: Int?

cajaCebollas = 10
if cajaCebollas != nil{
    var cantidadCebollas = cajaCebollas!
    print("Tengo \(cantidadCebollas) cebollas en la caja")
}
else{
    print("No hay cebollas dentro de la caja")
}
```

Ejercicio 3:

```
var cajaCebollas: Int?

if let cantidadCebollas = cajaCebollas {
    print("Tengo \(cantidadCebollas) cebollas")
}
else{
    print("No hay cebollas en la caja");
}
```

Ejercicio 4:

```
var cajaCebollas: Int?

cajaCebollas = 10

if let cantidadCebollas = cajaCebollas {
    print("Tengo \(cantidadCebollas) cebollas")
}
else{
    print("No hay cebollas en la caja");
}
```