

6.- Explicar el ciclo de vida de View Controller

1.- viewDidLoad: Método que se ejecuta primero en la vista, solo se instancia una vez, se ejecuta una vez que la vista se carga a la memoria.

2.- viewWillAppear: Esta por mostrarse la vista, eso se utiliza comúnmente para actualizar datos.

3.- viewDidAppear: Se muestra la vista. Usado para procesos que llevarían más tiempo en ejecutarse como los consumos de servicios web.

4.- viewWillDisappear: Punto de desaparecer. Se ejecuta como consecuencia a la eliminación de la vista, me manda a llamar antes de que la vista se elimine por completo.

5.- viewDidDisappear: Desaparece la vista. Este método se ejecuta cuando la vista es destruida/ eliminada por completo.

7.- Explicar el ciclo de vida de la aplicación

Esto se puede decir que se compone en tres partes, **Inicialización, Funcionamiento y Finalización.**

Dando como inicio cuando se abre(dando tap) a la aplicación por primera vez esto a su vez ejecuta el método Main el cual lanza la clase UIApplicationMain encargada de cargar todo lo necesario para el correcto funcionamiento de la aplicación. El método application:willFinishLaunchingWithOptions del AppDelegate se encarga de inicializar los framework entre otras cosas, posteriormente se muestra el Launch Screen el cual es utilizado para mostrar el logo de la empresa por medio de una imagen y en algunos casos consumir algunos servicios web iniciales/validaciones de la app. Una vez finalizando esto se abre el view controles indicado a su vez iniciando el ciclo de vida del view controller. Esto se hace las veces que sean invitadas las vistas.

Durante la navegación del usuario dentro de la App puede decidir Cerrar/Matar o pasar a segundo plano de la aplicación, dando esto como consecuente el uso de los métodos applicationWillResignActive esto indica que la aplicación se paso a segundo plano y applicationDidEnterBackground esto indica que la aplicación regreso del segundo plano

8.- En qué casos se usa un weak, un strong y un unowned.

Weak siempre es de tipo opcional por lo cual se tiene que comprobar por lo contrario el compilador no deja avanzar.

Unowned siempre tiene un valor de lo contrario si se fuerza y el no tiene ningún valor va a crashear la app.

9.- ¿Que es un ARC?

Automatic Reference Counting (Contador de referencias automático), es una funcionalidad que permite liberar la memoria de aquellos elementos que no posean referencias fuertes hacia ellos (strong references). Es como un Garbage Collector pero con un mayor rendimiento todo con los problema relacionados con la memoria.

10.- Punto extra.

En primera instancia se ejecuta el método dentro **didFinishLaunchingWithOptions** del ciclo de vida de la aplicación, esto dentro del **AppDelegate** o **SceneDelegate** el cual pinta de color amarillo el fondo de la vista.

Posteriormente se crea la vista el cual ocupa un lugar en la memoria e inicia su ciclo de vida del View controller , donde se cambia de color rojo el fondo de la vista dentro del método **viewDidLoad**.

Como conclusión podemos decir que la vista queda de color rojo ya que la **didFinishLaunchingWithOptions** se ejecuta cuando se abre la app mientras que el **viewDidLoad** se ejecuta cuando abre la pantalla a mostrar dando este como ultima ejecución del cambiado de color.

Punto Extra

Analizar las siguientes capturas de pantalla y justificar tu respuesta :

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        view.backgroundColor = UIColor.red
    }

}
```

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
    // Creando el controlador
    let storyboard = UIStoryboard(name: "main", bundle: Bundle.main)
    let viewController = storyboard.instantiateViewController(withIdentifier: "ViewController")
    viewController.view.backgroundColor = UIColor.yellow
    // Se asigna el controlador como root de UIWindow
    window = UIWindow(frame: UIScreen.main.bounds)
    window?.rootViewController = viewController
    window?.makeKeyAndVisible()
    return true
}
```

Tomando en cuenta las capturas de código, mencionar cuál será el color de la vista del