Formas de desarrollar una interfaz en iOS.

Guillermo Aron Navarro de la Rosa aka Leyeneddx

196870

1 – Storyboard (Interfaz visual en Xcode)

Los Sotryboards nos permiten crear de una manera visual las vistas de nuestra app.

Ventajas:

* Visualización previa: Permite ver como se verá y funcionará la interfaz directamente en el editor gráfico.
* Fácil de usar: Ideal para desarrolladores principiantes, ya que no requiere mucha codificación para crear vistas.
* Diseño rápido: Puedes arrastrar y soltar elementos de la interfaz de usuario para gestionar diseño responsivo.
* Autolayout: Herramientas para gestionar el diseño responsivo.
* Transiciones automáticas: Configuración fácil de segues entre pantallas.

Desventajas:

* Escalabilidad limitada: Los proyectos grandes con múltiples pantallas pueden hacer que el storyboard se vuelva difícil de manejar.
* Problemas con el control de versiones: Como el storyboard es un archivo XML grande, los conflictos en Git pueden ser complicados de resolver.
* Menos personalizable: Puede ser restrictivo si necesitas lógica compleja o interfaces personalizadas.



2 – UIKit

Es una plataforma que posee un conjunto de herramientas que facilitan la creación de aplicaciones.

Ventajas:

* Flexibilidad total: Puedes personalizar completamente la interfaz y la lógica de diseño.
* Control completo: No dependes de archivos visuales, lo que es útil para diseños dinámicos o altamente personalizables.
* Mas ligero: No necesitas cargar archivos adicionales, lo que puede hacer que la aplicación sea más rápida.
* Facilidad para el control de versiones: Todo está en código, lo que reduce los conflictos.

Desventajas:

* Mayor curva de aprendizaje: Requiere un conocimiento más profundo de UIKit y Swift
* Diseño más lento: Crear interfaces en código lleva más tiempo que en Storyboard o XIB.
* Complejidad: La administración del diseño puede volverse complicada en proyectos grandes.



3 – SwiftUI

Es un entorno de trabajo que te ayuda a crear aplicaciones para cualquier dispositivo Apple.

Ventajas:

* Moderno y declarativo: Define la interfaz y el comportamiento de manera simple y legible.
* Código más limpio: Es más conciso y organizado en comparación UIKit.
* Compatibilidad nativa con todas las plataformas de Apple: Puedes usarlo en iOS, macOS, watchOS, y tvOS.
* Mejor integración con datos: Maneja cambios de estado fácilmente con @State y @Binding.

Desventajas:

* Compatibilidad limitada: Funciona solo en iOS 13 y versiones posteriores.
* Falta de soporte: Algunas funciones avanzadas aún no están completamente soportadas.
* Menor personalización: Puede ser restrictivo si necesitas control absoluto de ciertos detalles.
* Curva de aprendizaje: Aunque es más fácil de aprender que UIKit, requiere entender los conceptos declarativos.



Extra – XIB Files (Interfaces separadas por pantalla)

Es un editor grafico para desarrollo y pruebas de interfaces de usuario en Apple, almacena sus datos en archivos XIB. Estos archivos contienen básicamente información sobre la interfaz de una aplicación.

Ventajas:

* Modularidad: Cada archivo XIB representa una pantalla o vistas especifica, lo que facilita su mantenimiento y organización.
* Control individual: Menos propenso a problemas en el control de versiones, ya que cada vista está en su propio archivo.
* Ideal para componentes reutilizables: Perfecto para crear vistas personalizadas que se puedan usar en varias partes del proyecto.

Desventajas:

* Menos intuitivo: No puedes ver las transiciones entre pantallas como en un storyboard.
* Fragmentación: Gestionar múltiples archivos XIB puede ser complicado si el proyecto crece demasiado.
* Configuración adicional: Necesitas vincular los archivos XIB manualmente al controlador de vistas

