Desarrollo de interfaces iOS

SwiftUI

Ventajas

1. Una de las ventajas es el tipo de programación declarativa lo cual hace que el declarar las actividades es de manera mas abstracto. Esto permite la realización de aplicaciones más seguras, reutilizables y compactas con la menor cantidad de código.
2. Ofrece la opción de correspondencia en tiempo real del código. Esto ayuda a evitar tener que dar una serie de reglas para que la interfaz funcione de forma eficiente.
3. Los property wrappers ayudan a agregar funcionalidades extra a la propiedad donde se llevan a cabo sus acciones.
4. Se puede utilizar framework Combine en la programación de carga del flujo de datos. Esto ayuda a que tenga una compatibilidad multiplataforma (se crean aplicaciones para todos los dispositivos).
5. SwiftUI tiene compatibilidad con UIKit, lo cual crea la posibilidad de usar aplicaciones anteriores antiguas.
6. Tiene gran capacidad para crear una experiencia de usuario amigable y fluida entre todos los dispositivos iOS.

Desventajas

1. Aun es muy nuevo, la cantidad de recursos puede ser menor a las demás.
2. Hay limitaciones para encontrar documentación para resolver problemas específicos.
3. Puede ser más difícil aprender a usar esta herramienta para aquellos que están acostumbrados a trabajar con herramientas o frameworks tradicionales.
4. A pesar de tener compatibilidad con UIKit que ayuda a conectar con aplicaciones antiguas. Por si mismo, la compatibilidad con versiones anteriores puede ser difícil. Algunas funcionalidades solo están disponibles en versiones nuevas.

Storyboard

Ventajas

1. Elimina la necesidad de cierto código sencillo en el programa
2. Al crear tablas de contenido estático es mucho más rápido.
3. Es más sencillo agregar diseño.
4. Es más sencillo sumergirse en el desarrollo de iOS.
5. En pocas horas se puede crear un prototipo con su sencilla visualización sin tener que escribir tanto código.
6. La biblioteca de objetos es bastante grande.
7. La comunidad da un buen soporte.

Desventajas

1. Es difícil trabajar en equipo. Al todos tener que trabajar en el mismo fichero de storyboard es más sencillo que todo termine en caos.
2. Si se utiliza un sistema de control de versiones, si se usa el mismo fichero a la vez va a crear conflictos que se tendrán que resolver manualmente.
3. Tienen menos flexibilidad y control.
4. El control de la fuente es complicado, el código del storyboard está en XML, lo que complica resolver conflictos ya que el código es difícil de entender.
5. Hay que tener cuidado con el autolayout, esto hace que las interfaces se ajusten automáticamente, y se tiene que ajustar y revisar constantemente.

UIKit

Ventajas

1. Ahorra tiempo. Los componentes ya creados consiguen que no te tengas que preocupar en crearlos desde cero, ni por detalles de cada uno.
2. Ayuda a mantener coherencia y consistencia ya que mantiene la relación lógica entre sus partes y componentes. Esto da solidez a la interfaz.
3. Son versátiles. Algo ya existente se puede modificar y adaptar para el proyecto que se esté creando.
4. Tiene un extenso conjunto de API y componentes UI.
5. Tiene compatibilidad con Objective-C y Swift. Esto permite a los desarrolladores integrar código heredado.
6. Es bastante estable.

Desventajas

1. Se necesita escribir mucho más código que las anteriores dos opciones para gestionar eventos. Esto, a su vez, genera un código aún más complejo.
2. Es un proceso de desarrollo mucho más lento ya que los creadores deben dedicar tiempo al perfeccionar los diseños de interfaz de usuario.