Swift UI

Temario

1. Introducción a SwiftUI
    1.1. ¿Qué es SwiftUI?
    1.1.1. Diferencias con UIKit
    1.1.2. Ventajas y limitaciones de SwiftUI
    1.2. Entorno de desarrollo
        1.2.1. Creación de un nuevo proyecto en SwiftUI
        1.2.2. Simuladores y previsualización en tiempo real
    1.3. Estructura básica de una vista
        1.3.1. Sintaxis declarativa
        1.3.2. Uso de struct y @main
        1.3.3. Vista ContentView

2. Conceptos Básicos de SwiftUI
    2.1. Vistas y Controles esenciales
        2.1.1. Text. Image, Button, Spacer, Divider, etc
    2.2. Modificadores de vista
        2.2.1. Personalización con .font(), .foregroundColor(), .padding(), .background(), etc.
    2.3. Componentización
        2.3.1. Creación de vistas reutilizables
        2.3.2. Custom views con parámetros
    2.4. Estilos de controles
        2.4.1. Botones personalizados
        2.4.2. TextField y SecureField
    2.5. Imágenes y recursos
        2.5.1. Carga de imágenes locales y remotas (AsyncImage)
        2.5.2. Personalización de imágenes (escala, forma, recorte, sombreado)
    2.6. TabBar y Control de flujo
        2.6.1. Creación de TabBar con TabView
        2.6.2. Navegación dentro de cada pestaña

3. Gestión de Estado y Datos
    3.1. Estado y propiedades reactivas
        3.1.1. @State, @Binding, @ObservedObject, @EnvironmentObject
        3.1.2. Diferencias y uso práctico de cada propiedad
    3.2. Listas y Colecciones
        3.2.1. List y ForEach
        3.2.2. Uso de identificadores únicos (id)
        3.2.3. Modificación de listas y ScrollViewReader
    3.3. Formulario de entrada de datos

3.3.1. Uso de TextField, Toggle, Slider, Stepper y otros controles de entrada
3.3.2. Validación de formularios

4. Navegación y Arquitectura
4.1. Navegación entre pantallas
4.1.1. Uso de NavigationView, NavigationLink
4.1.2. Navegación con librería NavigationStack
4.1.3. Paso de datos entre pantallas
4.2. Patrones de arquitectura
4.2.1. Patrones de diseño en SwiftUI
4.2.2. Separación de lógica de negocio y lógica de vista
4.3. Inyección de dependencias
4.3.1. Inyección de dependencias con Container

5. Animaciones
5.1. Introducción a animaciones en SwiftUI
5.1.1. Animaciones implícitas con .animation()
5.1.2. Animaciones explícitas con .withAnimation()
5.2. Efectos visuales
5.2.1. Sombras, opacidad, desenfoques (.blur(), .shadow())

6. Diseño Avanzado de Interfaz de Usuario
6.1. Eventos avanzados
6.1.1. Uso avanzado de onChange, fullScreenCover, sheet
6.2. Calcular tamaño de vista dinámicamente
6.2.1. Creación de interfaces responsive (GeometryReader)
6.3. Estilos y temas
6.3.1. Personalización de temas y colores
6.3.2. Creación de paletas de colores personalizadas
6.3.3. Soporte para modo oscuro y modo claro

7. Asincronía y Concurrencia
7.1. Introducción a async/await
7.1.1. ¿Qué es la concurrencia en Swift?
7.1.2. Uso de async y await para tareas asíncronas
7.2. Consumo de APIs remotas
7.2.1. Uso de URLSession con async/await

8. Módulo 8: Integración con UIKit
8.1. Interoperabilidad con UIKit
8.1.1. Uso de UIViewControllerRepresentable y UIViewRepresentable
8.1.2. Inserción de vistas de UIKit en SwiftUI
8.1.3. Inserción de vistas de SwiftUI en UIKit (UIHostingViewController)

9. Módulo 9: Depuración de Vistas
    9.1. Depuración de vistas
        9.1.1. Uso de Previews (.previewLayout, .previewDevice)
        9.1.2. Inspección de vistas y depuración con Console