

# Intro to SwiftUI

Homero Oliveira

# Homero Oliveira

Desenvolvedor iOS na  
PoaTek



# Objetivos

- Declarativo
- Automático
- Composição
- Consistente

# O que é uma View?

- Similar a *UIView*
- É um protocolo e não uma classe.
- **View** deve sempre retornar uma **View** na propriedade *body*.

```
struct ContentView: View {  
    var body: some View {  
        Text("Hello World")  
    }  
}
```

# Criando *Stack*

- Se você quiser colocar três ***Labels*** acima um do outro
- HStack, VStack e ZStack

```
var body: some View {  
    VStack {  
        Text("Hello World")  
        Text("Hello World")  
        Text("Hello World")  
    }  
}
```

# Criando *List*

- Registrar célula
- Dizer quantidade de items
- Fazer dequeue
- Configurar a célula
- Etc...

# Criando *List*

```
struct UserRow: View {  
    var user: User  
  
    var body: some View {  
        Text("\(user.firstName) \(user.lastName)")  
    }  
}
```

```
struct ContentView: View {  
    var body: some View {  
        // create some example data  
        let user1 = User(id: 1, firstName: "Piper", lastName: "Chapman")  
        let user2 = User(id: 2, firstName: "Gloria", lastName: "Mendoza")  
        let users = [user1, user2]  
  
        // show that data  
        return List(users) { user in  
            UserRow(user: user)  
        }  
    }  
}
```

# Apresentando tela de detalhes

```
struct DetailView: View {  
    var selectedUser: User  
  
    var body: some View {  
        Text(selectedUser.lastName)  
            .font(.largeTitle)  
            .foregroundColor(.red)  
    }  
}
```

```
return NavigationView {  
    List(users) { user in  
        NavigationLink(destination: DetailView(selectedUser: user)) {  
            UserRow(user: user)  
        }  
    }.navigationBarTitle("Users")  
}
```



Demo

# UIKit equivalente no SwiftUI

UIKit	SwiftUI	Note
UILabel	Text	
UITabBar	TabView	
UITabBarItem	TabView	<code>.tabItem</code> under <code>TabView</code>
UITextField	TextField	For password ( <code>isSecureTextEntry</code> ) use <code>SecureField</code>
UITableView	List	also <code>VStack</code> and <code>Form</code>
UINavigationController	NavigationView	Part of <code>NavigationView</code>
UIBarButtonItem	NavigationView	<code>.navigationBarItems</code> in <code>NavigationView</code>
UICollectionView	-	
UIStackView	HStack	<code>.axis</code> == <code>.Horizontal</code>
UIStackView	VStack	<code>.axis</code> == <code>.Vertical</code>

# UIKit equivalente no SwiftUI

UIScrollView	ScrollView	
UIActivityIndicatorView	-	
UIImageView	Image	
UIPickerView	Picker	
UIButton	Button	
UIDatePicker	DatePicker	
UIPageControl	-	
UISegmentedControl	Picker	A style ( <code>SegmentedPickerStyle</code> ) of <code>Picker</code>
UISlider	Slider	
UIStepper	Stepper	
UISwitch	Toggle	
UIToolBar	-	

# Integração de frameworks

## - UIKit no SwiftUI

```
struct ActivityIndicator: UIViewRepresentable {  
    @Binding var isAnimating: Bool  
  
    func makeUIView(context: Context) -> UIActivityIndicatorView {  
        let v = UIActivityIndicatorView()  
  
        return v  
    }  
  
    func updateUIView(_ uiView: UIActivityIndicatorView, context:  
        Context) {  
        if isAnimating {  
            uiView.startAnimating()  
        } else {  
            uiView.stopAnimating()  
        }  
    }  
}
```

# Integração de frameworks

## - SwiftUI no UIKit

```
let vc = UIHostingController(rootView: ContentView())|
```

# Sugestões para aprendizado

<https://developer.apple.com/tutorials/SwiftUI>

Perguntas?

Obrigado