Cómo migrar a Jetpack Compose y No Morir en el intento 😎

Esteban Higuita

@soyEstebanDev
/estebanhiguitad
@estebanhiguitad

Erix Mendoza

@codemendozaa
/erix-javier-mendoza
@codemendozaa

Qué aprenderemos

- Introducción a Compose
- ¿Por qué migrar desde XML a Jetpack Compose?
- Errores comunes en la migración
- Estrategia para migrar
- Live coding

Introducción a Compose

- Es la nueva forma declarativa para crear interfaz de usuario
- Ya no tendremos mucho XML, en lugar de eso tendremos funciones
- El paradigma declarativo es la base para el desarrollo

Introducción a Compose

Declarar botón en XML

```
<Button
    android:id="@+id/my_button"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="24dp"
    android:layout_marginStart="24dp"
    android:layout_marginTop="32dp"
    android:minHeight="56dp"
    android:text="SUBMIT"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/other_view" />
```

Recuperarlo o crear un binding

Asignar el valor en algún lugar

```
my_button.text = getString(R.string.text_for_my_button)
```

Decidir en algún lugar si está o no habilitado

```
my_button.isEnabled = false
```

Asignar una acción

```
my_button.setOnClickListener { it: View!
    one_view.visibility = View.GONE
    other_view_to_handle.visibility = View.VISIBLE
    my_button.text = StringUtils.EMPTY_STRING
    animation_to_show_or_hide.visibility = View.VISIBLE
    animation_to_show_or_hide.playAnimation()
}
```

Introducción a Compose

En Compose verás algo similar a esto, dónde creamos y gestionamos **todo para el botón en el mismo lugar**

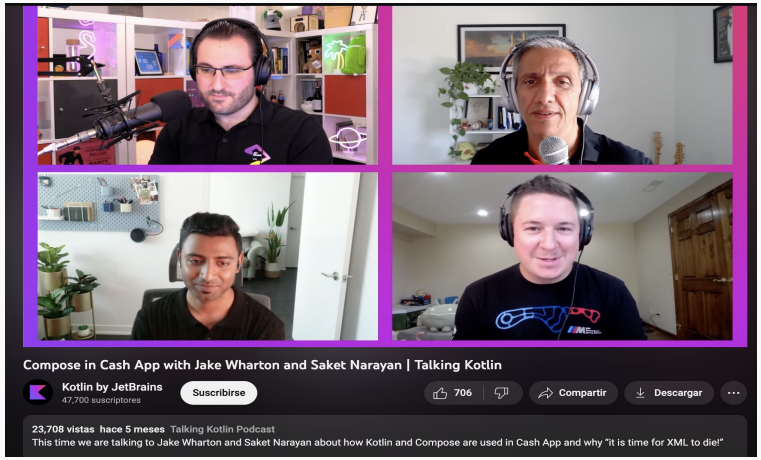
```
@Composable
fun MyButton(isEnabled: Boolean, onClick: () -> Unit) {
    Button(onClick = onClick, enabled = isEnabled) { this: RowScope
        Text(text = stringResource(id = R.string.text_for_my_button))
    }
}
```

¿Por qué migrar a Jetpack Compose?

- El radar de thoughtworks sugiere adoptarlo desde 2021
- Se va a optimizar el tiempo de desarrollo y la facilidad de mantener código
- Es una tendencia en el mercado hacer interfaz declarativa

¿Por qué migrar a Jetpack Compose?

- El equipo de Jake Wharton junto a Kotlin, sugieren migrar, y dicen algo como: “Es tiempo de que muera el XML”

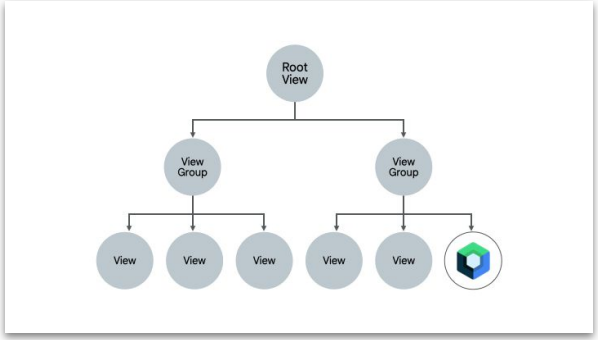


Errores comunes en una migración

- No tener una arquitectura clara antes de migrar
- Hacer refactor y migración al mismo tiempo
- No tener un plan
  - Definir una arquitectura a seguir
  - Capacitar los devs con buenas bases
  - Hacer pruebas de concepto de acuerdo al proyecto

Estrategias para migrar

1. Migrar todo

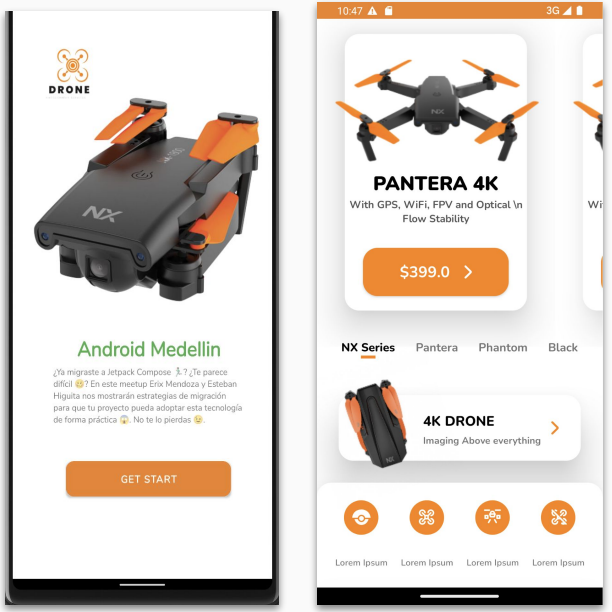


2. Migrar paulatinamente
    a. Crear nuevas funcionalidades con Compose
    b. Identificar componentes reutilizables, migrarlos y tener una librería para estilos
    c. Reemplazar de a poco funcionalidades existentes

Explicación del caso de estudio

App venta
tipo shopee

https://github.com/codemendozaa/MigratingfromXmltoCompose.git



Fuente:https://www.instagram.com/p/Cn7TphcJCqf/

Preguntas

Esteban Higuita

@soyEstebanDev
/estebanhiguitad
@estebanhiguitad

Erix Mendoza

@codemendozaa
/erix-javier-mendoza
@codemendozaa