

María Rodríguez García

TAREA: HUEVOXML

1. ¿Qué diferencias notas entre esta forma (XML + paleta + findViewById) y la que hicimos antes con Compose?

La principal diferencia es que con XML la interfaz se diseña en un archivo separado (activity_main.xml), usando etiquetas y una paleta de componentes visual. Luego, en el código Kotlin, hay que buscar cada vista con findViewById para poder interactuar con ella. En cambio, con Compose la interfaz se crea directamente con código Kotlin, sin necesidad de archivos XML ni findViewById. Cada elemento visual se define con funciones composables, como Text() o Button(). Compose también actualiza automáticamente la interfaz cuando cambian los datos, sin tener que hacerlo manualmente.

2. ¿Qué te ha parecido más fácil de entender y modificar?

Me ha parecido más fácil de entender Compose, porque todo está en el mismo archivo y se ve directamente qué hace cada parte del código. Además, no hace falta ir al archivo XML ni usar findViewById.

3. ¿Cuál te parece más cómoda para proyectos grandes?

Para proyectos grandes, me parece más cómodo Compose, porque permite organizar la interfaz en pequeñas funciones reutilizables y evita errores por ids o referencias nulas. Además, es más moderno y eficiente para mantener el código limpio. El sistema antiguo con XML puede volverse complicado al manejar muchos archivos y tener que conectar cada vista manualmente.