# Practica: Crear una app con diseño adaptable

Autor: Patrick Alexander Baeza Zamudio

Esta practica consiste en una aplicación Android utilizando Jetpack Compose que se adapta a diferentes tamaños de pantalla (como dispositivos móviles, tabletas y pantallas más grandes) mediante el uso de la API de WindowSizeClass. Esta aplicación configura la UI para que sea sensible a la orientación y el tamaño de la ventana, y renderiza el contenido de manera eficiente en función de estas características.

**Componentes Principales**

**1. MainActivity**

Clase principal que extiende ComponentActivity, siendo la entrada de la aplicación.

**Dentro del método onCreate:**

Se activa el modo Edge-to-Edge para permitir que la UI use toda la pantalla, incluyendo las áreas alrededor de la pantalla de forma más extendida (por ejemplo, bordes y áreas de notificación).

Se establece el contenido de la actividad mediante setContent, que configura la aplicación con un tema personalizado (ReplyTheme).

Se obtiene la dirección de diseño actual de la pantalla (LocalLayoutDirection.current) para calcular los márgenes de forma adecuada.

Se utiliza WindowInsets para aplicar el relleno necesario para respetar el área segura de la pantalla y evitar que el contenido se superponga con elementos de la interfaz del sistema.

La clase calculateWindowSizeClass(this) se utiliza para determinar el tamaño de la ventana y pasar esta información como un parámetro a ReplyApp.

**2. ReplyApp**

Componente de UI (no mostrado en el fragmento de código) que se importa desde com.example.reply.ui y que representa la aplicación principal o la estructura de la interfaz de usuario.

Recibe un parámetro windowSize que se utiliza para ajustar el diseño de la interfaz en función de la clase de tamaño de la ventana (compacta, media o expandida).

**3. @Preview**

Se definen tres vistas previas:

* ReplyAppCompactPreview: Vista previa para el tamaño de ventana compacto.
* ReplyAppMediumPreview: Vista previa para el tamaño de ventana mediano con un ancho de 700dp.
* ReplyAppExpandedPreview: Vista previa para el tamaño de ventana expandido con un ancho de 1000dp.

Estas vistas previas ayudan a verificar cómo se verá la aplicación en diferentes configuraciones de pantalla sin necesidad de ejecutarla en un dispositivo físico.

**Funcionamiento del Código**

**Inicio de la Aplicación:**

La actividad principal MainActivity se crea y configura con el tema ReplyTheme.

Se calcula la clase de tamaño de la ventana utilizando calculateWindowSizeClass(this), lo que permite que la aplicación adapte su contenido según el ancho disponible.

**Configuración de la IU:**

Surface se utiliza como un contenedor de la aplicación, aplicando márgenes dinámicos según el área segura de la pantalla.

ReplyApp se renderiza con el tamaño de la ventana actual, ajustando su diseño según sea necesario (compacto, mediano o expandido).

**Vista previa en Android Studio:**

Las funciones de vista previa permiten a los desarrolladores ver cómo se comporta la UI en diferentes tamaños de pantalla sin necesidad de desplegar la aplicación en un dispositivo físico o emulador.

**Características Técnicas**

**Uso de Jetpack Compose:** La interfaz de usuario se implementa de manera declarativa y moderna.

**Adaptabilidad de la IU:** La aplicación utiliza WindowSizeClass para manejar de manera eficiente el contenido en dispositivos de diferentes tamaños.

**Internacionalización y soporte de orientación:** Utiliza LocalLayoutDirection para adaptar el diseño a la dirección de la pantalla (LTR o RTL).

**Vistas previas:** Facilitan la comprobación de la apariencia de la interfaz en diferentes tamaños de pantalla.

**Ventajas del Enfoque**

**Interfaz adaptable:** El uso de WindowSizeClass permite que la aplicación ajuste automáticamente su contenido para aprovechar al máximo el espacio disponible en pantalla.

**Desarrollo eficiente:** La implementación de vistas previas en Android Studio agiliza el proceso de desarrollo al mostrar cambios en tiempo real.

**Conclusión**

Este código representa una base sólida para una aplicación que se adapta dinámicamente a diferentes tamaños de pantalla, proporcionando una experiencia de usuario optimizada en dispositivos de múltiples tamaños y resoluciones. Utiliza herramientas modernas de Jetpack Compose y la API WindowSizeClass para ofrecer una interfaz de usuario flexible y eficiente.