

## Buenas Prácticas para Desarrollo de Aplicaciones Nativas Android:

1. **Seguir el Material Design:** Construir la interfaz de usuario siguiendo las pautas de Material Design para proporcionar una experiencia de usuario coherente y atractiva.
2. **Utilizar ConstraintLayout:** Utilizar ConstraintLayout para crear interfaces de usuario flexibles y adaptables a diferentes tamaños de pantalla.
3. **Gestionar la Memoria de Manera Eficiente:** Evitar las fugas de memoria y gestionar los recursos cuidadosamente.
4. **Seguir el Principio de Separación de Responsabilidades:** Dividir el código en módulos y clases para facilitar la legibilidad y el mantenimiento.
5. **Utilizar Patrones de Diseño de Software:** Usar patrones como MVVM (Model-View-ViewModel) para facilitar la organización y escalabilidad del código.
6. **Optimizar el Rendimiento:** Realizar pruebas de rendimiento y optimización para asegurarte de que la aplicación funcione sin problemas.
7. **Gestionar los Ciclos de Vida de la Actividad/Fragamento:** Manejar adecuadamente los eventos del ciclo de vida para tener estabilidad.
8. **Gestionar la Persistencia de Datos:** Utilizar Room o SharedPreferences para almacenar y recuperar datos de manera eficiente.

## Consideraciones para las pruebas manuales (equipo de QA)

1. **Planificación de Pruebas:** Elaborar un plan de pruebas detallado que incluya casos de prueba para todas las funcionalidades y escenarios posibles.
2. **Dispositivos y Sistemas Operativos:** Probar la aplicación en una variedad de dispositivos y versiones de sistemas operativos para garantizar la compatibilidad.
3. **Registro de Errores:** Mantener un registro detallado de los errores encontrados, incluyendo pasos para reproducirlos y su gravedad.
4. **Documentación de Pruebas:** Documentar los casos de prueba, resultados y problemas encontrados para futuras referencias y para compartir con el equipo de desarrollo.