Universidad del Valle de Guatemala
Plataformas móviles
Esteban Zambrano

Laboratorio 8

Legibilidad y Mantenibilidad:

Corrutinas: Las corrutinas facilitan la redacción de código asincrónico de forma que se puede leer secuencialmente, similar a la forma en que se redactaría el código síncrono, lo cual mejora la legibilidad y facilita la mantenibilidad del código.
Callbacks: Los callbacks pueden conducir a una situación denominada "callback hell" o "infierno de callbacks", en la cual el código se torna complicado para leer y mantener debido a los varios niveles de anidamiento involucrados.

Gestión de errores:

Corrutinas: Ofrecen una gestión de errores más simplificada y directa mediante el uso de estructuras try/catch, facilitando así la identificación y el manejo de errores en el código asincrónico.
Callbacks: En el caso de los callbacks, la gestión de errores puede resultar más complicada y podría necesitar de estructuras adicionales o verificaciones para asegurar una adecuada gestión de los errores.

Composición y Orquestación:

Corrutinas: Facilitan la composición y orquestación de operaciones asincrónicas, permitiendo que múltiples tareas asincrónicas se coordinen de manera sencilla.
Callbacks: La composición y coordinación de operaciones asincrónicas puede ser más desafiante y menos intuitiva con callbacks, especialmente cuando hay múltiples operaciones asincrónicas interdependientes.

Performance y Uso de Recursos:

Corrutinas: Son más ligeras en comparación con los threads, y permiten una gran cantidad de operaciones asincrónicas con un menor uso de recursos.
Callbacks: No tienen un impacto significativo en el uso de recursos, pero pueden llevar a un uso ineficiente de los threads si no se manejan correctamente.

Flexibilidad y Escalabilidad:

Corrutinas: Ofrecen una mayor flexibilidad y escalabilidad, permitiendo un fácil escalado de operaciones asincrónicas sin un aumento significativo en la complejidad del código.
Callbacks: Pueden volverse más difíciles de gestionar a medida que aumenta el número de operaciones asincrónicas y la complejidad del código.

Concurrencia Estructurada:

Corrutinas: Proporcionan un modelo de concurrencia estructurada que facilita el manejo seguro de operaciones concurrentes.
Callbacks: No proporcionan un modelo de concurrencia estructurada, lo que puede llevar a condiciones de carrera y otros problemas de concurrencia.

Interoperabilidad:

Corrutinas: Kotlin proporciona una buena interoperabilidad entre corrutinas y otras características y bibliotecas, lo que permite una integración fluida con otras partes del ecosistema de Kotlin.
Callbacks: La interoperabilidad puede ser limitada o requerir adaptadores adicionales para trabajar bien con otras bibliotecas y características.