Aroldo Xavier López Osoy - 22716

¿Cuáles son las ventajas de usar corrutinas vs Callbacks?

Las corrutinas en Kotlin ofrecen diversas ventajas sobre los callbacks tradicionales, como las que se muestra a continuación:

- Código más legible: Las corrutinas permiten escribir código asíncrono que parece síncrono. Es decir, puedes escribir código secuencialmente sin tener que anidar callbacks, lo que reduce el fenómeno conocido como "callback hell" o "pyramid of doom".

- Menos sobrecarga: Las corrutinas son más ligeras que los threads tradicionales. Se pueden lanzar miles o incluso millones de corrutinas en un solo thread sin que esto represente un gran costo en recursos.

- Composición flexible: Es fácil combinar varias operaciones asíncronas con corrutinas. Por ejemplo, se pueden usar funciones como async y await para ejecutar tareas en paralelo y esperar a que todas se completen.

- Manejo de errores simplificado: Con los callbacks, el manejo de errores puede volverse complicado debido a la necesidad de propagar errores a través de diferentes niveles de anidamiento. Las corrutinas, con la ayuda de try-catch, simplifican este proceso.

- Integración con librerías: Kotlin Coroutines tiene una integración profunda con muchas librerías populares. Esto significa que se pueden aprovechar el poder de las corrutinas con tus bibliotecas favoritas sin tener que hacer muchas adaptaciones.

- Soporte de cancelación: Las corrutinas tienen un soporte integrado para la cancelación de tareas. Se puede cancelar una corrutina y, automáticamente, cualquier tarea secundaria o recurso asociado también se gestionará adecuadamente.

- Scope de ejecución: Las corrutinas en Kotlin ofrecen conceptos como CoroutineScope que ayudan a delinear el ciclo de vida de las operaciones asíncronas. Esto es útil para evitar fugas de memoria y para asegurarse de que las operaciones asíncronas se cancelen cuando ya no sean necesarias (por ejemplo, cuando una actividad o fragmento en Android es destruido).

- Soporte nativo en Kotlin: Dado que las corrutinas son una característica nativa de Kotlin, se benefician de optimizaciones y soporte continuo por parte del equipo de desarrollo de Kotlin.

- Interoperabilidad con otras soluciones asíncronas: Aunque se decida utilizar corrutinas, no se está limitado a ellas. Las corrutinas pueden interoperar con otros mecanismos asíncronos como RxJava, Future, etc.

- Control de flujo: Con herramientas como flow en el paquete de corrutinas, se puede trabajar con flujos de datos reactivos de manera más sencilla y legible.