Desarrollo de aplicaciones móviles II

Tema Nº 3:Corrutinas

Indicador de logro Nº3:Reconoce la Programación Orientada a Objetos en Kotlin, considerando los principios SOLID.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº3:**

Corrutinas

**Subtema 3.1:**

Introducción a corutinas en Kotlin.

Una corrutina es un patrón de diseño de simultaneidad que puedes usar en Android para simplificar el código que se ejecuta de forma asíncrona. Las corrutinas se agregaron a Kotlin en la versión 1.3 y se basan en conceptos establecidos de otros lenguajes.

En Android, las corrutinas ayudan a administrar tareas de larga duración que, de lo contrario, podrían bloquear el subproceso principal y hacer que tu app dejara de responder. Más del 50% de los desarrolladores profesionales que usan corrutinas informaron que vieron un aumento en la productividad. En este tema, se describe cómo puedes usar las corrutinas de Kotlin para solucionar estos problemas, lo que te permite escribir código de apps más limpio y conciso.

**Funciones**

Las corrutinas son nuestra solución recomendada para la programación asíncrona en Android. Las funciones más importantes son las siguientes:

* Ligereza: Puedes ejecutar muchas corrutinas en un solo subproceso debido a la compatibilidad con la suspensión, que no bloquea el subproceso en el que se ejecuta la corrutina. La suspensión ahorra más memoria que el bloqueo y admite muchas operaciones simultáneas.
* Menos fugas de memoria: Usa la simultaneidad estructurada para ejecutar operaciones dentro de un alcance.
* Compatibilidad con cancelación incorporada: Se propaga automáticamente la cancelación a través de la jerarquía de corrutinas en ejecución.
* Integración con Jetpack: Muchas bibliotecas de Jetpack incluyen extensiones que proporcionan compatibilidad total con corrutinas. Además, algunas bibliotecas proporcionan su propio alcance de corrutina, que puedes usar para la simultaneidad estructurada.

**Ejemplo:**

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

**Subtema 3.2:**

Corutinas vs Hilos.

Un hilo está directamente relacionado con el hilo nativo en el sistema operativo correspondiente (sistema operativo) y consume una cantidad considerable de resources. En particular, consume mucha memory para su stack. Es por eso que no puedes simplemente crear hilos de 100k. Es probable que se quede sin memory. El cambio entre hilos implica al asignador de núcleo del sistema operativo y es una operación bastante costosa en términos de ciclos de CPU consumidos.

Una corutina, por otro lado, es puramente una abstracción de lenguaje a nivel de usuario. No vincula ningún recurso nativo y, en el caso más simple, utiliza solo un object relativamente pequeño en el montón de JVM. Es por eso que es fácil crear 100k corutinas. El cambio entre corutinas no implica kernel de sistema operativo en absoluto. Puede ser tan barato como invocar una function regular.

**Subtema 3.3:**

Cancellation y Timeouts.

**Ejemplos:**

Imagen que contiene Word

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Subtema 3.4:**

Funciones y canales.

**Ejemplos:**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Imagen que contiene Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Actividad:**

* ¿Qué diferencias tiene los AsyncTask con las Corrutinas?
* ¿En donde se puede implementar las corrutinas ?
* ¿Qué métodos podemos utilizar en las corrutinas?
* ¿Se puede aplicar expresiones lambda en corrutinas? ¿Porqué?
* Implemente dos ejemplos prácticos de corrutinas.

**Conclusiones de la experiencia:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Referencias**

*Coroutines. (2021).* Kotlin. <https://kotlinlang.org/docs/coroutines-overview.html>

*Corrutinas de Kotlin en Android. (2021).* Android Developer. <https://developer.android.com/kotlin/coroutines?hl=es-419&gclid=CjwKCAjw55-HBhAHEiwARMCsznlxSCKP0wGmLrhgSNoqREZ1L0k1D-tVahWA6rBs5YXiI11pFHDlaBoCr-MQAvD_BwE&gclsrc=aw.ds>