

Instituto Tecnológico de Costa Rica
Unidad de Computación

Desarrollo de Aplicaciones Móviles
“Investigación clase Thread y AsyncTask”

Laurenth Chaves Méndez
Jonathan Rojas Vargas
Kevin Walsh Muñoz

Sede San Carlos
30/3/2016



Contenido

Clase Thread.....	3
Clase AsyncTask.....	4
Bibliografía	6

Clase Thread

Todos los componentes de una aplicación Android, tanto las actividades, los servicios o los *broadcast receivers* se ejecutan en el mismo hilo de ejecución, el llamado *hilo principal*, *main thread* o *GUI thread*, éste último nombre indica es el hilo donde se ejecutan todas las operaciones que gestionan la interfaz de usuario de la aplicación.

La clase Thread se encarga de crear un proceso o hilo en segundo plano. Se pueden crear todos los hilos (Thread) que sean necesarios. Estos hilos dejan de ser parte de la aplicación y funcionan de manera independiente, generando el problema de que los hilos no pueden modificar ni insertar datos en hilo principal. Para solucionar este problema existe los Handler.

Handler

Son un puente un hilo secundario (thread) y el hilo principal (aplicación).

Clase AsyncTask

AsyncTask permite el uso apropiado y fácil del hilo de interfaz de usuario. Esta clase permite realizar operaciones en segundo plano y publicar los resultados sobre el hilo de interfaz de usuario sin tener que manipular los hilos.

AsyncTask también está diseñado para ser una clase de ayuda al thread y al handler, este no constituye al marco genérico del threading. Las tareas asíncronas son ideales para los procesos de corto tiempo. Sin embargo en situaciones es necesario mantener subprocesos que se ejecutan durante largo tiempo.

Algunas API recomendadas a utilizar, proporcionadas por el paquete `java.util.concurrent` tales como `Executor`, `ThreadPoolExecutor` y `FutureTask`.

Existen tres tipos de AsyncTask:

- **Params:** El tipo de parámetros que son enviados a la tarea en ejecución.
- **Progress:** Es el tipo de las unidades de progresos publicados durante el cálculo de fondo.
- **Result:** Es el tipo de resultado del cálculo de fondo.

También existen cuatro pasos de suma importancia, los cuales son los siguientes:

1. **onPreExecute():** Este es invocado en el subproceso de la interfaz para mostrar al usuario una barra de progreso. También se utiliza para la configuración de la tarea.
2. **doInBackground(Params...):** Este subproceso es invocado de fondo inmediatamente después del método `onPreExecute()` termina de ejecutarse. Este paso se utiliza para realizar el cálculo de fondo que puede llevar mucho tiempo. Los parámetros de la tarea asíncrona se pasan a este paso. El resultado del cálculo debe ser devuelto por este paso y se pasa de nuevo al último paso. Este paso también se puede utilizar en el método `publishProgress (Progress ...)` para publicar una o más unidades de progreso. Estos valores se publican en el subproceso de interfaz de usuario, en el paso del método `onProgressUpdate (Progress ...)`.

3. **onProgressUpdate(Progress...):** Este subproceso es invocado en la interfaz de usuario después de llamar al método `publishProgress(Progress ...)`. El momento de la ejecución no está definido. Este método se utiliza para mostrar de cualquier forma de progreso en la interfaz de usuario, mientras que el cálculo de fondo se sigue ejecutando. Por ejemplo, se puede utilizar para mostrar una barra de progreso o mostrar los registros en un campo de texto.
4. **onPostExecute(Result...):** Este método es invocado en el subproceso de interfaz de usuario después de que finalice el cálculo de fondo. El resultado del cálculo de fondo se pasa a este paso como un parámetro.

Bibliografía

(2016, 03). Developers. Obtenido de <http://developer.android.com/intl/es/reference/android/os/AsyncTask.html>

(2013, 09). jarroba. Obtenido 03, 2016, de <http://jarroba.com/asynctask-en-android/>

(2013, 09). Tutoriales HTML5 Obtenido de <http://www.tutorialeshtml5.com/2012/05/android-thread-hilo-y-handler-proceso.html>