***Tecnológico de Costa Rica***

***Sede Regional San Carlos***

***Escuela Ingeniería en Computación***

***Desarrollo de Aplicaciones para Dispositivos Móviles***

***Tarea #2***

***“Investigación de Asynctask”***

***Responsable:***

***Henry Solís Chacón 2013085706***

***Hellen Rojas Rojas 2013083934***

***30 de marzo del 2016***

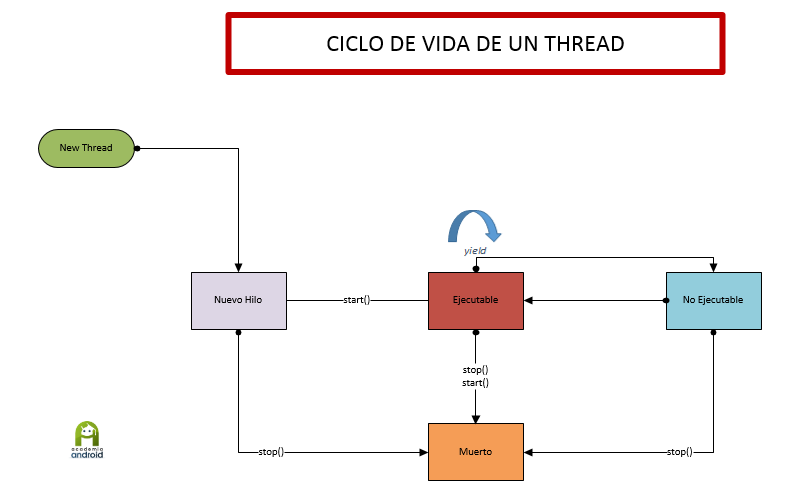
***Santa Clara, San Carlos***

**Asynctask y Thread**

Los hilos son una clase que contiene su unidad de ejecución, y es posible definirla como “**la unidad de procesamiento más pequeña que puede ser planificada por un sistema operativo”.** Una de sus características principales es la realización de  **varias tareas de simultáneamente**

Los hilos se han utilizado en Android y en otros lenguajes de programación para realizar tares en segundo plano y no interrumpir el hilo principal. Estos pueden ser muy útiles cuando se tienen cálculos largos que requieren mucho tiempo, ya que ente caso se puede hacer un hilo y encomendarle esta tarea y así permitir al hilo principal realizar las otras tareas sin problemas.

Los métodos principales de la clase Thread son start() ,stop() y run();



Ahora existe una clase especial llamada AsyncTask, que permite **comunicarse con** el subproceso del  hilo principal. Para esto ejecuta operaciones en segundo plano, mostrando así los resultados en subprocesos de la interfaz de usuario. Y además facilita mucho la vida y es mucho más óptimo.

La clase AsyncTask, contiene métodos principales (claves) a sobrescribir que se ejecutan para realizar operaciones que necesiten actualizaciones en la interfaz. Los métodos son los siguientes:

onPreExecute: este se ejecuta antes de iniciar el proceso en el mismo hilo de ejecución que la aplicación. Se usa en algunos casos para inicializar alguna barra de progreso que informe del estado de una tarea.

doInBackground(): inicia la tarea en un hilo distinto al de ejecución para no bloquear la interfaz. Es el único método que es obligatorio sobrescribir, y muchas veces es el único que se utiliza.

onProgressUpdate(): este método obtiene información acerca el progreso de la tarea utilizando la función publishProgress(). Se ejecuta en el hilo principal y se puede utilizar para informar al usuario del progreso.

onPostExecute: se utiliza cuando finaliza el método doInBackground() y recibe el resultado para procesarlo y actualizar la interfaz.

