6. UI Fragments y Fragment Manager

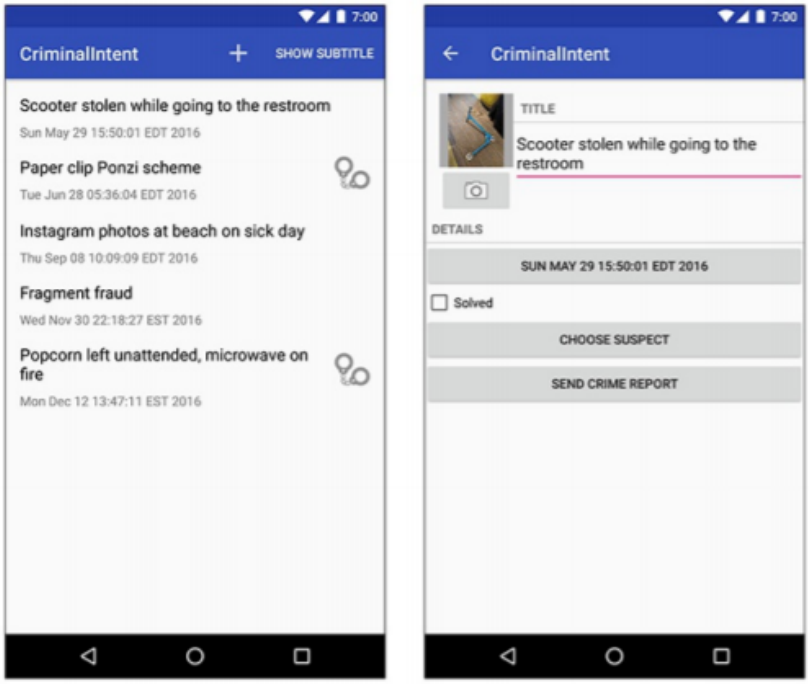
Este capítulo corresponde al capítulo 7 del libro oficial *Android Programming Guide Ranch*.

En este capítulo se empezará a programar otra aplicación distinta a GeoQuiz. La nueva aplicación se llamará Criminal Intent.

Criminal Intent mantiene un registro de crímenes, donde cada uno de ellos tendrá un título, una fecha y una foto. También se podrá identificar a un sospechoso desde la lista de contactos y presentarle una queja por correo electrónico, Twitter, Facebook u otras aplicaciones.

Criminal Intent será desarrollada durante 13 capítulos del libro oficial y será la app más compleja de desarrollar de toda la guía, ya que contendrá muchos conceptos de Android como, vistas, listas, fragments…

Un ejemplo de cómo será la aplicación visualmente es la siguiente.



La necesidad de una UI flexible

Es normal pensar que en una aplicación en la que nos aparece una lista de elementos habrá dos activities: una para la lista y otra para el detalle del elemento en el que cliquemos.

Funcionaría, pero si queremos una presentación más sofisticada deberemos modelar la aplicación para que corra de distinta forma si el dispositivo se encuentra en vertical/horizontal o bien si se trata de una Tablet.

Una UI flexible se define por la habilidad de componer y descomponer la vista de una activity en tiempo de ejecución dependiendo del dispositivo o de la orientación de este.

La propia Activity de Android no nos permite esa flexibilidad. La vista de la Activity cambiará en tiempo de ejecución, pero el código de control de esas vistas debería vivir dentro de esa activity.

Introducción a Fragments

Un Fragment es un controlador que una activity puede usar para delegar algunas tareas. La tarea más común es la de manejar la UI de la Activity, en la que ésta puede ser la pantalla entera o bien solo una parte de ella.

La parte que queramos que cambie según la orientación o el dispositivo, será nuestro fragment, y a éste le inflaremos el layout correspondiente, consiguiendo de esta forma que exista tan sólo una activity con distintos aspectos.

Nota: Cabe recalcar que una activity podrá disponer de más de un fragment en su vista.

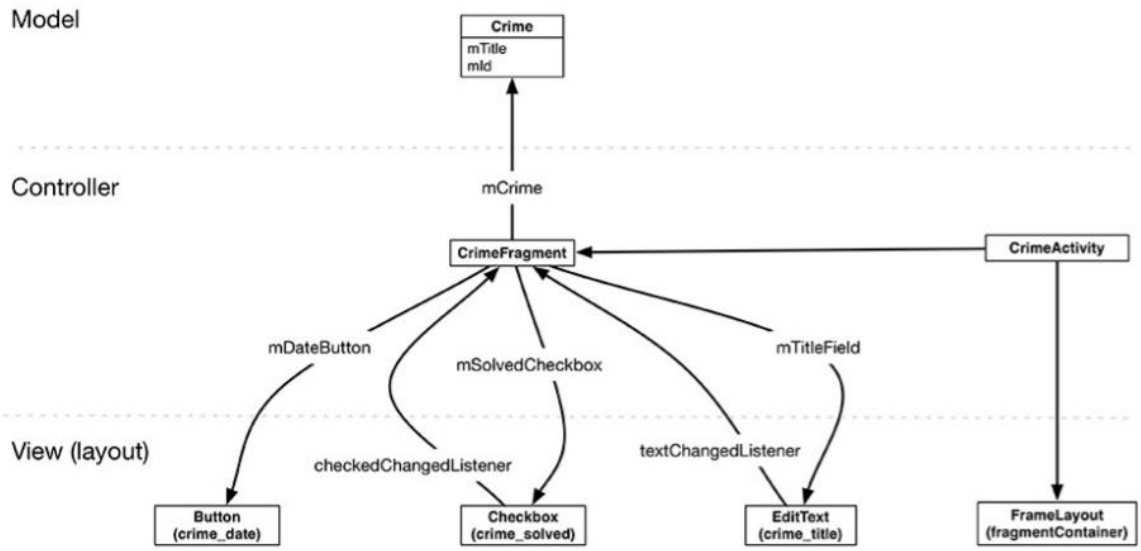
Empezando CriminalIntent

En este capítulo empezaremos diseñando la parte de creación de un crimen, en la que deberemos detallar un título para el crimen cometido y como detalle tendremos la fecha en la que se ha realizado.

Este será el aspecto, al final del capítulo, de la pantalla que diseñaremos.

La pantalla será gestionada por un fragment al que llamaremos CrimeFragment, que será albergado por una activity a la que llamaremos CrimeActivity. Por ahora pensemos que el layout que mostraremos por pantalla lo contendrá el fragment, y que la activity contendrá ese fragment.

Al tratarse de un proyecto que será muy largo, sería bueno que el estudiante se hiciera a la idea, a grandes rasgos, de cómo será el proyecto y, así, no perder la concentración y saber qué es lo que se está haciendo en todo momento. Para ello el siguiente esquema de cómo se va a estructurar el CrimeFragment que crearemos.



Tres de las clases que podemos ver en el esquema van a ser programadas por nosotros: Crime, CrimeFragment y CrimeActivity.

El proyecto

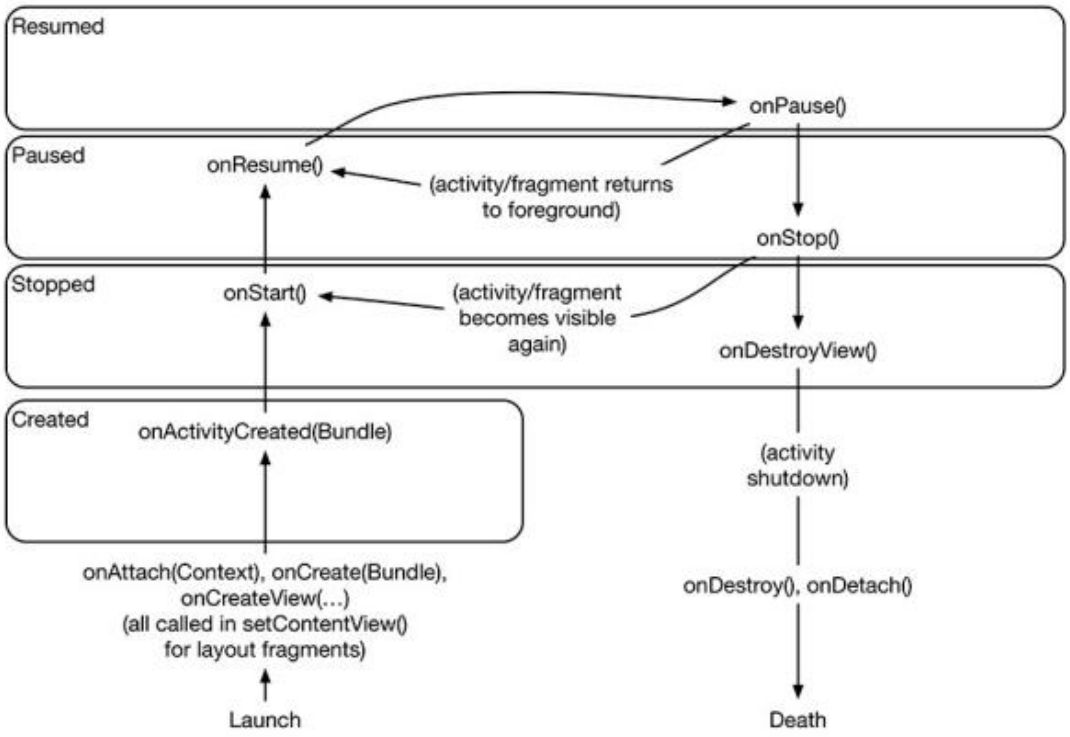
Una vez descarguemos el proyecto Android, veremos que las implementaciones de las clases Crime, CrimeFragment y CrimeActivity, no son complicadas. Aunque el término de Fragment nos pueda parecer un poco extraño, veremos fácil como funciona:

Primero de todo nos centraremos en la clase CrimeActivity:

La clase CrimeActivity será la clase que se ejecutará al inicio de nuestra aplicación. Ésta tan solo contendrá el método onCreate(), que inicializará el fragment para que muestre el contenido que queramos.

Podemos ver también el layout de la CrimeActivity, para así ver que lo único que tiene es un FrameLayout, en el que cargaremos el fragment.

El CrimeFragment que tenemos creado es una instancia de Fragment, con lo cual hereda todos los métodos de este. Esos métodos hacen relación al ciclo de vida que tiene un fragment, ya que un Fragment tiene su propio ciclo de vida, que podemos ver en el siguiente esquema.



Como podemos ver, el ciclo de vida de un Fragment se asemeja mucho con el ciclo de vida de una Activity y están relacionados entre ellos.

Por el momento solo implementaremos los métodos onCreate y onCreateView de nuestro Fragment:

- onCreate: El método que crea la instancia de crimen.
- onCreateView: Función que devuelve la View que hemos inyectado en el fragment a partir de su layout. Se recomienda al alumno que le eche un vistazo a esta función.

Nota: Un gran aspecto a tener en cuenta es que, a diferencia con el ciclo de vida de una Activity, estos métodos no son llamados por el sistema operativo, sino que son llamados por la propia Activity que los contiene.

Para el CrimeFragment deberemos tener una o más layouts que le queramos inyectar, así que deberemos crear otro resource layout file para ello. En el definiremos los elementos de detalle de un crimen y lo llamaremos fragment_crime.xml.

Además, crearemos la clase Crime, que será referente a un crimen.