netmind

WeKnowIT

JJD311 – Desarrollo Avanzado de Apps para Android 5 Lollipop

Material práctico

Práctica 1 Desarrollo Avanzado de Apps para Android 5 Lollipop

Fragments

En este caso práctico se trabajarán los conceptos aprendidos en el tema 1 "Fragments" del material didáctico proporcionado por *net**mind***.

La metodología a seguir consistirá en realizar, conjuntamente con el formador, una aplicación Android que incluya Fragments, permitiendo así una flexibilizar el código para los dos modos de visualización disponibles: vertical o *portrait* y horizontal o *landscape*.

Objetivo

Conocer, estudiar y emplear con soltura los Fragments en Android. Analizar las situaciones en las que es conveniente usar este tipo de elementos y entender el procedimiento que permite la comunicación ágil y precisa entre un Fragment y la Activity que los alberga.

Contenidos

- Creación de Fragments
- Tipos de Fragments
- Añadiendo Fragments a una Activity
- Gestión de Fragments y comunicación con una Activity

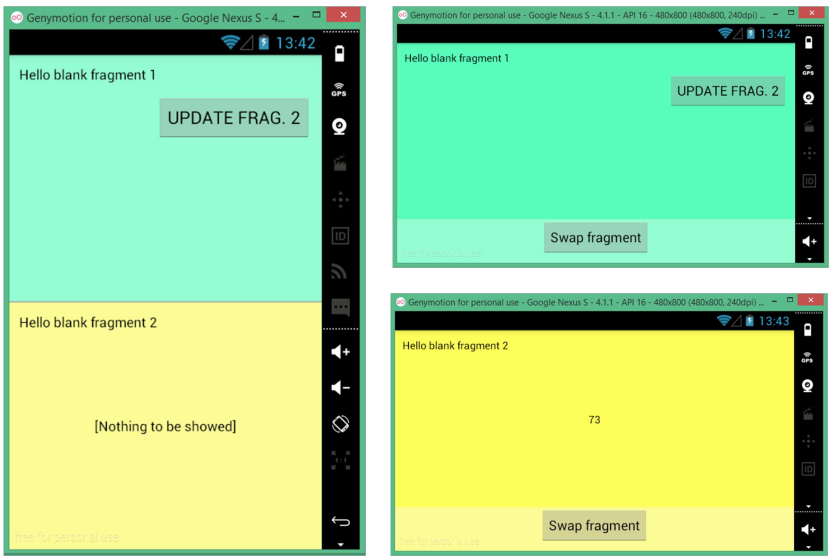
Planteamiento del caso

La aplicación a implementar consta de dos Fragments que se utilizan de forma distinta en función del modo de visualización: en modo *portrait* se muestran uno encima de otro, mientras que en modo *landscape* solo uno se ve en pantalla.

- Modo *portrait*. Ambos Fragments se añaden "estáticamente" al layout. Al accionar el botón del Fragment superior, un número aleatorio es generado y mostrado en el Fragment inferior.
- Modo *landscape*. Los Fragments se añaden alternativamente y de forma dinámica al layout cuando el usuario pulsa sobre el botón "Swap fragment". El botón del Fragment superior sigue teniendo la misma funcionalidad, por lo que al ser accionado en este modo provocará que se añada el segundo Fragment y que éste muestre un número aleatorio.

El desarrollador debe comprobar el correcto funcionamiento de la aplicación en los diferentes modos de visualización. Para ello, deberá emplear un terminal físico o hacer uso de la funcionalidad "rotar pantalla" disponible en el emulador.

Resultado final



Soluciones

El proyecto completo para Android Studio puede encontrarse en la carpeta "FragmentManager" o en el siguiente enlace de repositorio de GitHub:

https://github.com/pablodeafsapps/Netmind-JJD311_AndroidProjects/tree/master/FragmentManager

Al finalizar la práctica has aprendido…

Conocer y entender los Fragments, que son subprocesos independientes que implementan su propio ciclo de vida y que pueden ser fácilmente integrables en cualquier Activity. A lo largo de este caso práctico se ha examinado y detallado el funcionamiento de estos componentes cuya relevancia ha crecido enormemente desde su aparición en la versión 3.0.

Práctica 2 Desarrollo Avanzado de Apps para Android 5 Lollipop

Adaptadores de datos

En este caso práctico se trabajarán los conceptos aprendidos en el tema 2 "Adaptadores de datos" del material didáctico proporcionado por *net**mind***.

La metodología a seguir consistirá en realizar, conjuntamente con el formador, una aplicación Android que haga uso de un adaptador de datos que permita interrelacionar una fuente de datos con un elemento gráfico (View) para su visualización.

Objetivo

Conocer, estudiar y emplear con soltura los adaptadores de datos en Android. Analizar las situaciones en las que es conveniente usar este tipo de elementos y aprender a implementarlos convenientemente para optimizar el rendimiento general de la aplicación.

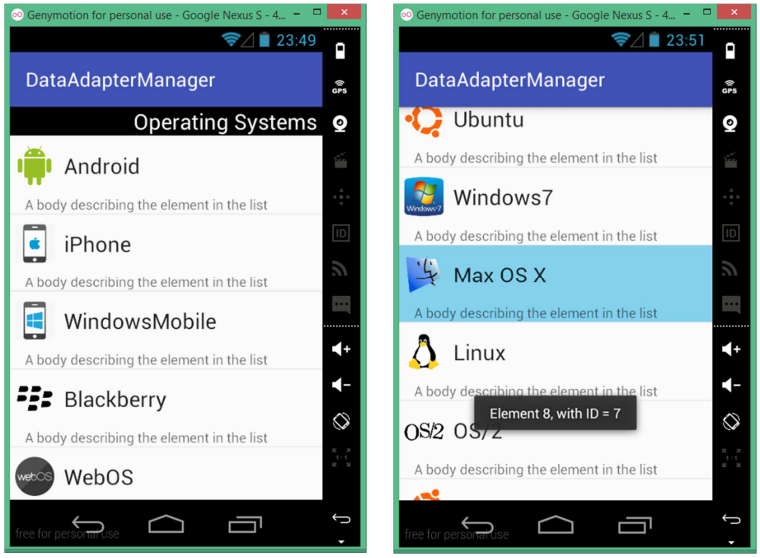
Contenidos

- Introducción de Adapters y listeners
- Adapters por defecto y ArrayAdapter
- Creación de Adapters personalizados
- Adapters optimizados con RecyclerView

Planteamiento del caso

La aplicación a implementar consta principalmente de un ListView que permite mostrar datos en una lista desplazable de elementos seleccionables. A lo largo de la práctica se irá modificando la implementación del ListView para ilustrar los diferentes Adapters y listeners que pueden emplearse a la hora de conseguir una aplicación más robusta y flexible a futuros cambios.

Resultado final



Soluciones

El proyecto completo para Android Studio puede encontrarse en la carpeta "DataAdapterManager" o en el siguiente enlace de repositorio de GitHub:

https://github.com/pablodeafsapps/Netmind-JJD311_AndroidProjects/tree/master/DataAdapterManager

Al finalizar la práctica has aprendido…

En este capítulo se han presentado y explicado los *Adaptadores de datos*, que son elementos del sistema que hacen las veces de interfaz entre un widget de la UI y una fuente de datos (arrays, Content Providers, etc.). A lo largo del caso práctico se ha examinado y detallado el funcionamiento de estos componentes que se hacen imprescindibles para cualquier desarrollador que desee que sus aplicaciones implementen widgets para mostrar listas de datos.

Práctica 3 Desarrollo Avanzado de Apps para Android 5 Lollipop

Material Design

En este caso práctico se trabajarán los conceptos aprendidos en el tema 3 "Diseño avanzado GUI: Material Design" del material didáctico proporcionado por *net**mind***.

La metodología a seguir consistirá en realizar, conjuntamente con el formador, una aplicación Android que siga las convenciones de diseño instauradas con la llegada de Material Design a partir de Android 5 Lollipop, Se incluirán elementos típicos de una aplicación "convencional" Android, como menús y desplegables.

Objetivo

Conocer, estudiar e implantar los principios de diseño de Material Design en una aplicación Android. La idea es presentar los elementos más relevantes a incluir en la aplicación y dotarlos de las características y funcionalidades necesarias recomendadas a la hora de implementar este nuevo paradigma de diseño.

Contenidos

- Principios de diseño con Material Design con la biblioteca Android Design Support
- *Themes* y paleta de colores para crear superficies tangibles y diseño similar al editorial
- Mejoras prácticas para estructurar elementos visuales y mejorar la navegación
- Animación y retroalimentación táctil para expresar movimiento con significado

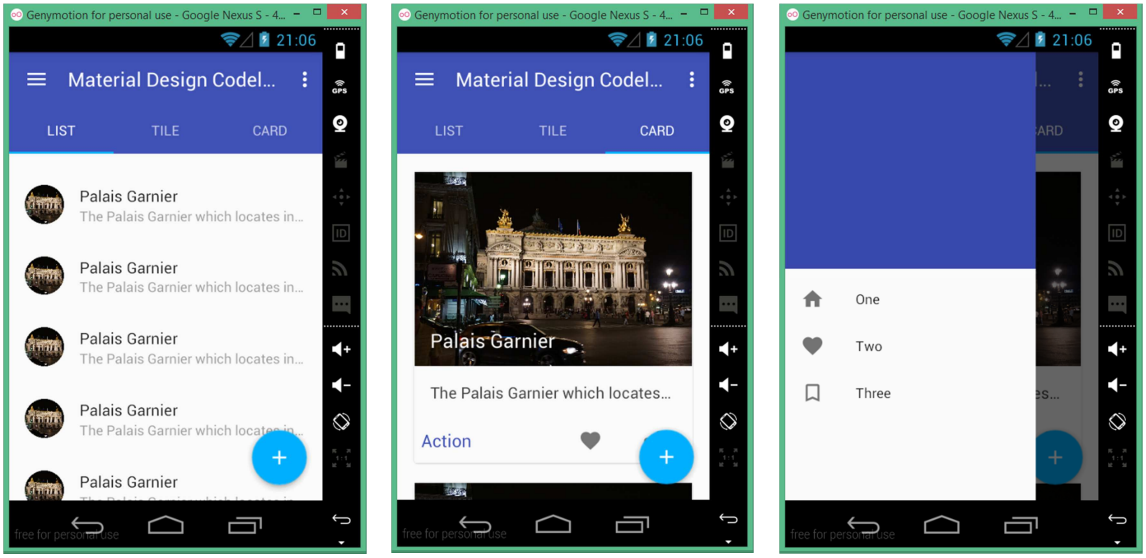
Planteamiento del caso

La aplicación a implementar está basada en la solución proporcionada en el siguiente enlace:

https://codelabs.developers.google.com/codelabs/material-design-style-sp/index.html?index=..%2F..%2Findex#0

El layout principal incluirá un ListView desplazable de elementos seleccionables, tres pestañas y la App Bar (Toolbar). Al seleccionar las distintas pestañas se accede a nuevas interfaces que incluyen otros elementos destacables como la CardView o el Navigation Drawer.

Resultado final



Soluciones

El proyecto completo para Android Studio puede encontrarse en la carpeta "MaterialDesignManager" o en el siguiente enlace de repositorio de GitHub:

https://github.com/pablodeafsapps/Netmind-JJD311_AndroidProjects/tree/master/MaterialDesignManager

Al finalizar la práctica has aprendido…

En esta práctica se ha presentado y explicado una de las principales novedades introducidas en la versión Android 5.0 Lollipop: *Material Design*. Este paradigma de diseño consiste en una serie de reglas y normas de estilo que persiguen homogeneizar el aspecto de las aplicaciones desarrolladas para la plataforma Android, para mejorar así la experiencia de usuario.

Práctica 4 Desarrollo Avanzado de Apps para Android 5 Lollipop

Content Provider

En este caso práctico se trabajarán los conceptos aprendidos en el tema 4 "Componentes de una App III – Content Provider" del material didáctico proporcionado por *net**mind***.

La metodología a seguir consistirá en realizar, conjuntamente con el formador, una aplicación Android que incluya un Content Provider que envuelva una base de datos con información relevante. Asimismo, se implementarán diversas funcionalidades asociadas a este tipo de elemento (Operaciones CRUD),

Objetivo

Presentar, examinar, estudiar y aprender a utilizar con soltura los Content Provider. Analizar las situaciones en las que es conveniente usar este tipo de elementos y entender el procedimiento que permite la comunicación ágil y precisa entre un Content Provider con la base de datos relacional que alberga y con otras aplicaciones externas.

Contenidos

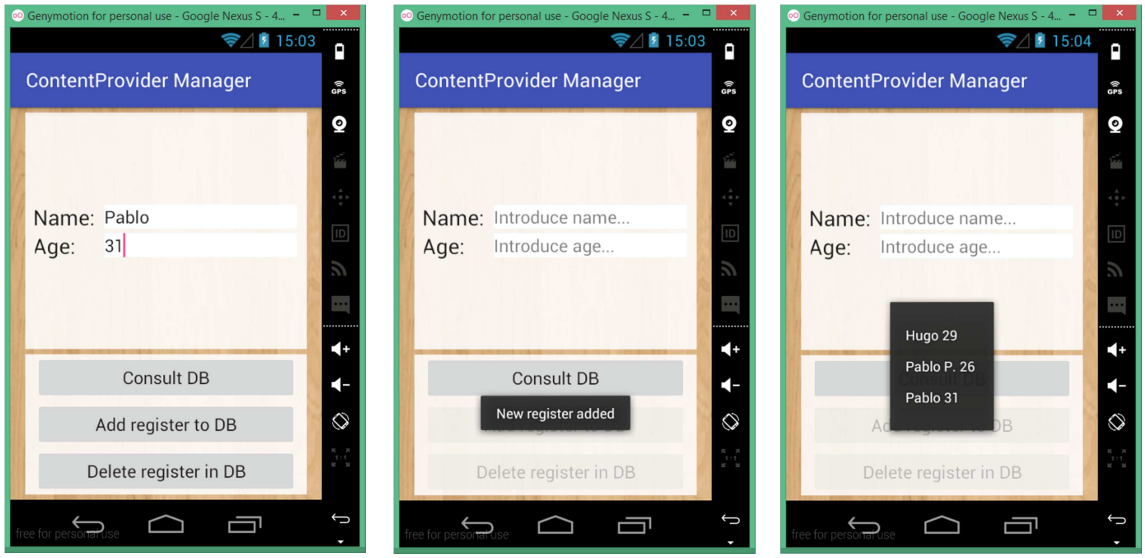
- Implementación de la clase ContentProvider y el elemento <provider>
- Métodos CRUD para un ContentProvider y diseño de URIs de contenido
- Acceso a un provider desde otra aplicación

Planteamiento del caso

La aplicación a implementar consta de un layout con campos de texto y botones que permiten al usuario introducir, eliminar y listar registros de una base de datos relacional a través de un Content Provider.

Para facilitar la confección de la aplicación, el formador facilitará al alumno el fichero XML del layout para así centrar el caso práctico en la definición de los diferentes métodos implicados en la implementación del Content Provider (operaciones CRUD).

Resultado final



Soluciones

El proyecto completo para Android Studio puede encontrarse en la carpeta "ContentProviderManager" o en el siguiente enlace de repositorio de GitHub:

https://github.com/pablodeafsapps/Netmind-JJD311_AndroidProjects/tree/master/ContentProviderManager

Al finalizar la práctica has aprendido…

En este caso práctico se ha presentado y explicado el último de los componentes de aplicación, el *Content Provider*. Un Content Provider es un elemento que hace las veces de interfaz entre una fuente de datos (una base de datos, por ejemplo) y la aplicación, y que permite que este repositorio sea accedido por cualquier aplicación presente en el terminal.

Práctica 5 Desarrollo Avanzado de Apps para Android 5 Lollipop

Tareas en background

En este caso práctico se trabajarán los conceptos aprendidos en el tema 5 "Gestión de tareas en *background*" del material didáctico proporcionado por *net**mind***.

La metodología a seguir consistirá en realizar, conjuntamente con el formador, una aplicación Android que incluya varias tareas de procesamiento en hilos distintos al principal (*UI thread* o *main thread*). Esta posibilidad dentro de las prestaciones ofrecidas por la plataforma Android permite dividir la carga de trabajo de una aplicación en diferentes módulos o bloques que interactúan paralelamente.

Objetivo

Conocer, estudiar y emplear con soltura las alternativas que ofrece la plataforma Android a la hora de paralelizar la ejecución de tareas. El alumno debe ser consciente de las limitaciones que un dispositivo móvil presenta (aunque cada vez menos) debidas a sus "limitadas" prestaciones hardware. Dado que el usuario interactúa constantemente con la aplicación, se hace necesario procesar cualquier tarea "pesada" de forma independiente. De no hacerlo así, la experiencia de usuario se resentirá considerablemente.

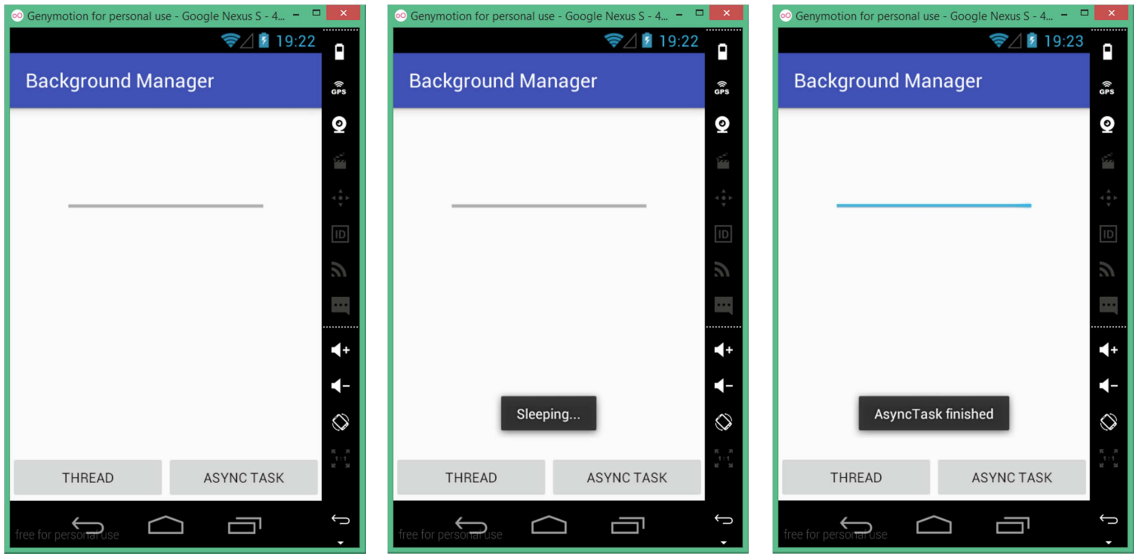
Contenidos

- Análisis y descripción de las alternativas para procesamiento *multihilos* en Android.
- Threads
- AsyncTasks

Planteamiento del caso

La aplicación a implementar presenta diferentes alternativas para ejecutar tareas "pesadas" de forma paralela. En concreto se utilizarán un Thread y un AsyncTask.

Resultado final



Soluciones

El proyecto completo para Android Studio puede encontrarse en la carpeta "BackgroundManager" o en el siguiente enlace de repositorio de GitHub:

https://github.com/pablodeafsapps/Netmind-JJD311_AndroidProjects/tree/master/BackgroundManager

Al finalizar la práctica has aprendido…

En este caso práctico se han presentado y explicado las distintas alternativas ofrecidas por la plataforma Android para ejecutar *tareas en background*, en hilos de procesamiento distintos al principal.

La ejecución de tareas en *worker threads* es además **un requerimiento de la plataforma** a la hora de implementar alguna de las funcionalidades típicamente presentes en una aplicación, como las operaciones de red.

Práctica 6 Desarrollo Avanzado de Apps para Android 5 Lollipop

Networking en Android

En este caso práctico se trabajarán los conceptos aprendidos en el tema 6 "Networking en Android" del material didáctico proporcionado por *net**mind***.

La metodología a seguir consistirá en realizar, conjuntamente con el formador, una aplicación Android que incluya tareas que requieran acceso a Internet. La idea es ilustrar las diferentes posibilidades que ofrece la plataforma Android a la hora de consumir datos adquiridos desde ubicaciones remotas.

Objetivo

Conocer, estudiar y aprender a implementar con soltura tareas en red (networking) en Android. Analizar las situaciones en las que es conveniente usar este tipo de procedimientos y entender la correcta gestión de las posibles excepciones en las que se puede incurrir.

Contenidos

- Chequeo de la conexión de red y ejecución en un *worker thread*
- Conexión y descarga de datos
- *Parseo* de datos XML para la lectura de un *feed*

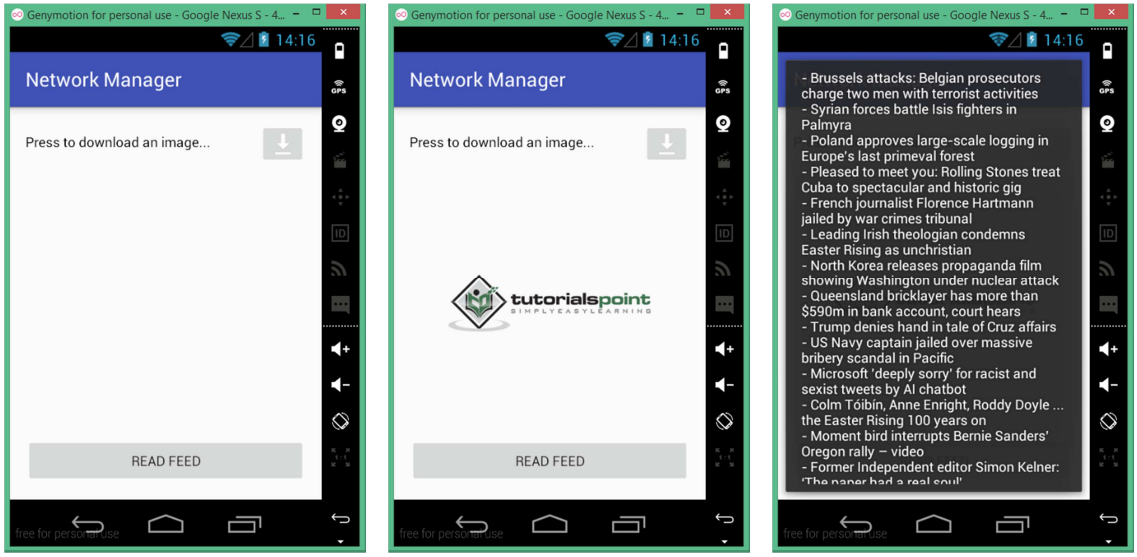
Planteamiento del caso

La aplicación a implementar se divide en dos secciones:

- En la primera el alumno debe desarrollar un código para permitir que la aplicación se descargue una imagen desde una URL y la ubique en el centro de la pantalla.
- En la segunda se debe *parsear* un *feed* de un periódico o blog para mostrar en pantalla información asociada de interés para el usuario.

Ambas tareas conllevan operaciones de networking que, por la propia naturaleza del sistema operativo Android, requieren ser ejecutadas en un hilo diferente al principal.

Resultado final



Soluciones

El proyecto completo para Android Studio puede encontrarse en la carpeta "NetworkManager" o en el siguiente enlace de repositorio de GitHub:

https://github.com/pablodeafsapps/Netmind-JJD311_AndroidProjects/tree/master/NetworkManager

Al finalizar la práctica has aprendido…

En este caso práctico se han presentado y explicado los conceptos relativos a *networking y operaciones de red* en Android, que permiten que una aplicación se comunique con el mundo exterior y acceda a contenidos y servicios de terceros. El hecho de que una aplicación integre funcionalidades de red hace que aumente su versatilidad y popularidad. Hoy día el «mundo Android» no se entiende sin Internet.

Práctica 7 Desarrollo Avanzado de Apps para Android 5 Lollipop

Programación Multimedia

En este caso práctico se trabajarán los conceptos aprendidos en el tema 7 "Programación Multimedia" del material didáctico proporcionado por *net**mind***.

La metodología a seguir consistirá en realizar, conjuntamente con el formador, una aplicación Android que permita la reproducción de un archivo de audio siguiendo diferentes procedimientos.

Objetivo

Conocer, estudiar y emplear con soltura la clase `MediaPlayer`, que es la responsable de permitir la reproducción y la grabación de contenidos multimedia en la plataforma Android. Analizar las situaciones en las que es conveniente usar este tipo de procedimientos y examinar la idoneidad de combinarlos con Services y Broadcast Receivers para agregar funcionalidades adicionales a la aplicación.

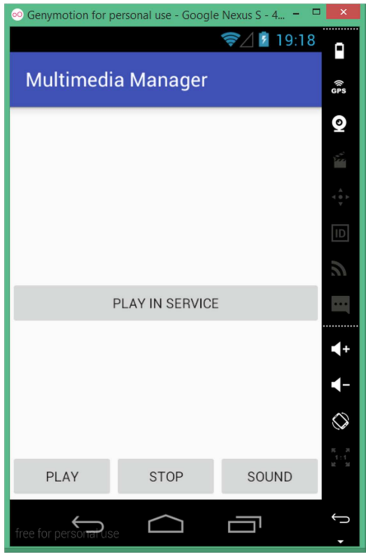
Contenidos

- La clase MediaPlayer
- MediaPlayer funcionando en *background*
- Cómo reaccionar al intent *AUDIO_BECOMING_NOISY*

Planteamiento del caso

La aplicación a implementar únicamente incluye una Activity cuyo *layout* alberga una serie de botones repartidos ordenadamente en la pantalla. En total se exploran dos formas distintas de reproducir contenidos multimedia (audio en este caso): en el hilo principal y en un *worker thread*. El fichero de audio será incluido como un recurso de tipo *raw* en el proyecto.

Resultado final



Soluciones

El proyecto completo para Android Studio puede encontrarse en la carpeta "MultimediaManager" o en el siguiente enlace de repositorio de GitHub:

https://github.com/pablodeafsapps/Netmind-JJD311_AndroidProjects/tree/master/MultimediaManager

Al finalizar la práctica has aprendido…

En este caso práctico se ha presentado y explicado la *API MediaPlayer* de Android, que permite de forma rápida y sencilla la reproducción y grabación de contenidos multimedia. Una de las principales consideraciones a hacer cuando se usa este framework, es que los dispositivos Android solo incluyen una salida de audio. Por ello, las aplicaciones deben «pelear» por este recurso y actuar en consecuencia.

Práctica 8 Desarrollo Avanzado de Apps para Android 5 Lollipop

Sensores

En este caso práctico se trabajarán los conceptos aprendidos en el tema 8 "Sensores" del material didáctico proporcionado por *net**mind***.

La metodología a seguir consistirá en realizar, conjuntamente con el formador, una aplicación Android que incluya el uso de sensores integrados en el terminal, permitiendo así la adquisición de datos ambientales del entorno y su posterior procesamiento.

Objetivo

Conocer, estudiar y emplear con soltura los Sensores en Android. Analizar las situaciones en las que es conveniente usar este tipo de elementos y entender el procedimiento que permite una adquisición de datos ágil y precisa en una aplicación.

Contenidos

- Tipos de sensores soportados en Android
- Identificación de sensores y sus capacidades
- Monitorización de eventos de sensores
- *Best practices* para el acceso y el uso de sensores

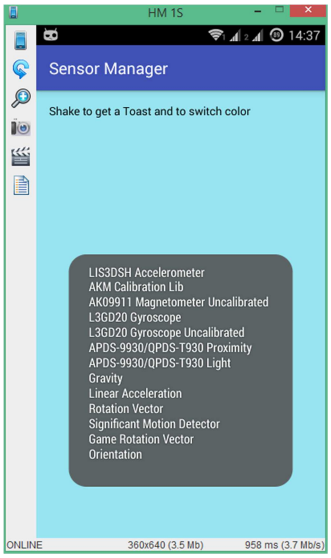
Planteamiento del caso

La aplicación a implementar requiere codificar dos tareas:

- Mostrar por pantalla (mediante un elemento Toast) un mensaje emergente con una lista de los sensores incluidos en el terminal en el que se ejecuta la aplicación.
- Utilizar los datos provenientes del sensor de aceleración (habitualmente incluido en cualquier terminal Android) para que ante un movimiento brusco del dispositivo se modifique alternativamente el color de fondo de la pantalla.

La idea es que la primera de las tareas se ejecute automáticamente al iniciarse la aplicación, mientras que la segunda requiera, lógicamente, la interacción del usuario.

Resultado final



Soluciones

El proyecto completo para Android Studio puede encontrarse en la carpeta "SensorManager" o en el siguiente enlace de repositorio de GitHub:

https://github.com/pablodeafsapps/Netmind-JJD311_AndroidProjects/tree/master/SensorManager

Al finalizar la práctica has aprendido…

En este caso práctico se ha presentado y explicado el *framework Sensor* de Android, que permite al desarrollador hacer uso de los sensores presentes en el terminal para la adquisición de datos ambientales. El desarrollador debe ser cuidadoso al ofrecer este tipo de prestación en la aplicación, ya que la variedad de dispositivos y configuraciones del «mundo Android» es enorme.

.

Práctica 9 Desarrollo Avanzado de Apps para Android 5 Lollipop

Mapas en Android

En este caso práctico se trabajarán los conceptos aprendidos en el tema 9 "Google API: Maps" del material didáctico proporcionado por *net**mind***.

La metodología a seguir consistirá en realizar, conjuntamente con el formador, una aplicación Android que incluya un mapa proporcionado por la API de Google.

Objetivo

Conocer, estudiar y emplear con soltura la API de mapas de Google, incorporando elementos habituales de n mapa como movimientos de cámara, controles de zoom, marcadores, etc.

Contenidos

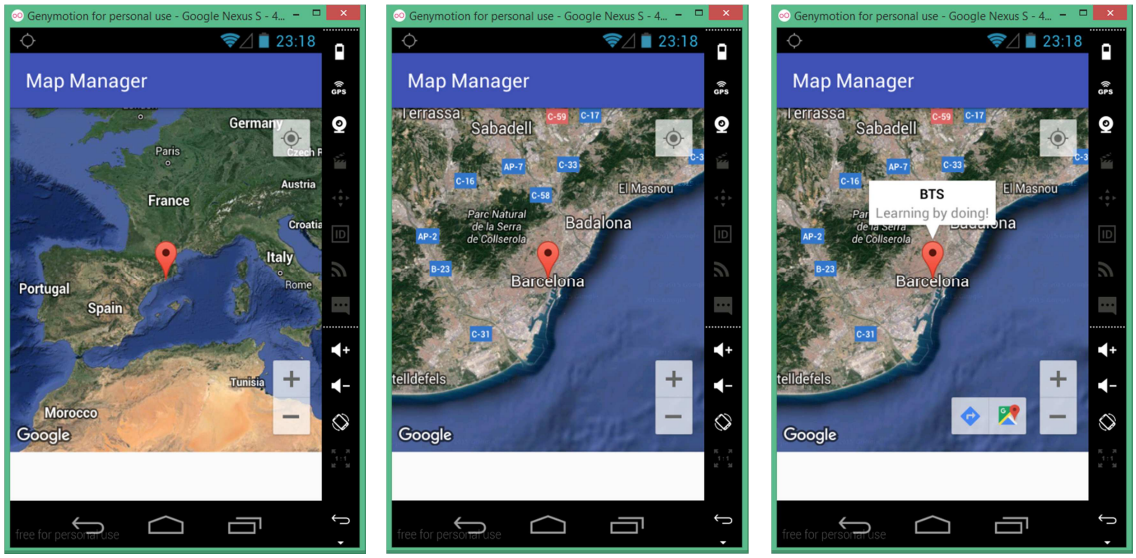
- La clase `MapFragment`
- La interfaz `OnMapReadyCallback()`
- *Markers* y *GoogleView*
- Obtención de un API key para *Google Maps*
- Habilitando Google APIs en el emulador *Genymotion*

Planteamiento del caso

La aplicación a implementar incluye un mapa sobre el que al pulsar se añade un marcador con un mensaje informativo. Además, se agregarán otras prestaciones que puedan mejorar la experiencia de usuario.

El alumno aprenderá previamente a configurar el acceso del programa a los servicios ofrecidos por Google a través de la *Google Developers Console* y la generación de la llamada API key.

Resultado final



Soluciones

El proyecto completo para Android Studio puede encontrarse en la carpeta "MapManager" o en el siguiente enlace de repositorio de GitHub:

https://github.com/pablodeafsapps/Netmind-JJD311_AndroidProjects/tree/master/MapManager

Al finalizar la práctica has aprendido…

En este caso práctico se ha presentado y explicado la *API Google Maps*, que permite a los desarrolladores integrar mapas en sus aplicaciones gracias al conocido servicio proporcionado por Google. La lección se centra en el uso del *Android Studio IDE*, ya que este entorno ha facilitado enormemente la gestión y manipulación de mapas. Hasta hace no tanto, el proceso podía llegar a ser una auténtica odisea.

Control de cambios del documento

Versión	Fecha	Descripción	Autor
V.1.0	16/05/2016	Contenidos originales	