Programación para Dispositivos Móviles(T.E.)

Ciclo I – 2019

Clase:004

Coordinador:Ing. César González

Android Studio

OBJECTIVE-C

Unidad II

Interfaz Grafica

Basado en:
http://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/

Agenda

- Desarrollo de Apps para Android.
  - Generalidades.
  - Codificación.
  - Componentes Básicos.
  - Intents.
  - Componentes del SDK.
  - Android código Abierto.

Agenda

- XML.
  - Introducción al XML.
  - Estructura de un documento XML.
  - Papel que desempeña.
  - Tipos.
  - DTD.
  - XML en ANDROID
  - Sitio para ampliar conceptos
- Activity.
- Demo Android

Desarrollo de Apps para Android
Generalidades

El desarrollo de aplicaciones Android se realiza con un grupo de herramientas que son suministradas en el SDK(software development kit).

La utilización de este grupo de herramientas puede ser de dos formas: utilizando un Entorno de Desarrollo Integrado (IDE) en combinación con un plugin llamado ADT (*Android Development Tools*, Herramientas de Desarrollo para Android) o bien desde la línea de comandos

Desarrollo de Apps para Android
Generalidades

Principales características que incluye Android Studio:

***Soporte** para programar aplicaciones para **Android Wear** (sistema operativo para dispositivos corporales como por ejemplo un reloj).

***Herramientas Lint** (detecta código no compatible entre arquitecturas diferentes o código confuso que no es capaz de controlar el compilador) para detectar problemas de rendimiento, usabilidad y compatibilidad de versiones.

Principales características que incluye Android Studio…

*Utiliza **ProGuard** para optimizar y reducir el código del proyecto al exportar a APK (muy útil para dispositivos de gama baja con limitaciones de memoria interna).

*Integración de la herramienta Gradle encargada de gestionar y automatizar la construcción de proyectos, como pueden ser las tareas de testing, compilación o empaquetado.

Principales características que incluye Android Studio…

*Posibilita el **control de versiones** accediendo a un repositorio desde el que poder descargar Mercurial, Git, Github o Subversion.

Vista previa en diferentes dispositivos y resoluciones.Integración con Google Cloud Platform, para el acceso a los diferentes servicios que proporciona Google en la nube.

Desarrollo de Apps para Android
Generalidades

Pasos básicos para desarrollar aplicaciones Android.

1)Instalación: en esta etapa se instala el entorno de desarrollo completo incluyendo el EDI y el SDK de Android, y se crean AVD (*Android Virtual Device*, Dispositivos Virtuales Android).

2)Desarrollo: en esta etapa se crea y desarrolla el proyecto Android, creando el código fuente de la aplicación y añadiendo todos los archivos fuentes que pueda necesitar como imágenes o demás recursos.

3) Depuración y pruebas: en esta etapa, en primer lugar se genera un paquete de depuración *.apk* que contiene el proyecto desarrollado en la etapa anterior. Este paquete se puede instalar y arrancar en cualquier emulador o teléfono Android. Si se usa Eclipse, cada vez que se guarda el proyecto automáticamente se genera el *.apk* correspondiente.
A continuación se depura la aplicación usando un depurador JDWP y las herramientas debug del SDK Android. Eclipse proporciona su propio depurador.

4) Publicación: en esta última etapa se configura y se construye la aplicación para generarse una versión *release* (una versión de entrega) para distribuir entre los usuarios.

Desarrollo de Apps para Android
Codificación

- Escritas en Java, pero no son aplicaciones Java
- El código fuente es precompilado a archivos .dex(Java precompila a .class)
- Layouts, configuraciones, menú y otros son declarados en XML. Los archivos XML también son precompilados para aumentar el desempeño .
- La implementación de la maquina virtual en Android es Dalvik.
- Las aplicaciones son empaquetadas en ficheros APK
- La VM provee protección al sistema operativo.
- Android 2.2 introdujo introduce compilación JIT para aumentar el rendimiento

Desarrollo de Apps para Android
Codificación

- Es creado un usuario de linux para cada aplicación
- Cada usuario creado para cada aplicación solo tiene los privilegios asignados(y solicitados) por la aplicación al momento de la instalación.
- Como las aplicaciones corren en un VM, no es posible interacción directa a otras aplicaciones o acceso a archivos restringidos(Sandbox)
- Los archivos APK deben de ser firmados digitalmente por el desarrollador. Los certificados son generados por medio de el SDK

Desarrollo de Apps para Android
Componentes Básicos

Activity

Representa una ventana (pantalla de la aplicación)

Service

Proveen ejecución de procesos en segundo plano

Broadcast Reciever

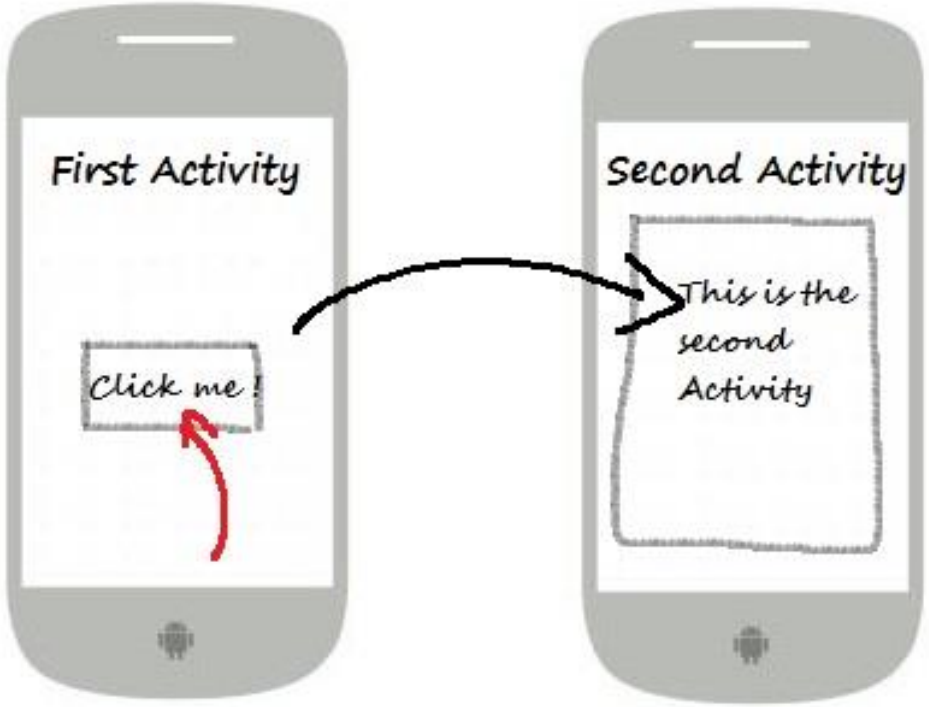
Escuchan eventos producidos desde el sistema o por otras aplicaciones

Content Providers

Proveen un medio de consultas y modificación de datos de otras aplicaciones

Desarrollo de Apps para Android
Intents

Son mensajes asíncronos, que permiten la comunicación entre aplicaciones y notificaciones al sistema operativo. Los intents permiten la sustitución de cualquier aplicación de Android.



Desarrollo de Apps para Android
Componentes del SDK

El SDK de Android esta compuesto de los siguientes Módulos:

- Documentación
- Herramientas de Desarrollo(ADB, AVD, AST etc.)
- Emuladores
- Ejemplos

Desarrollo de Apps para Android
Codigo Abierto

El sistema operativo es de código abierto por lo que cualquiera podría hacer su celular poner el sistema android xx modificado y utilizarlo

Sitio de descarga…
http://source.android.com/

Introducción al XML

SGML son las siglas de **Standard Generalized Markup Language** o "Estándar de Lenguaje de Marcado Generalizado". Consiste en un sistema para la organización y etiquetado de documentos. La Organización Internacional de Estándares (ISO) normalizó este lenguaje ISO 8879:1986, Information processing — Text and office systems — Standard Generalized Markup Language (SGML).

Introducción al XML

El lenguaje SGML sirve para especificar las reglas de etiquetado de documentos y no impone en sí ningún conjunto de etiquetas en especial. El lenguaje **HTML** está definido en términos del SGML. XML es un estándar de creación posterior, que incorpora un subconjunto de la funcionalidad del SGML (suficiente para las necesidades comunes), y resulta más sencillo de implementar pues evita algunas características avanzadas de SGML

Introducción al XML

XML, siglas en inglés de **Extensible Markup Language** ('lenguaje de marcas extensible'), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML).

Estructura de un documento XML

Ahora que conocemos como funciona HTML, observaremos que no existe mucha diferencia. La única diferencia es el contexto del uso de los dos lenguajes de marcado. XML es el lenguaje en donde el diseñador crea sus propias etiquetas y a partir de las cuales puede desarrollar nuevos estándares de lenguajes de marcado a través de lo que se denomina el **DTD.** El DTD hace la misma función que **DOCTYPE** dentro de HTML; es decir, realizar la respetiva validación según sea la DTD creada.

Papel que desempeña

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Tipos de documentos XML

Existen dos tipos de documentos XML, estos son:

1- Documento Bien formado o bien construido

Debe cumplir con una estructura correcta:

* Todas las etiquetas se cierran,
* Todos los atributos deben de estar entrecomillados.
* No deben haber etiquetas traslapadas
* Se deben respetar mayúsculas y minúsculas

LAS ETIQUETAS A UTILIZAR SON DEFINIDAS POR LA PERSONA QUE LO ELABORA. DICHA PERSONA DECIDE SI USA ETIQUETAS Y/O ATRIBUTOS SEGUN SU CONSIDERACION

Tipos de documentos XML

TODO DOCUMENTO XML DEBE POSEER UNA DECLARACION:

< ?xml version="1.0" encoding="utf-8" standalone="yes"?>

2- Documento Valido

Básicamente un documento XML está formado por elementos y sus atributos. El documento XML valido es el que está bien formado y además cumple con una especificación dada por la declaración de tipo de documento en relación a una DTD.

DTD

Una DTD debe ser capaz de definir todos los elementos de un documento, los atributos de cada uno de los elementos, y la relación entre los elementos.

Documento Bien Formado

```
<?xml version="1.0" encoding="UTF-8"?>
<EditMensaje>
      <Mensaje>
          <Remitente>
               <Nombre>Nombre del remitente</Nombre>
               <Mail> Correo del remitente </Mail>
          </Remitente>
          <Destinatario>
               <Nombre>Nombre del destinatario</Nombre>
               <Mail>Correo del destinatario</Mail>
          </Destinatario>
      </Mensaje>
</EditMensaje>
```

XML en ANDROID

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
              android:layout width="fill parent"
              android:layout height="fill parent"
              android:orientation="vertical" >
    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button" />
</LinearLayout>
```

Sitio para ampliar conceptos

Si se quiere profundizar podemos entrar el sitio

http://developer.android.com/guide/topics/ui/declaring-layout.html

Activities

- Necesita Pantallas – Interfaz grafica
- A cada actividad se le asigna una ventana en la cual dibujar su interfaz grafica
- Una actividad puede llenar una ventana o flotar en la ventana
- Una aplicación esta compuesta de varias actividades(Normalmente)
- Las actividades deben de ser declaradas en el AndroidManifest.xml, de lo contrario se produce un error en tiempo de ejecución.

Activities

Una Actividad debe de ser una subclase(directa o indirectamente) de la clase Activity

```
import android.app.Activity;

public class MainActivity extends Activity{

}
```

Activities

- View es un espacio rectangular dentro de una activity
- La clase View es la clase Base de los Widgets
- Los Widgets son los controles ya incluidos en el SDK para ser incluidos en las aplicaciones
- Widgets comunes:
  - TextView
  - Button
  - CheckBox
  - Etc.

Activities

Existen 2 formas de agregar views a una activity. Por medio de un layout en xml o programáticamente

Por layout, debemos de llamar al método setContentView() dentro de el método onCreate de la Activity

Activities

Sin referencia a objetos view

```
import android.app.Activity;
import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    public void onCreate(Bundle SavedInstance) {
    super.onCreate(SavedInstance);
    setContentView(R.layout.main);
    }

}
```

Activities

Android Studio

Con referencia a objetos view

```
import android.app.Activity;
import android.os.Bundle;
import android.widget.EditText;

public class MainActivity extends Activity {
EditText Text1;

@Override
public void onCreate(Bundle SavedInstance) {
super.onCreate(SavedInstance);
setContentView(R.layout.main);
Text1 = (EditText) findViewById(R.id.editText1);
Text1.setText("Hola Mundo!");
}

}
```