

## Laboratorio 7

### Introducción a Android Studio

**Tipo de trabajo:** ☒ Individual ☐ Grupal

**Fecha:** 11/12/2019

**Paralelo:** 101

**Nombre:** Bolivar Nuñez Montoya

**Objetivo:**

- Reconocer el entorno de trabajo del software Android Studio a través de una práctica básica para la introducción en el modelado, simulado y control de redes.

**Duración:**

Ciento veinte (120) minutos.

**Materiales y Herramientas:**

- Software Android Studio

**Introducción**

Android Studio es un IDE o entorno de desarrollo para la creación de aplicaciones Android basado en IntelliJ IDEA que ha nacido para sustituir al entorno usado casi exclusivamente hasta ahora, Eclipse con el plugin ADT(Android Developer Tools). Al igual que en Eclipse con el plugin ADT, Android Studio proporciona herramientas de desarrollo y depuración integradas para facilitar la tarea de los desarrolladores Android.

Además de las capacidades que dispone IntelliJ, Android Studio ofrece:

- Refactorización específica para Android y ayuda para dar soluciones rápidas a los errores que puedan surgir.
- Herramientas para capturar errores de rendimiento, facilidad de uso, compatibilidad con versiones antiguas y demás problemas.
- Ofuscación de código mediante Proguard.
- Firma de aplicaciones.
- Asistentes basados en plantillas para crear diseños y componentes comunes de Android.
- Un editor de diseño muy completo que permite, entre otras cosas, gestionar los componentes de interfaz de usuario mediante drag and drop (arrastrar y soltar) o vistas previas con múltiples configuraciones de pantalla.

**Procedimiento**

**Paso 1:** Iniciando el entorno Android Studio

- a) Iniciar el software Android Studio. En la nueva ventana seleccionar *“Start a new Android Studio project”*.
- b) En el campo *“Application name”* colocar laboratorio7TLM.
- c) En *“Project location”* escoger la ruta donde se guardará el proyecto y clic en *“Next”*.
- d) En la nueva ventana se escoge el SDK (Software Development Kit) mínimo que debe tener el dispositivo para poder ejecutar nuestra aplicación. Seleccionar API 18 y clic en *“Next”*.

- e) Aparecen las diferentes ventanas para la aplicación, seleccionar *“Empty Activity”* y clic en *“Next”*.
- f) En la siguiente ventana dejar los parámetros por defecto y clic en *“Finish”*.

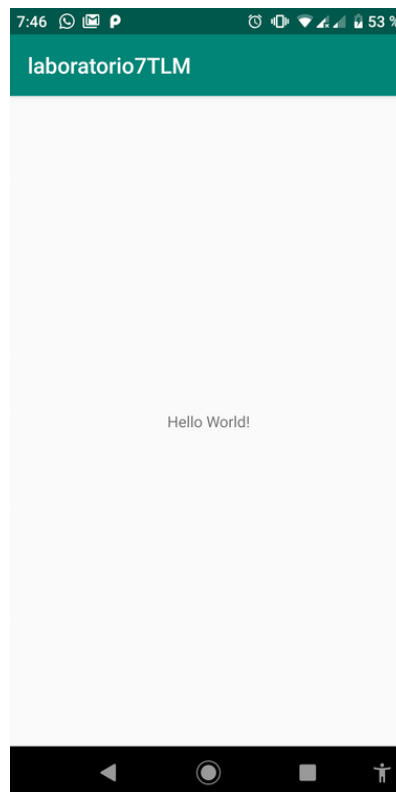


Al ejecutar el software, aparecerá el último proyecto en el que se ha estado trabajando, en caso de querer empezar un nuevo proyecto seleccionar en el menú principal *“File → Close Project”*

## **Paso 2:** Creación del dispositivo virtual para pruebas

- a) En el menú principal, seleccionar *“Tools → Android → AVD Manager (Android Virtual Device Manager)”*
- b) En la nueva ventana dar clic en *“Create Virtual Device...”*
- c) Se muestra una lista de dispositivos descargados con diferentes resoluciones de pantalla, seleccionar la opción Pixel de 5.0” y clic en *“Next”*.
- d) En la siguiente lista aparecen los sistemas operativos recomendados para trabajar, seleccionar la versión más reciente y clic en *“Next”*.
- e) En la ventana que aparece permite cambiar ciertas configuraciones del emulador, como la conectividad a Internet, la simulación de la cámara, entre otra; para la práctica dejar todas las opciones por defecto y dar clic en *“Finish”*.

**Actividad 1:** Presente una captura del dispositivo virtual ejecutando la aplicación “Hola Mundo” creada por defecto en el Paso 1.



**Paso 3:** Configuración de un dispositivo real para pruebas con Android Studio.

Dependiendo de la versión del sistema operativo, los pasos mencionados a continuación pueden cambiar ligeramente.

- Abrir la opción “*Configuración*” o “*Ajustes*” del celular.
- Seleccionar la opción “*Acerca del teléfono*”.
- Presionar 7 veces consecutivas el “*Número de compilación*”, aparecerá un mensaje indicando que ha sido habilitado las opciones de desarrollador.
- Al volver a la pantalla anterior, se podrá escoger “*Opción de desarrollador*”.
- En la sección “*Depuración*”, habilitar la opción “*Depuración mediante USB*”.

Los dispositivos con tecnología Android presentan diferentes opciones para el desarrollador a las que puedes acceder en el teléfono. Con ellas puedes hacer lo siguiente:



- Habilitar la depuración mediante USB.
- Obtener rápidamente informes de errores en el dispositivo.
- Mostrar el uso de la CPU en la pantalla.
- Mostrar información de depuración en la pantalla, como límites, actualizaciones en vistas de GPU y capas de hardware, y otros datos.
- Muchas más opciones para simular esfuerzos de la app o habilitar opciones de depuración.

**Paso 4:** Diseño de Formulario**Actividad 2:** ¿Qué es un Layout en Android Studio y para qué se utiliza?

Es una estructura visual que sirve como contenedor de los elementos tipo View que servirán para diseñar nuestra aplicación, permitiendo desarrollar la interfaz gráfica de una activity de nuestra aplicación.

- En el archivo “*activity\_main.xml*”, seleccionar la pestaña “*Design*” y crear el siguiente formulario:



El layout por defecto es “*ConstraintLayout*”, éste permite ubicar los elementos haciendo referencia a la ubicación de otro elemento.

**Actividad 3:** Investigue qué otros Layouts permite usar Android Studio y describa cada uno de ellos.

**Constraint Layout:** nos permite trabajar sin grupos de vistas anidadas similar a Relative Layout. Útil para trabajar con grandes cantidades de elementos ya que crea un orden jerárquico dinámico.

**Frame Layout:** Alinea todos los elementos de nuestro diseño al lado izquierdo.

**Linear Layout:** Alinea todos nuestros elementos de forma vertical u horizontal, la orientación debe ser definida en `android:orientation="orientacion"`.

**Coordinator Layout:** se usa para contener a los elementos que presenten animaciones e interacciones con otros elementos que tenga nuestra aplicación.

**Relative Layout:** Permite dar una ubicación específica de los elementos, los cuales dependen entre si para ubicarlos.

### Paso 5: Funcionalidad del Formulario

a) En el archivo “*MainActivity.java*”, editar el código para que quede de la siguiente forma:

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import android.content.Intent;
import android.net.Uri;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    EditText etnombres, etapellidos, etphone, etmail;
    Intent emailIntent = new Intent(Intent.ACTION_SEND);

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        etnombres = (EditText)findViewById(R.id.editTextNombres);
        etapellidos = (EditText)findViewById(R.id.editTextApellidos);
        etphone = (EditText)findViewById(R.id.editTextPhone);
        etmail = (EditText)findViewById(R.id.editTextMail);
    }

    public void enviar (View v){
        String[] to = {"cvaccaro@espol.edu.ec"};
        String[] cc = {"cvaccaro@fiiec.espol.edu.ec"};
        emailIntent.setData(Uri.parse("mailto:"));
        emailIntent.setType("text/plain");
        emailIntent.putExtra(Intent.EXTRA_EMAIL, to);
        emailIntent.putExtra(Intent.EXTRA_CC, cc);
        emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Formulario de Registro PST");

        emailIntent.putExtra(Intent.EXTRA_TEXT, "Datos de Contacto\n" +
```




```
        "Nombres:" + etnombres.getText().toString()+ "\n" +
        "Apellidos:" + etapellidos.getText().toString() + "\n" +
        "Teléfono:" + etphone.getText().toString() + "\n" +
        "Correo Electrónico:" + etmail.getText().toString()+ "\n");

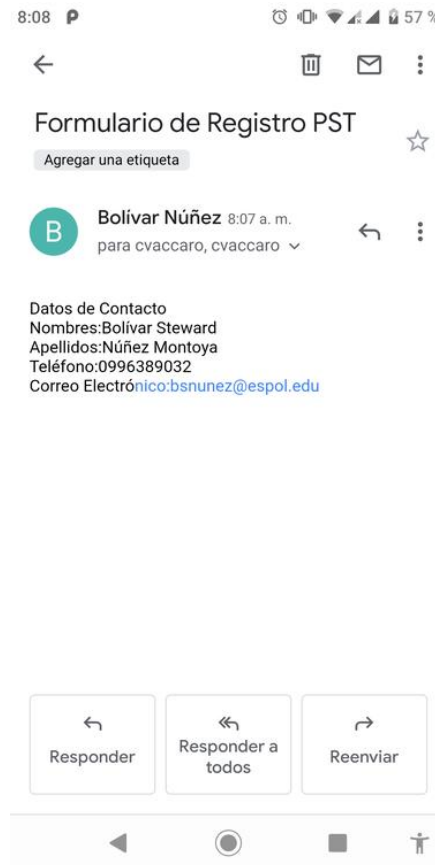
    try {
        startActivity(Intent.createChooser(emailIntent, "Enviando Email"));
        Log.i("termina envio de email", "");
    }
    catch (android.content.ActivityNotFoundException ex) {
        Toast.makeText(MainActivity.this, "No existe cliente Email
instalado.", Toast.LENGTH_SHORT).show();
    }
}
```



Asegúrese de que los nombres de los componentes creados en el Paso 4 sean iguales a los nombres usados en el Paso 5

#### **Paso 6:** Ejecución del Formulario

- a) En la barra de herramientas dé clic en la opción *“Run App”* , y seleccionar el dispositivo a ejecutar.
- b) Ingrese los datos en el formulario.
- c) Envíe los datos presionando el botón ENVIAR.
- d) Escoja una aplicación para enviar los datos
- e) Verifique el envío del correo



**RETO (10 puntos extras): A partir del formulario creado en el laboratorio, genere una nueva función para el botón la cual ya no abra una aplicación externa, sino que en la misma app muestre una nueva ventana de bienvenida con los campos del formulario dentro de un TextView (Similar a la app Hello World).**

### Conclusiones y Recomendaciones

#### Conclusiones;

- Existen diversos tipos de layout con los cuales podemos trabajar y que nos facilitan el desarrollo de nuestra interfaz grafica para una mejor apreciación de la data hacia el usuario.
- Los botones pueden implementar métodos del archivo java desde el archivo xml en donde se crea el método onClick "name del metodo". En el archivo java se puede controlar la acción del botón creando el método tipo View o llamando al onClickListener().
- Se logro enviar un correo con la información recopilada de los textView del archivo xml fácilmente mediante el método enviar.

#### Recomendaciones:

- Por lo general, la mayoría de los estudiantes empezaron desarrollando la interfaz grafica en el archivo xml con el layout que sale por default Constraint Layout, sin embargo, para facilitar el diseño del formulario con los textView fácilmente se pueden lograr con LinearLayout.
- Las API que brinda Android studio para ejecutar la aplicación, por lo general, no permiten realizar ciertas acciones como acceder al correo, por lo que es recomendable ejecutar en un smartphone con Android.
- Declarar las variables de los View en el archivo java con el mismo nombre o id que le colocamos en el archivo xml para una mayor comprensión del código y las variables no sean repetidas y redundantes.