Seminario de Lenguajes
Android
Práctica 1

1) Abrir Android Studio y crear un nuevo proyecto con una Actividad vacía (Empty Activity)

2) Abrir el Android Virtual Device Manager y crear como dispositivo virtual un Galaxy Nexus con el nombre Nexus Seminario Android

3) Probar la aplicación creada en el ejercicio 1 en el emulador

4) Describir qué representa una Activity

5) Abrir el archivo AndroidManifest.xml. ¿Por qué MainActivity tiene un intent-filter con action MAIN y category LAUNCHER?

6) Crear 2 actividades (NuevaActivity1, NuevaActivity2) y ver qué se modificó en el archivo AndroidManifest.xml

7) Pegar el siguiente código en res/layout/activity_main.xml

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

        <Button
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Click"
            android:onClick="onBtnClick"/>
</RelativeLayout>
```

8) Añadir el siguiente código en la clase MainActivity.java, probar en el emulador y analizar el resultado

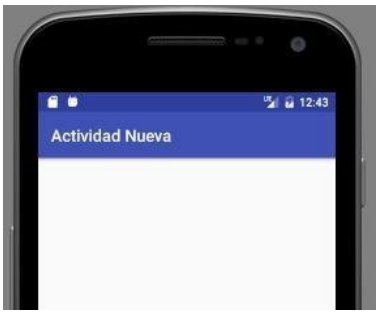
```
fun onBtnClick(view: View?) {
  val i = Intent(this, NuevaActivity2::class.java)
  startActivity(i)
}
```

* Se deben importar estos paquetes: *android.view.View*, *android.content.Intent*

9) ¿Qué significa *this* en el código del ejercicio anterior?

10) ¿Lo que utilizamos fue un intent implícito o explícito? ¿Cual es la diferencia entre ambos?

11) A través del AndroidManifest modificar el nombre que se muestra al usuario para la actividad 2 para que al hacer click se muestre con el texto *Actividad Nueva*



12) Modificar el comportamiento de onBtnClick para que a través de un intent abra una página web

13) En el método onCreate de la actividad nueva añadir la siguiente línea.

Log.*d*(**"APP_DE_PRUEBA"**, **"Este es mi mensaje de debug"**);

Correr la aplicación en modo debug y revisar la consola de Debug luego de abrir la actividad 2. ¿Qué ocurrió? ¿Cuál es su utilidad?

14) Al hacer click en el botón se debe pasar a la actividad 2 un texto como parámetro. Cuando la actividad 2 se muestra se debe imprimir por consola el texto recibido como parámetro.

15) Describa cuales son los estados por los que puede pasar una actividad

16) Describa cuales son los eventos generados a partir de un cambio de estado de una actividad

17) Crear una nueva aplicación en la que se imprima por la consola los cambios de estados de una actividad utilizando lo investigado en el punto 15 y 16

18) Genere una nueva aplicación con una actividad vacía en Android Studio. Edite el archivo AndroidManifest.xml y elimine las siguientes líneas:

```
<intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

a) ¿Qué error se produce en el entorno de desarrollo?. ¿Cuál es el motivo del error?
b) Vuelva a colocar el código eliminado para que la aplicación funcione correctamente
c) Agregue un TextView a la Activity generada con el valor "Español" y un botón con el valor 'Cambiar idioma. Al hacer click en el botón el textview debe cambiar su texto a "Inglés". Si se presiona nuevamente sobre el botón, el textview debe contener el valor "Español" nuevamente.

19) Crear una nueva aplicación con 2 actividades como se ven en la figura.

Al hacer click en "Realizar operación" se debe abrir la segunda actividad. Al hacer click en los botones "Incrementar" o "Decrementar" la actividad debe cerrarse y aplicar la operación correspondiente sobre el TextView. Si se presiona "Cancelar" solo debe cerrarse la segunda actividad sin realizar cambios sobre el TextView

A continuación se detallan los layouts de las activities para simplificar el ejercicio (Notar el uso de LinearLayout):

Activity1:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">


        <TextView android:id="@+id/txtContador"
                  android:layout_width="match_parent"
                  android:layout_height="wrap_content"
                  android:text="0"
                  android:textAlignment="center"
                  android:textSize="34sp"
            />

        <Button android:id="@+id/btnRealizarOperacion"
                  android:layout_width="match_parent"
                  android:layout_height="wrap_content"
                  android:text="Realizar operación"
             />
</LinearLayout>
```

Activity2:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">

        <Button android:id="@+id/btnIncrementar"
                  android:layout_width="wrap_content"
                  android:layout_height="wrap_content"
                  android:text="Incrementar"
            />

        <Button android:id="@+id/btnDecrementar"
                  android:layout_width="wrap_content"
                  android:layout_height="wrap_content"
                  android:text="Decrementar"
            />

        <Button android:id="@+id/btnCancelar"
                  android:layout_width="wrap_content"
                  android:layout_height="wrap_content"
                  android:text="Cancelar"
            />

</LinearLayout>
```

20) Agregar un control más al ejercicio anterior para que no pueda decrementarse si el valor es 0. Para ello al retornar a la actividad 1 verificar si el valor es 0 y desplegar un mensaje Toast informando el error.

21) Modifique el ejercicio anterior para que la segunda activity (operaciones) de abra con un intent implícito.

22) ¿Qué sucede en el ejercicio anterior si se modifica la orientación del dispositivo (horizontal/vertical)?
    a) Solucione el problema mediante saveInstanceState / restoreInstanceState.

23) Generar una actividad con nombre LifeCycleActivity y pruebe el siguiente código:

```
override fun onDestroy() {
    super.onDestroy()
    val i = Intent(this, LifeCycleActivity::class.java)
    this.startActivity(i)
}
```

    Ejecute la aplicación:
    a) Intente destruir la actividad mediante el botón "Atrás" del dispositivo.
    b) Intente destruir la actividad mediante el botón de intercambio de tareas. (Botón central del Nexus S)

24) En el ejercicio anterior, agregue a la actividad un TextView con id "texto".

    Agregue al método onDestroy el siguiente código:

```
override fun onDestroy() {
    super.onDestroy()
    (findViewById<View>(R.id.texto) as TextView).text = "HOLA MUNDO!"
    val i = Intent(this, LifeCycleActivity::class.java)
    this.startActivity(i)
}
```

    Intente destruir la actividad mediante el botón "Atrás" del dispositivo.
    a) ¿Se ve el mensaje "HOLA MUNDO" en la componente TextView?. ¿Por qué?
    b) ¿Se puede resolver mediante saveInstanceState / restoreInstanceState?
    c) ¿Qué sucede con la instancia de LifeCycleActivity?

25) Crear una nueva aplicación que tenga 2 actividades. En la actividad 1 reemplazar el código del archivo *activity_main.xml* por el presentado a continuación.

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
        <!-- Aquí van los botones →
</LinearLayout>
```

La primera actividad debe tener 2 botones. El primero debe tener el texto ¿Qué hora es? y al hacer click debe imprimir en la consola de Debug la hora actual.

El segundo debe contener el texto ¿Qué día es? y al hacer click pasar como parámetro el día de hoy a la 2da actividad. Al abrirse la segunda actividad debe imprimir el valor recibido.

** Investigar la clase SimpleDateFormat para convertir la fecha a String en un formato específico*