

PRACTICA 07: Creación de nuevas Actividades

CENTRO DE CIENCIAS BASICAS

I.S.C.

9º. SEMESTRE, GRUPO B, C

OPTATIVA, (PROGRAMACION DE MOVILES)

PROFESOR: M.C. MIGUEL ANGEL ORTIZ ESPARZA.

Octubre 2018.

Hoy en día cualquier sistema de información requiere de una serie de pantallas y transiciones entre ellas, mismas que hacen posible la buena navegación y por consiguiente la obtención del objetivo dentro de un sistema.

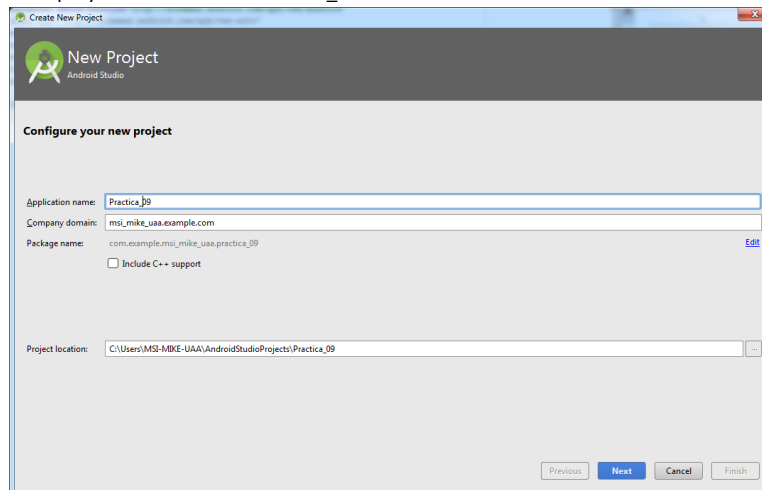
En los dispositivos móviles esto no es la excepción, se requiere de una serie de pantallas que necesitan orientar al buen uso de la aplicación, cada una de estas pantallas orienta la serie de interacciones que se deben tener en cada una de ellas, evitando aglomeraciones de datos y mostrando de una forma específica cada uno de los contextos requeridos

Objetivo.

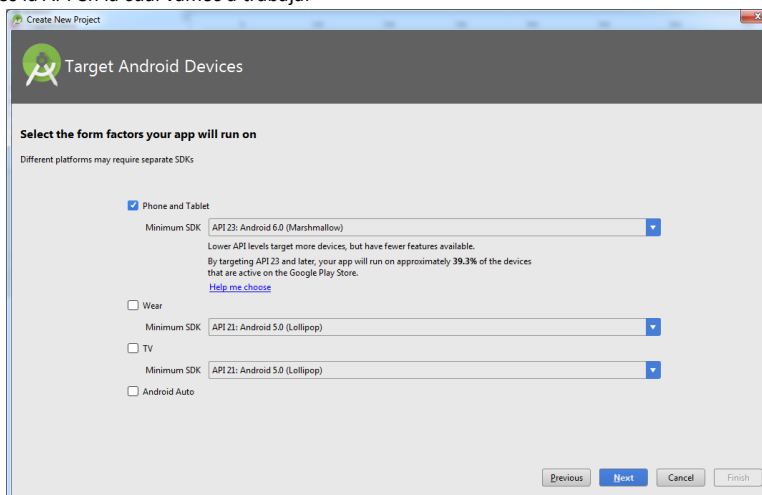
Conocer la importancia de los intents y la creación de más de un Layout para poder trabajar.

Procedimiento.

1. Creamos un nuevo proyecto denominado Práctica_07

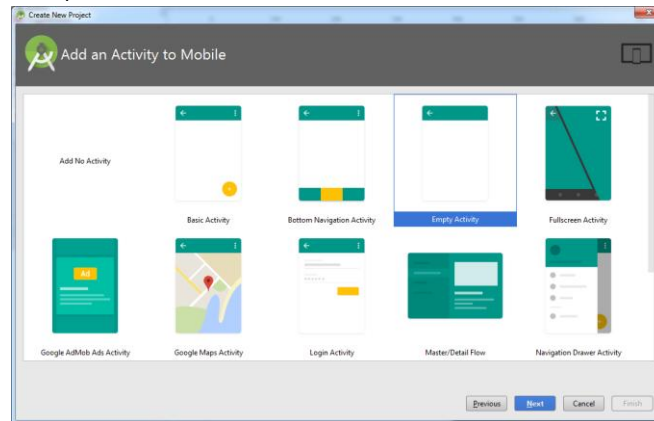


2. Seleccionamos la API en la cual vamos a trabajar

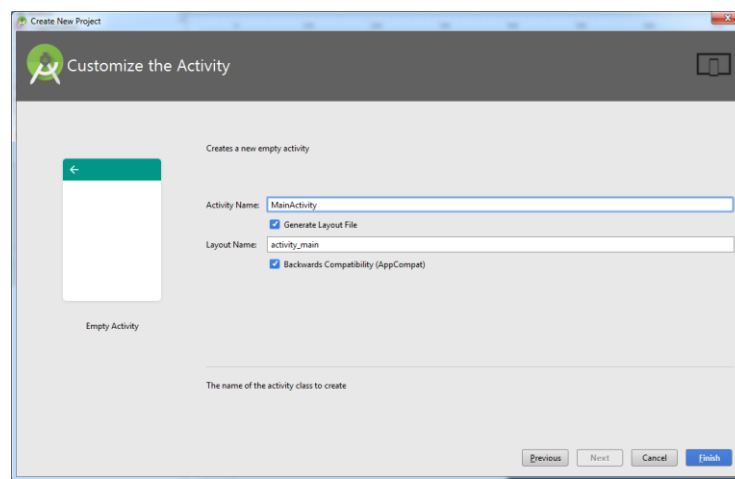


Universidad Autónoma de Aguascalientes.

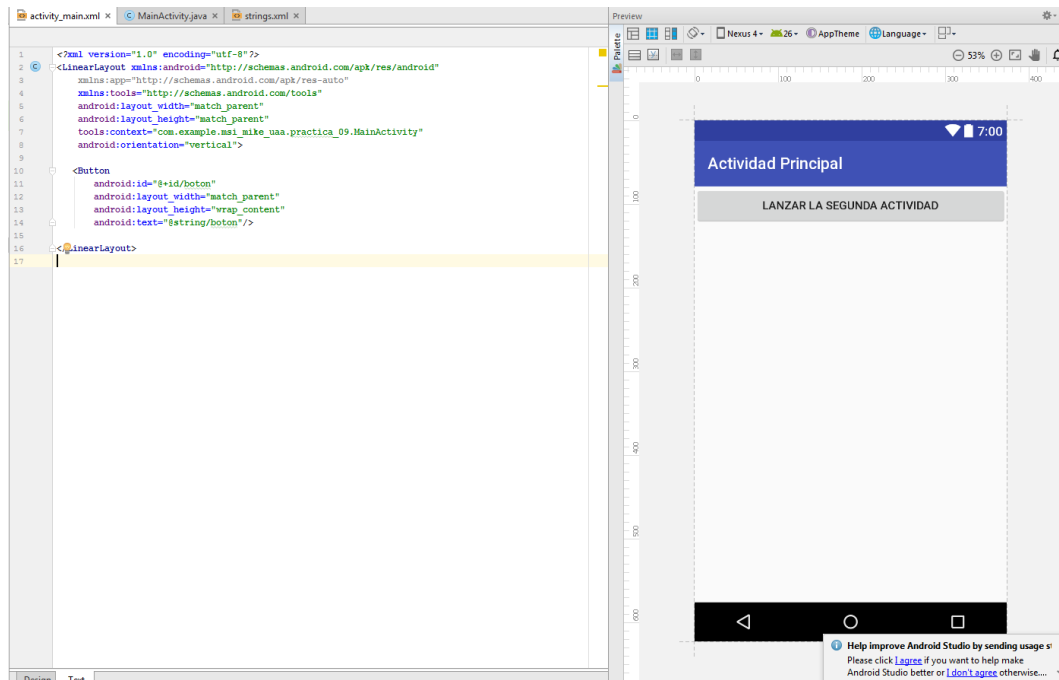
3. Seleccionamos una Activity vacía



4. Personalizamos la Activity

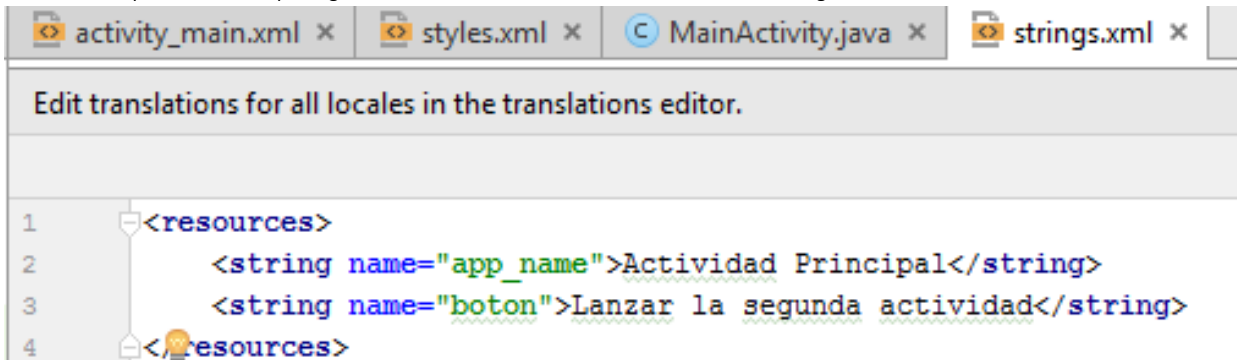


5. Una vez creada la Activity, pasamos al diseño de la interfaz en el XML, donde agregamos un botón con el id = "boton" como se muestra a continuación.



Universidad Autónoma de Aguascalientes.

6. Después de crear la parte grafica hacemos unas modificaciones en el archivo strings.xml

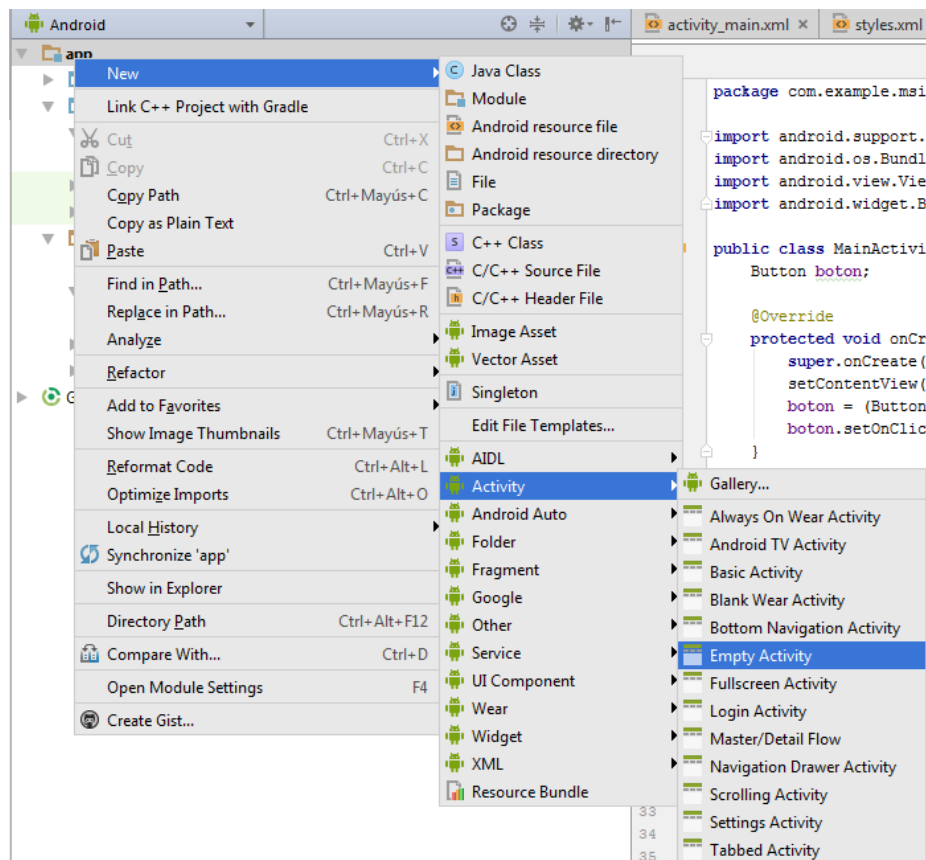


7. Ahora como hicimos en la práctica número 6, vamos a agregar un botón en el archivo MainActivity.java y después implementamos en OnClickListener

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener{
    Button boton;

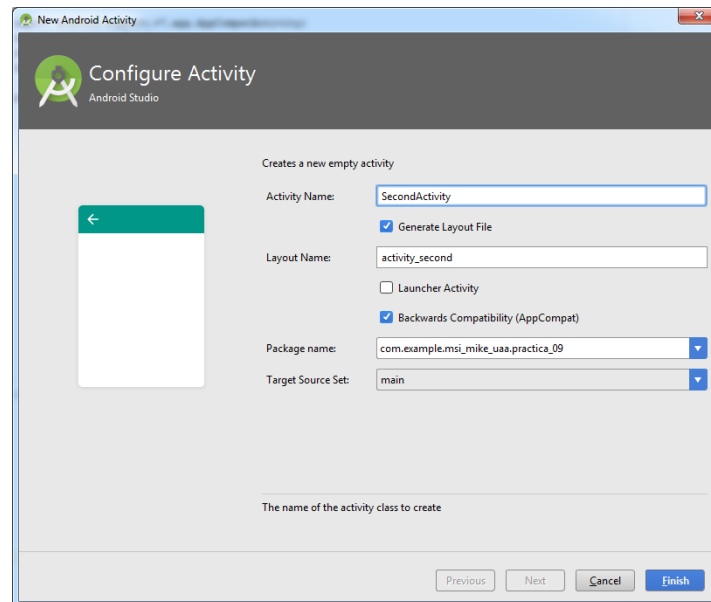
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        boton = (Button) findViewById(R.id.boton);
        boton.setOnClickListener(this);
    }
}
```

8. Ahora crearemos una nueva actividad para mandarla llamar cuando demos click en el botón anteriormente creado, para ello necesitamos realizar lo siguiente.
9. En el explorador del proyecto que tenemos en la parte izquierda de la pantalla, en donde dice **app**, con el mouse presionamos botón derecho y agregamos una "Empty Activity" como se muestra a continuación. Otra forma de hacer esto es por medio del menú File -> New -> Activity -> "Empty Activity"

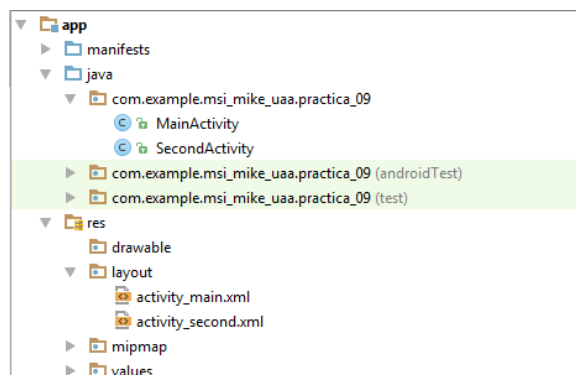


Universidad Autónoma de Aguascalientes.

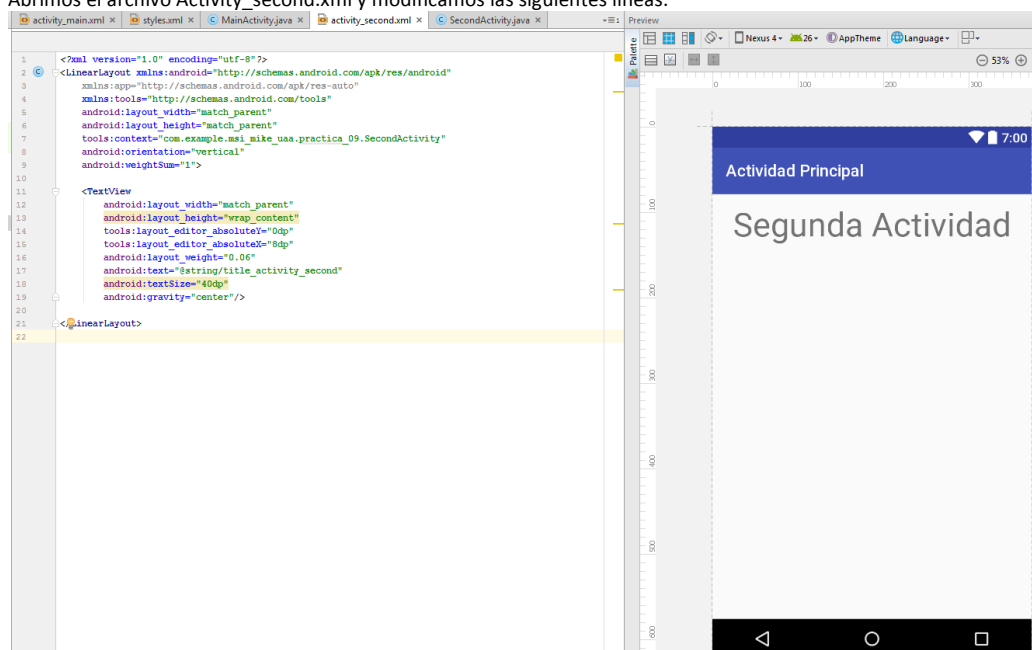
- Al seleccionar esta opción, el IDE nos muestra la siguiente pantalla, donde colocamos el nombre de nuestra nueva actividad, la cual nombraremos SecondActivity, como se muestra a continuación.



- Presionamos el botón Finish, la cual nos agregará un archivo XML y un archivo Java en el árbol de exploración como se muestra a continuación.



- Abrimos el archivo Activity_second.xml y modificamos las siguientes líneas.



13. Ahora modificamos en el AndroidManifest.xml, agregamos las siguientes líneas en el <activity> de la .SecondActivity como se muestra a continuación, colocando en Android.intent.category.DEFAULT, para que no la tome como principal, sino como actividades secundarias, en caso de agregar varias actividades llevarían estas características.

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

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

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

14. Por último, en el evento Onclick (View v) vamos a estar escuchando los clicks que se den a este elemento, cada vez que se presione el botón generará un elemento Intent, el cual tratará de abrir la nueva actividad, en la cual se pasaran dos parámetros, la actividad actual (MainActivity.this) y como segundo parámetro ponemos la actividad hacia donde queremos ir (SecondActivity.class).

Para esto vamos a agregar las siguientes líneas en el switch.

```
@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.boton:
            Intent siguiente = new Intent(MainActivity.this, SecondActivity.class);
            startActivity(siguiente);
            break;
    }
}
```

15. Como parte final checamos la aplicación corra y mande llamar la segunda actividad.

16. El código final quedo de la siguiente forma.

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    Button boton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        boton = (Button) findViewById(R.id.boton);
        boton.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.boton:
                Intent siguiente = new Intent(MainActivity.this, SecondActivity.class);
                startActivity(siguiente);
                break;
        }
    }
}
```