

ANDROID NIVEL I

LABORATORIO 05

ACTIVIDAD	:	Practica de Laboratorio Nro. 05
TEMA	:	ListView
DOCENTE	:	Ing. Eric Gustavo Coronel Castillo

OBJETIVO	2
DESCRIPCIÓN	2
DESARROLLO	2
PASO 01: CREACIÓN DEL PROYECTO.....	2
PASO 02: DISEÑO DE LA INTERFAZ DE USUARIO.....	3
PASO 03: DECLARACIÓN DE VARIABLES	3
PASO 03: INICIALIZANDO LA INTERFAZ DE USUARIO	4
PASO 04: CARGANDO ELEMENTOS AL LISTVIEW	4
PASO 05: ACTIVIDAD ACTIVIDAD_DIALOGO.XML	5
PASO 05: MENÚ DE OPCIONES	6
PASO 06: PROGRAMACIÓN DEL MENÚ.....	7
PASO 07: PROGRAMAR EL MÉTODO AGREGARNUEVOELEMENTO	8
PASO 08: PROGRAMAR EL MÉTODO ELIMINARELEMENTOACTUAL	9

OBJETIVO

En este laboratorio utilizaremos ListView para manejar una colección de datos.

DESCRIPCIÓN

Se desarrollara una aplicación que muestre una lista de ciudades, luego permitirá agregar nuevas ciudades a la lista o eliminar alguna de ellas.

DESARROLLO

Paso 01: Creación del Proyecto

Crear un nuevo proyecto con las siguientes especificaciones iniciales:

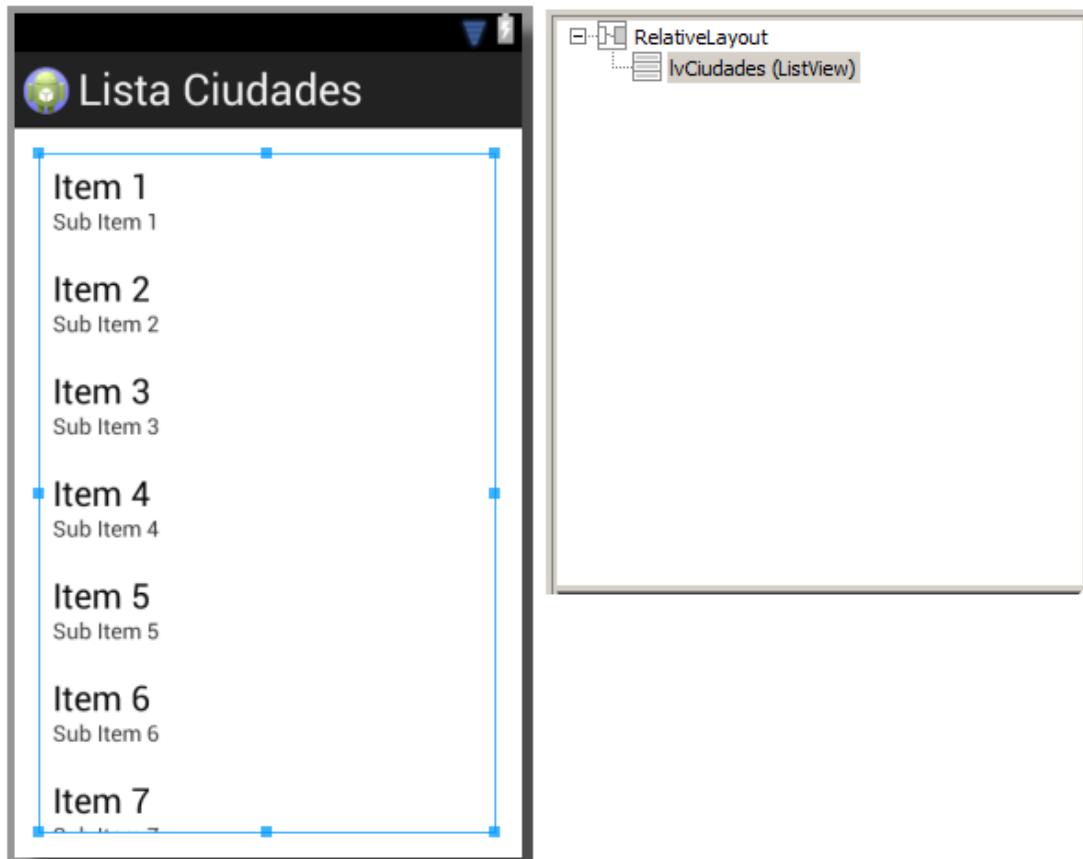
Application Name	Practica05
Company Domain	egcc.uss.pe
Project location	D:\UNI_SIPAN\Ejercicios\Practica05
Minimun SDK	API 16: Android 4.1 (Jelly Bean)

El proyecto debe tener una actividad en blanco con las siguientes especificaciones:

Activity Name	ActividadLista
Layout Name	actividad_lista
Title	LISTA DE CIUDADES

Paso 02: Diseño de la Interfaz de Usuario

A continuación se muestra una imagen con la interfaz de usuario que se debe diseñar:



En la imagen anterior se puede observar el ID que debe asignar al control ListView.

Paso 03: Declaración de variables

En la clase ActividadMate declare las siguientes variables:

```
// Variables
private ListView lvCiudades;
private List<String> ciudades;
private String mensaje;
private int p;
private String ciudad;
private ArrayAdapter<String> adapter;
```

Paso 03: Inicializando la interfaz de usuario

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.actividad_mate);
    llenarLista();
}

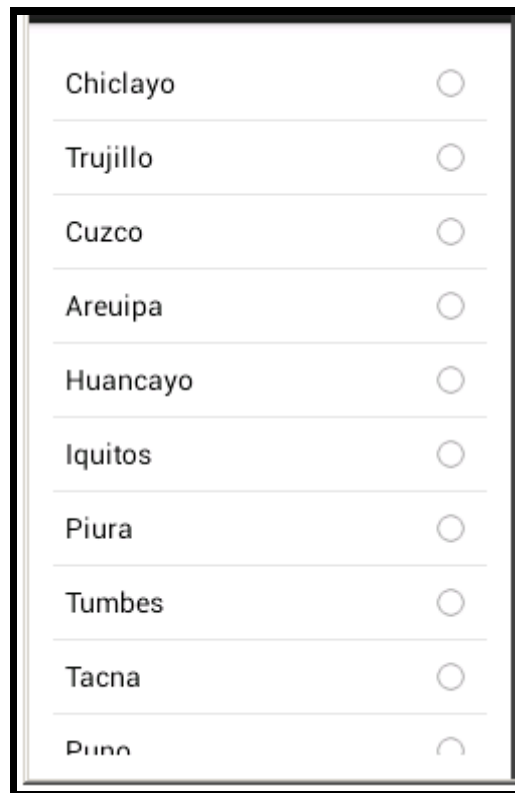
private void llenarLista() {
    ciudades = new ArrayList<String>();
    ciudades.add("Chiclayo");
    ciudades.add("Trujillo");
    ciudades.add("Cuzco");
    ciudades.add("Areuipa");
    ciudades.add("Huancayo");
    ciudades.add("Iquitos");
    ciudades.add("Piura");
    ciudades.add("Tumbes");
    ciudades.add("Tacna");
    ciudades.add("Puno");
}
```

Paso 04: Cargando elementos al ListView

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.actividad_mate);
    llenarLista();
    mostrarLista();
}

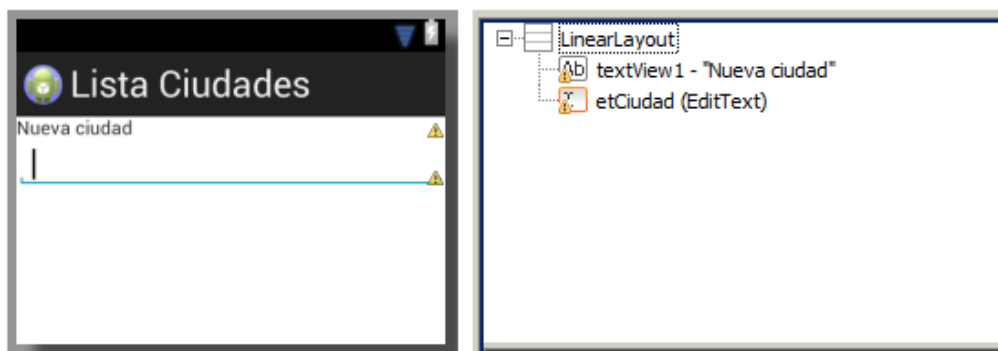
private void mostrarLista() {
    lvCiudades = (ListView) findViewById(R.id.lvCiudades);
    adapter = new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_single_choice,
        android.R.id.text1, ciudades);
    lvCiudades.setAdapter(adapter);
    lvCiudades.setChoiceMode(ListView.CHOICE_MODE_SINGLE);
}
```

En este momento ya se puede ejecutar la aplicación, el resultado será similar al que se muestra en la siguiente imagen:



Paso 05: Actividad actividad_dialogo.xml

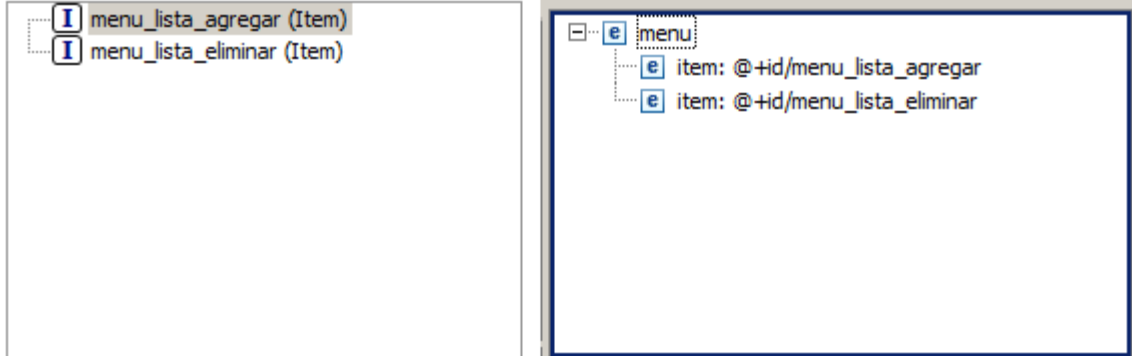
Esta actividad servirá para mostrar un dialogo que permita ingresar nuevas ciudades a la lista, a continuación tenemos la interfaz de usuario:





Paso 05: Menú de opciones

Defina el siguiente menú:



El XML es el siguiente:

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >

    <item android:id="@+id/menu_lista_agregar"
        android:title="@string/menu_lista_agregar"></item>

    <item android:id="@+id/menu_lista_eliminar"
        android:title="@string/menu_lista_eliminar"></item>

</menu>
```



Paso 06: Programación del menú

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.actividad_lista, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.menu_lista_eliminar:
            eliminarElementoActual();
            break;
        case R.id.menu_lista_agregar:
            agregarNuevoElemento();
            break;
    }
    return super.onOptionsItemSelected(item);
}

private void agregarNuevoElemento() {
    // Falta
}

private void eliminarElementoActual() {
    // Falta
}
```



Paso 07: Programar el método agregarNuevoElemento

```
private void agregarNuevoElemento() {
    LayoutInflater activador = LayoutInflater.from(this);
    final View entrada = activador.inflate(R.layout.actividad_dialogo, null);
    AlertDialog.Builder adb = new AlertDialog.Builder(this);
    adb.setTitle("Nueva ciudad");
    adb.setView(entrada);
    adb.setPositiveButton("Agregar", new OnClickListener() {

        @Override
        public void onClick(DialogInterface p_dialog, int p_which) {
            EditText caja = (EditText) entrada.findViewById(R.id.etCiudad);
            ciudades.add(caja.getText().toString());
            adapter.notifyDataSetChanged();
        }

    });
    adb.setNegativeButton("No", null);
    adb.show();
}
```

Ejecute la aplicación para probar su funcionalidad.



Paso 08: Programar el método eliminarElementoActual

```
private void eliminarElementoActual() {
    p = lvCiudades.getCheckedItemPosition();
    mensaje = "";
    if (!ciudades.isEmpty() && p == -1) {
        mensaje = "Seleccione un elemento";
    }
    else if (ciudades.isEmpty()) {
        mensaje = "La lista está vacía";
    }
    else {
        ciudad = lvCiudades.getItemAtPosition(p).toString();
        AlertDialog.Builder adb = new AlertDialog.Builder(this);
        adb.setTitle("Confirmación");
        adb.setMessage("¿Está seguro de eliminar " + ciudad + "?");
        adb.setCancelable(false);
        adb.setPositiveButton("Si", new OnClickListener() {

            public void onClick(DialogInterface dialog, int which) {
                adapter.remove(adapter.getItem(p));
                mensaje = "Posición: " + p + " Ciudad: "
                    + ciudad + " fué eliminado.";
            }

        });
        adb.setNegativeButton("Cancelar", null);
        adb.show();
    }
    Toast.makeText(getApplicationContext(), mensaje, Toast.LENGTH_SHORT).show();
}
```

Finalmente prueba la aplicación.