



FACULTAD DE INFORMÁTICA Y CIENCIAS APLICADAS

ESCUELA DE INFORMÁTICA



ASIGNATURA: Electiva 1

SECCION:01

Documento:

Ejercicio para evaluación 4

TRABAJO PRESENTADO POR:

INTEGRANTES

NOMBRES	CARNET
Perdomo Aragón, Roberto Esaú	29-4645-2013

25 de septiembre de 2020

Capturas Aplicación móvil

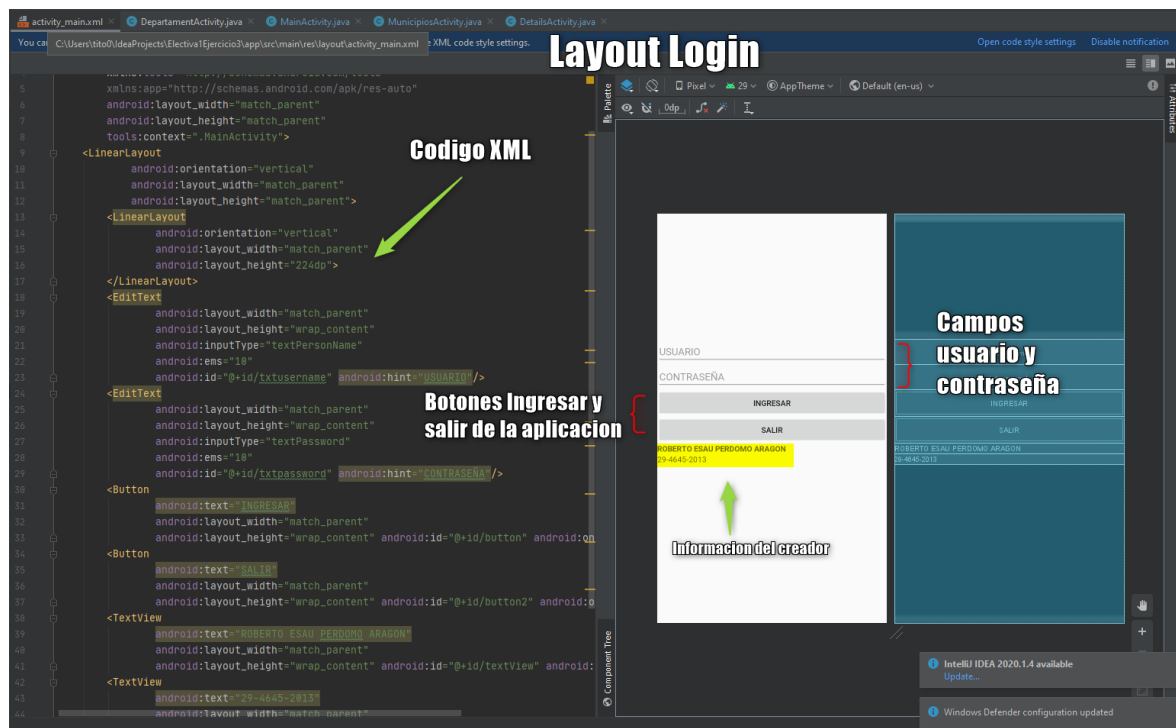
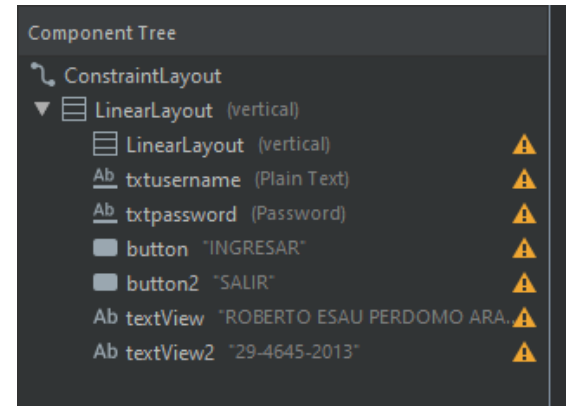
creación de las vistas

Login

Primero utilizamos el layout por default para utilizarlo como Login, para ello utilizo

- 2 TextEditor
 - Obtener Username
 - Obtener Contraseña
- 2 botones
 - Login
 - Salir de la aplicación
- 2 TextView
 - Para mostrar Mi Nombre y contraseña

El árbol se distribuye de la siguiente manera:



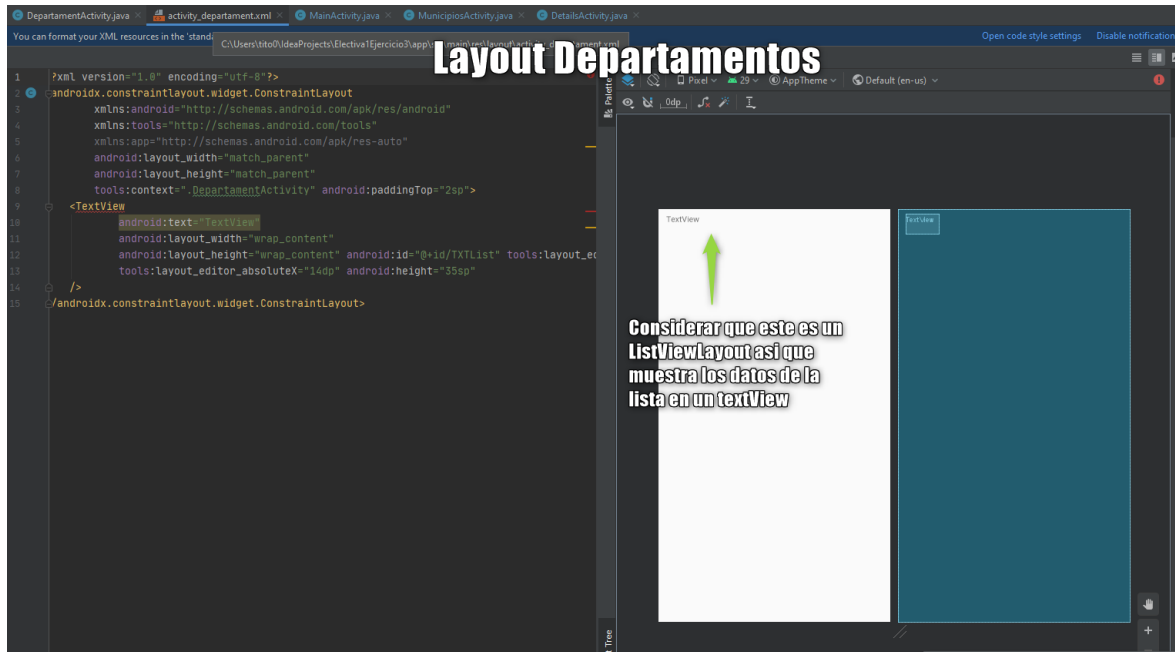
En el caso de los botones utilizan las siguientes Acciones:

INGRESAR: al momento de hacer onclick se activa la acción llamada **Redireccion(View view)**

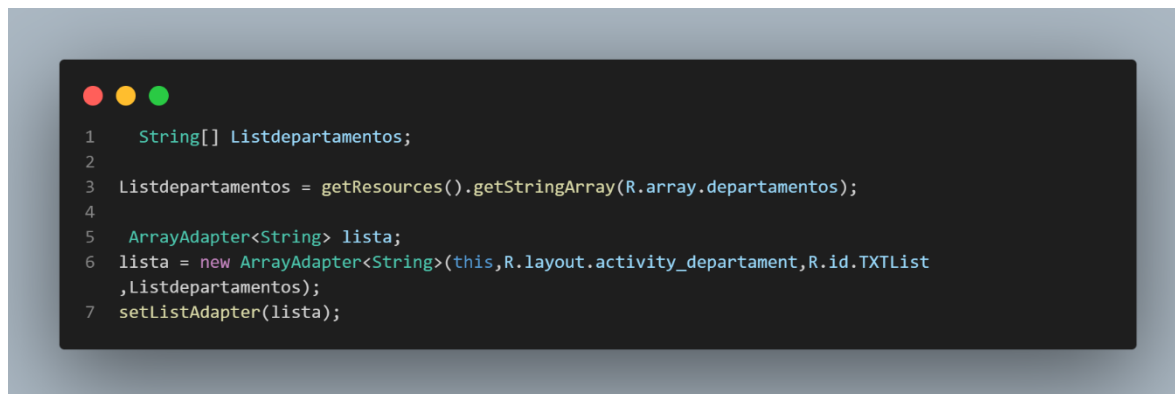
```
1 // Metodo encargado de redireccionar a ActivityDepartamentos
2 public void Redireccion(View view)
3 {
4     String Username =txtUsername.getText().toString();
5     String Password =txtPassword.getText().toString();
6
7     if (ValidatePassword(Username,Password))
8     {
9         Intent intent = new Intent(this, DepartamentActivity.class);
10        startActivity(intent);
11    }
12 }
13
14 //Metodo encargado de validar usuario y contraseña con los datos del XML
15 public boolean ValidatePassword(String Username,String Password)
16 {
17     boolean Result = false;
18
19     if (!IsNullOrBlank(Username) && !IsNullOrBlank>Password))
20     {
21         if (RecurrerData(Username,Password))
22         {
23             Result = true;
24         }
25     }else
26     {
27         Toast.makeText(this, "Usuario o contraseña son incorrectas", Toast.
LENGTH_LONG).show();
28     }
29 }else
30 {
31     Toast.makeText(this, "Usuario o contraseña estan vacios", Toast.LENGTH_LONG).show();
32 }
33 return Result;
34 }
35
36 //Este metodo recorrer los usuarios en el XML
37 public boolean RecurrerData(String Username,String Password) {
38     Boolean Resultado = false;
39     String[] ListaUsuarios,ListaPassword;
40     ListaUsuarios = getResources().getStringArray(R.array.usuarios);
41     ListaPassword = getResources().getStringArray(R.array.claves);
42
43     for(int i = 0; i <= ListaUsuarios.length - 1; i = i + 1)
44     {
45         if (ListaUsuarios[i].toString().equals(Username)){
46             if (ListaPassword[i].toString().equals>Password))
47             {
48                 Resultado= true;
49                 break;
50             }
51         }
52     }
53
54     return Resultado;
55 }
```

Lista departamentos

Este layout hace una extensión de **ListActivity** para lo cual solo vamos a necesitar obtener la data del XML y mostrar la data en un TextView el control hace todo el trabajo para mostrar una lista.



De la siguiente manera obtengo los datos del XML y los seteo en el adaptador para poder ser mostrados en el layout.



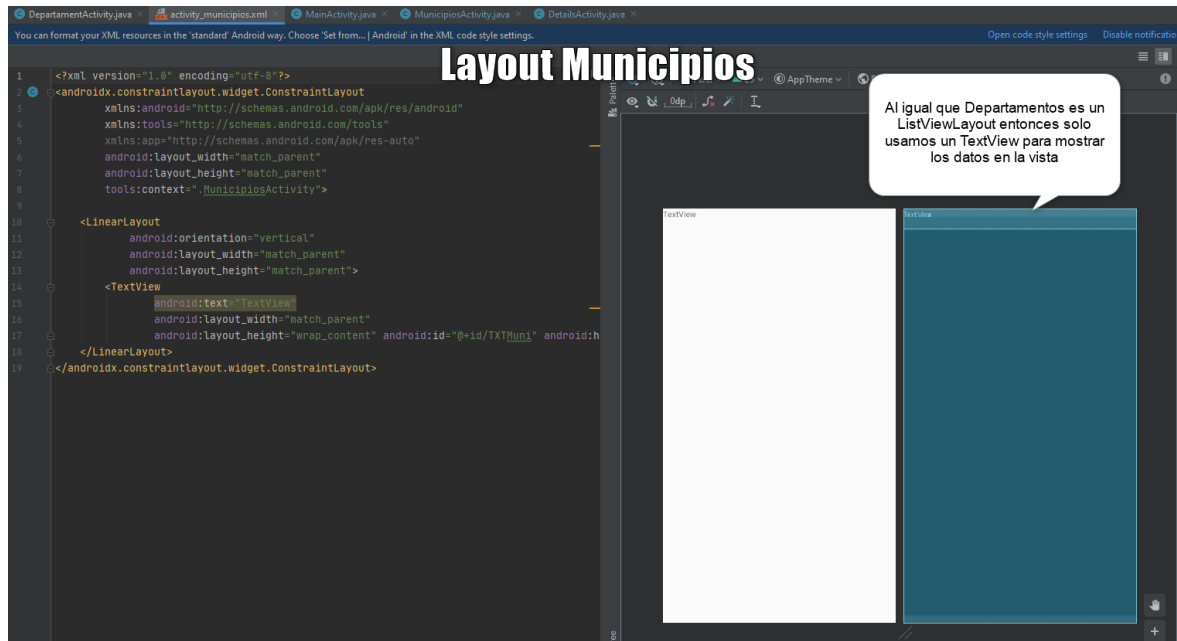
Los métodos más utilizados en departamentos son para obtener capturar el click del usuario en una posición de la lista y hacer la redirección a el layout municipios.

también tenemos uno eliminar acentos de la palabra departamento ya que esa utilizamos en la siguiente vista y así obtener la lista de municipio del XML.

```
1  @Override
2      protected void onItemClick(ListView l, View v, int position, long id) {
3          //super.onItemClick(l, v, position, id);
4
5          String Seleccion =eliminarAcentos(Listdepartamentos[position].toString().toLowerCase().
6          trim().replace(" ", "")) ;
7
8          Toast.makeText(this, Seleccion, Toast.LENGTH_LONG).show();
9
10         if (Seleccion != "")
11         {
12             Intent intent = new Intent(this, MunicipiosActivity.class);
13             intent.putExtra("Departamento", Seleccion);
14             intent.putExtra("IdDepartamento", position);
15             startActivity(intent);
16         }
17
18     public static String eliminarAcentos(String str) {
19
20         final String ORIGINAL = "ÁáÉéÍíÓóÚúÑñÜü";
21         final String REEMPLAZO = "AaEeIiOoUuNnUu";
22
23         if (str == null) {
24             return null;
25         }
26         char[] array = str.toCharArray();
27         for (int indice = 0; indice < array.length; indice++) {
28             int pos = ORIGINAL.indexOf(array[indice]);
29             if (pos > -1) {
30                 array[indice] = REEMPLAZO.charAt(pos);
31             }
32         }
33         return new String(array);
34     }
```

Lista Municipios

Al igual que departamentos estos layout extiende de **ListActivity** así que solamente usamos un **TextView** para mostrar la lista de municipios.



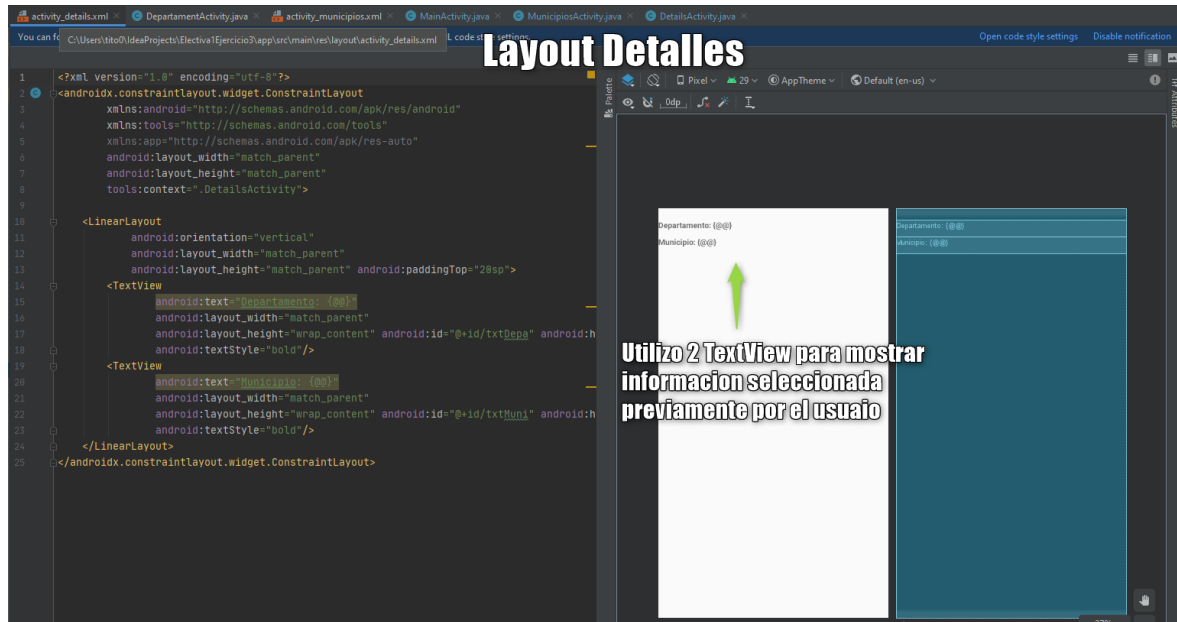
Entre las partes de Código mas importantes encontramos la forma de obtener el id del recurso del XML de forma dinámica.



De igual manera que departamento nosotros mostramos la lista de municipio y enviamos el Id del departamento y el dato del municipio al siguiente layout que es **detalles** para mostrar las selecciones del usuario.

Detalle

En este layout solo utilizamos 2 TextView para mostrar la información seleccionada por el usuario en los layout anteriores

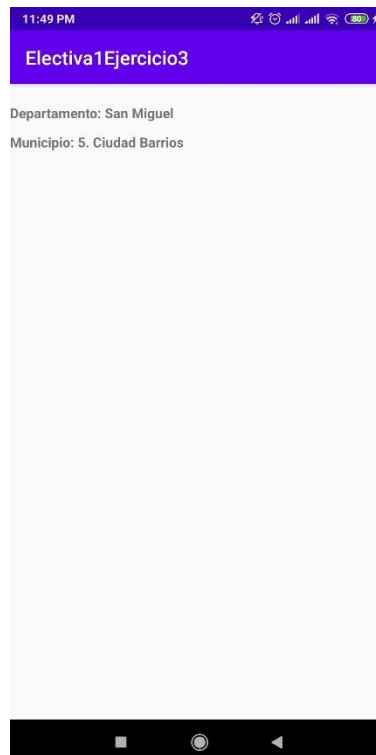
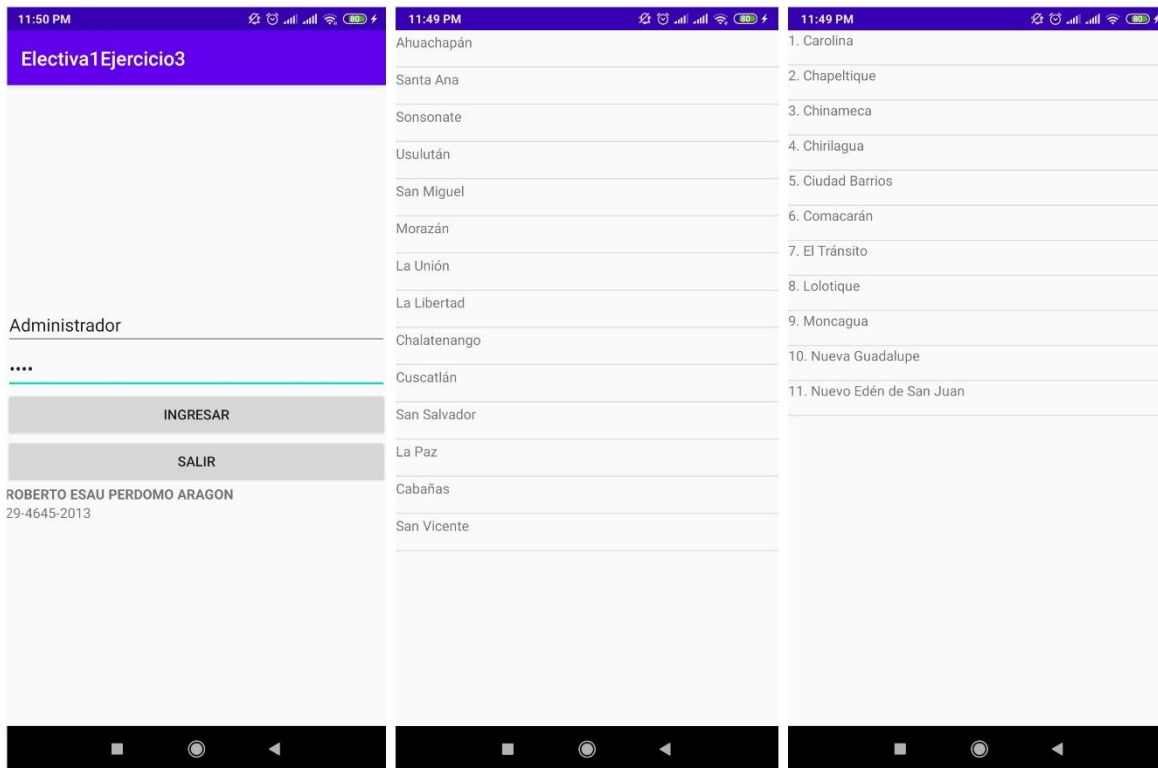


De esta manera capto los parámetros enviados por el layout anterior y los muestro en los TextView



Capturas aplicación en funcionamiento

Capturas en general de todo el flujo.



Bibliografía

Código Fuente creado por Estudiante

<https://github.com/Proyectos-universitarios/Electiva1-Android-Ejercicio-3>