SPINNER

```
String [] opciones = {“sumar”};
ArrayAdapter <String> adapter = new ArrayAdapter<~>(this,
android.R.layout.simple_spinner_item, opciones)
spinner1.setAdapter(adapter);

public void calcular(
string va1_string = e1.gettext.tostring

int va1_int = Integer.parseint(va1_String)

Sting selección = spinner1.getSelectedItem().toString();
if (Selección.equals(“sumar”)){
int suma = va1+va2
string resultado = String.valueof(suma)
}
```

DATE PICKER

```
private TextView mDisplay;
private DatepickerDialog.onDateSetListener mDateSetListener;

mDisplay = (Textview) findViewBydId(R.id.tvDate)
mDisplay.SetOncliklistener(new View.OnClickListener){
public void OnClick(View view){
        Calendar cal= Calendar.getinstance();
        int year = cal.get(Calendar.year);
        int day
        int month

        DatePickerDialog dialog = new DaterPickerDialog(
        MainActivity.this,
        android.R.style.Theme_holo,
        mDateSetListener,
        year,month,day);
        dialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
        dialog.show();

        }
    });

    mDateSetListener = new DatePickerDialog.OnDateSetListener(){
        public void onDateSet(DatePicker datepicker int year, int month, int
day){
        month=month+1;
        Log.d(TAG, “OnDateSet: mm/dd/yyyy:” “+month+”/“ + day + “/“ +
year);
        String date=month + “/“ + day +”/“ + year;
        m.setText(date);
    }
    }
 }
```

CARDVIEW

com.android.support:cardview-v7:25.3.1

en text de los layout escrbires cardview dentro de eso poner un linearlayout
 xmlns:card_view=“auto”

REFERENCIA DE FRAGMENT

```
public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                         Bundle savedInstanceState) {
```

```
    View view =
inflater.inflate(R.layout.fragment_fragment_consultar,
container, false);


return view;
}
```

TAB

```
<TabHost android:id="@android:id/tabhost"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <TabWidget android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@android:id/tabs"
            android:background="#e3f5f4" />

        <FrameLayout android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@android:id/tabcontent" >

            <LinearLayout
                android:id="@+id/tab1"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:orientation="vertical"
                android:visibility="invisible"
                android:weightSum="1”>

         contenido
```

```
     </LinearLayout>

CLASS
TabHost
tabs=(TabHost)view.findViewById(android.R.id.tabhost);
tabs.setup();

TabHost.TabSpec spec=tabs.newTabSpec("mitab1");
spec.setContent(R.id.tab1);
spec.setIndicator("Consulta por Codigo",

res.getDrawable(android.R.drawable.ic_btn_speak_now));
tabs.addTab(spec);

spec=tabs.newTabSpec("mitab2");
spec.setContent(R.id.tab2);
spec.setIndicator("Consulta Todos",
        res.getDrawable(android.R.drawable.ic_dialog_map));
tabs.addTab(spec);

tabs.setCurrentTab(0);
```

Estilos de botones

```
<Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="CONSULTAR"
    android:id="@+id/btnConsultarCT"
    android:layout_marginLeft="120dp"
    android:background="#22bb53"
    android:paddingLeft="15dp"
    android:paddingRight="15dp"
    android:textStyle="bold" />
```

```
@Override
public void onBackPressed() {
    final AlertDialog.Builder builder = new
AlertDialog.Builder(Login.this);
    builder.setMessage("¿Deseas salir de la aplicación?");
    builder.setCancelable(true);
    builder.setNegativeButton("Aceptar", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int
which) {
            finish();

        }
    });
    builder.setPositiveButton("Cancelar", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int
which) {
            dialog.cancel();
        }
    });
    AlertDialog alertDialog = builder.create();
    alertDialog.show();
}


@Override
public void onClick(View v) {
    switch (v.getId()) {

        case R.id.btnIngresar:

            if
(l_txtusuario.getText().toString().trim().equalsIgnoreCase(
"")) {
                l_txtusuario.setError("El campo no puede
estar vacio");
            } else {
                if
(l_txtPassword.getText().toString().trim().equalsIgnoreCase
("")) {
                    l_txtPassword.setError("El campo no
puede estar vacio");
                } else {

                    ConnectivityManager connectivity =
(ConnectivityManager)

getSystemService(Context.CONNECTIVITY_SERVICE);

                    NetworkInfo info_wifi =
connectivity.getNetworkInfo(ConnectivityManager.TYPE_WIFI);

                    NetworkInfo info_datos =
connectivity.getNetworkInfo(ConnectivityManager.TYPE_MOBILE
);


if(String.valueOf(info_wifi.getState()).equals("CONNECTED")
){
                        hiloconexion = new
ObtenerWebServiceLogin();
                        String cadenallamada = GET_SESION +
"?idalumno=" + l_txtusuario.getText().toString() +
"&clave=" + l_txtPassword.getText().toString();
                        hiloconexion.execute(cadenallamada,
"1");   // Parámetros que recibe doInBackground
                    }
                    else{

if(String.valueOf(info_datos.getState()).equals("CONNECTED"
)){
```

```
                            hiloconexion = new
ObtenerWebServiceLogin();
                            String cadenallamada =
GET_SESION + "?idalumno=" +
l_txtusuario.getText().toString() + "&clave=" +
l_txtPassword.getText().toString();

hiloconexion.execute(cadenallamada, "1");   // Parámetros
que recibe doInBackground
                        }
                        else{
                            MostrarToast();
                        }
                    }

                }
            }


            break;
        default:
            break;
    }

}
```