

```

**CONEXION**
String IP = "http://200.170.1.39/WebService";
// Rutas de los Web Services
String GET_SESION = IP + "/login1.php";
ObtenerWebServiceLogin hiloconexion;
public class ObtenerWebServiceLogin extends
AsyncTask<String, Integer, String> {

    @Override
    protected String doInBackground(String... params) {

        String cadena = params[0];
        URL url = null; // Url de donde queremos obtener
información
        String devuelve = "";

        if (params[1] == "1") {

            for(int i=0; i<100; i++){
                publishProgress(i);

                try {
                    Thread.sleep(100);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }

            try {
                url = new URL(cadena);
                HttpURLConnection connection =
(HttpURLConnection) url.openConnection();//Abrir la
conexión
                connection.setRequestProperty("User-Agent",

```

```

"Mozilla/5.0" +
                " (Linux; Android 1.5; es-ES)
Ejemplo HTTP");
        //connection.setHeader("content-type",
"application/json");
        connection.setRequestMethod("GET");
        int respuesta =
connection.getResponseCode();

        if (respuesta == HttpURLConnection.HTTP_OK)
{

                InputStream in = new
BufferedInputStream(connection.getInputStream()); //
preparo la cadena de entrada

                BufferedReader reader = new
BufferedReader(new InputStreamReader(in)); // la
introduzco en un BufferedReader

                // El siguiente proceso lo hago porque
el JSONObject necesita un String y tengo
                // que tranformar el BufferedReader a
String. Esto lo hago a traves de un
                // StringBuilder.
                StringBuilder result = new
StringBuilder();

                String line = "";

                while ((line = reader.readLine()) !=
null) {

result.append(line).toString();          // Paso toda la
entrada al StringBuilder

                }
                //Creamos un objeto JSONObject para

```

poder acceder a los atributos (campos) del objeto.

```
        JSONObject respuestaJSON = new
JSONObject(result.toString());    //Creo un JSONObject a
partir del StringBuilder pasado a cadena
        //Accedemos al vector de resultados

        String resultJSON =
respuestaJSON.getString("estado");    // estado es el nombre
del campo en el JSON

        if (resultJSON == "1") {            // hay un
alumno que mostrar
                // prg.setVisibility(View.VISIBLE);
                Intent i = new
Intent(getApplicationContext(), MainActivity.class);
                startActivity(i);
                Acceso = 1;
                notificacion();

        } else if (resultJSON == "2") {
                devuelve = "Datos incorrectos";
                // MostrarToast();
                Acceso = 0;
        }

    }
} catch (MalformedURLException e) {
    devuelve = "Error URL incorrecta";
    e.printStackTrace();
} catch (IOException e) {
    devuelve = "Error de respuesta del
servidor";

    e.printStackTrace();
} catch (JSONException e) {
    devuelve = "Error desconocido";
    e.printStackTrace();
}
```

```

        }

        return devuelve;

    }

    return null;
}

@Override
protected void onCancelled(String s) {
    super.onCancelled(s);
}

@Override
protected void onPostExecute(String s) {

    l_error.setText(s);
    pb.setVisibility(View.GONE);
    pb.setProgress(0);
    Limpiar();
    //super.onPostExecute(s);

}

@Override
protected void onPreExecute() {

    Toast alertamensaje =
Toast.makeText(getApplicationContext(), "Iniciando Sesion",
Toast.LENGTH_SHORT);
    alertamensaje.show();
    super.onPreExecute();
    pb.setMax(100);
    btnIngresar.setVisibility(View.GONE);
    pb.setVisibility(View.VISIBLE);

```

```

    }

    @Override
    protected void onProgressUpdate(Integer... values) {
        super.onProgressUpdate(values);
        pb.setProgress(values[0]);
    }
}

public void Limpiar(){
    l_txtusuario.setText("");
    l_txtPassword.setText("");
    btnIngresar.setVisibility(View.VISIBLE);
}

public void MostrarToast(){
    LayoutInflater inflater = getLayoutInflater();
    View layout = inflater.inflate(R.layout.toast_layout,
    (ViewGroup) findViewById(R.id.toast_root));

    Toast toast = new Toast(getApplicationContext());
    toast.setGravity(Gravity.CENTER, 0 ,0);
    toast.setDuration	Toast.LENGTH_LONG);
    toast.setView(layout);
    toast.show();
}

**OBTENER CAMPOS ARREGLO**
JSONArray alumnosJSON =
respuestaJSON.getJSONArray("alumnos");    // estado es el
nombre del campo en el JSON

```

```

        devuelve = devuelve + "Codigo: " +
alumnosJSON.getJSONObject(0).getString("idalumno") + "\n" +
"Nombre: " +

alumnosJSON.getJSONObject(0).getString("nombre") + "\n" + "
Carrera: " +

alumnosJSON.getJSONObject(0).getString("carrera") + "\n" +

"-----
-----\n\n";

```

****INGRESAR DATOS CON IMAGEN****

```

JsonObjectRequest jsonObjectRequest;

StringRequest stringRequest;

private void cargarWebService() {

    progreso=new ProgressDialog(getContext());
    progreso.setMessage("Cargando...");
    progreso.show();

    String ip="http://200.170.1.39";

    String url=ip+"/WebService/wsJSONRegistroMovil.php?";

    stringRequest=new StringRequest(Request.Method.POST,
url, new Response.Listener<String>() {

        @Override
        public void onResponse(String response) {
            progreso.hide();

            if
(response.trim().equalsIgnoreCase("registra")){

```

```

        a_txtNombre.setText("");
        a_txtcodigo.setText("");
        a_txtcarrera.setText("");
        Toast.makeText(getApplicationContext(),"Se ha
registrado con exito",Toast.LENGTH_SHORT).show();
    }else{
        Toast.makeText(getApplicationContext(),"No se ha
registrado ",Toast.LENGTH_SHORT).show();
        Log.i("RESPUESTA: ",""+response);
    }

}
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(getApplicationContext(),"No se ha podido
conectar",Toast.LENGTH_SHORT).show();
        progreso.hide();
    }
}){
    @Override
    protected Map<String, String> getParams() throws
AuthFailureError {
        String n=a_txtNombre.getText().toString();
        String cod=a_txtcodigo.getText().toString();
        String ca=a_txtcarrera.getText().toString();

        String imagen=convertirImgString(bitmap);

        Map<String,String> parametros=new HashMap<>();
        parametros.put("nombre",n);
        parametros.put("codigo",cod);
        parametros.put("carrera",ca);
        parametros.put("imagen",imagen);

        return parametros;
    }
};

```

```

        //request.add(stringRequest);
        stringRequest.setRetryPolicy(new
DefaultRetryPolicy(DefaultRetryPolicy.DEFAULT_TIMEOUT_MS *
2, DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
DefaultRetryPolicy.DEFAULT_BACKOFF_MULT));

VolleySingleton.getIntanciaVolley(getContext()).addToReques
tQueue(stringRequest);
}

```

****INGRESAR DATOS SIN IMAGEN****

```

if(params[1]=="1"){    // ingresar

    try {

        HttpURLConnection urlConn;

        DataOutputStream printout;
        DataInputStream input;
        url = new URL(cadena);
        urlConn = (HttpURLConnection)
url.openConnection();
        urlConn.setDoInput(true);
        urlConn.setDoOutput(true);
        urlConn.setUseCaches(false);
        urlConn.setRequestProperty("Content-Type",
"application/json");
        urlConn.setRequestProperty("Accept",
"application/json");
        urlConn.connect();
        //Creo el Objeto JSON
JSONObject jsonParam = new JSONObject();
        jsonParam.put("idalumno", params[2]);
        jsonParam.put("nombre", params[3]);
        jsonParam.put("carrera", params[4]);
        // Envio los parámetros post.
        OutputStream os = urlConn.getOutputStream();
    }
}

```



```

        BufferedWriter writer = new BufferedWriter(
            new OutputStreamWriter(os, "UTF-8"));
        writer.write(jsonParam.toString());
        writer.flush();
        writer.close();

        int respuesta = urlConn.getResponseCode();

        StringBuilder result = new StringBuilder();

        if (respuesta == HttpURLConnection.HTTP_OK) {

            String line;
            BufferedReader br=new BufferedReader(new
InputStreamReader(urlConn.getInputStream()));
            while ((line=br.readLine()) != null) {
                result.append(line);
                //response+=line;
            }

            //Creamos un objeto JSONObject para poder
acceder a los atributos (campos) del objeto.
            JSONObject respuestaJSON = new
JSONObject(result.toString()); //Creo un JSONObject a
partir del StringBuilder pasado a cadena
            //Accedemos al vector de resultados

            String resultJSON =
respuestaJSON.getString("estado"); // estado es el nombre
del campo en el JSON

            if (resultJSON == "1") { // hay un
alumno que mostrar
                devuelve = "Alumno guardado
correctamente";
                // notificacion();

```

```

        } else if (resultJSON == "2") {
            devuelve = "El alumno no pudo
guardarse";
        }

```

```

    }

```

```

    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (JSONException e) {
        e.printStackTrace();
    }

```

```

    return devuelve;

```

```

}

```

```

    return null;

```

```

}

```

****PHP OBTENER ALUMNOS****

```

<?php
$entityBody=file_get_contents('php://input');
$dbHost="localhost";
$dbUsername="root";
$dbPassword="";
$dbName="alumn";
$idalumno=array();
$db = new mysqli($dbHost,$dbUsername,$dbPassword,$dbName);

```

```

if ($db->connect_error){
    die("Conexión fallida: ". $db->connect_error);
}

if(isset($_GET['idalumno']))
{

    $idalumno=$_GET['idalumno'];
    $consulta=("SELECT * FROM alumnos where idalumno =
'$idalumno'");
    $query=$db->query($consulta);
    $result=$query->num_rows;

    if($result>0)
    {
        $idalumnoo["estado"]=1;          // cambio "1" a 1
        porque no coge bien la cadena.
        // Enviar objeto json del alumno

        $idalumnoo["alumnos"]=array();

        while($filas=$query->fetch_object())
        {
            $resultados=array();
            $resultados["idalumno"]=$filas->idalumno;
            $resultados["nombre"]=$filas->nombre;
            $resultados["carrera"]=$filas->carrera;
            $resultados["imagen"]=base64_encode($filas-
>imagen);
            array_push($idalumnoo["alumnos"],
$resultados);

        }

        print json_encode($idalumnoo);
    }
}

```

```

    }
    else{

        $idalumnoo["estado"]=2;          // cambio "1" a 1
        porque no coge bien la cadena.
        // Enviar objeto json del alumno
        print json_encode($idalumnoo);
    }

}

```

?>

```

**DECLARACION PARA BITMAP IMAGEN**
File fileImagen;
Bitmap bitmap;
private static final String CARPETA_PRINCIPAL =
"WebService/";//directorio principal
private static final String CARPETA_IMAGEN = "imagenes";//
carpeta donde se guardan las fotos
private static final String DIRECTORIO_IMAGEN =
CARPETA_PRINCIPAL + CARPETA_IMAGEN;//ruta carpeta de
directorios
private String path;//almacena la ruta de la imagen

```