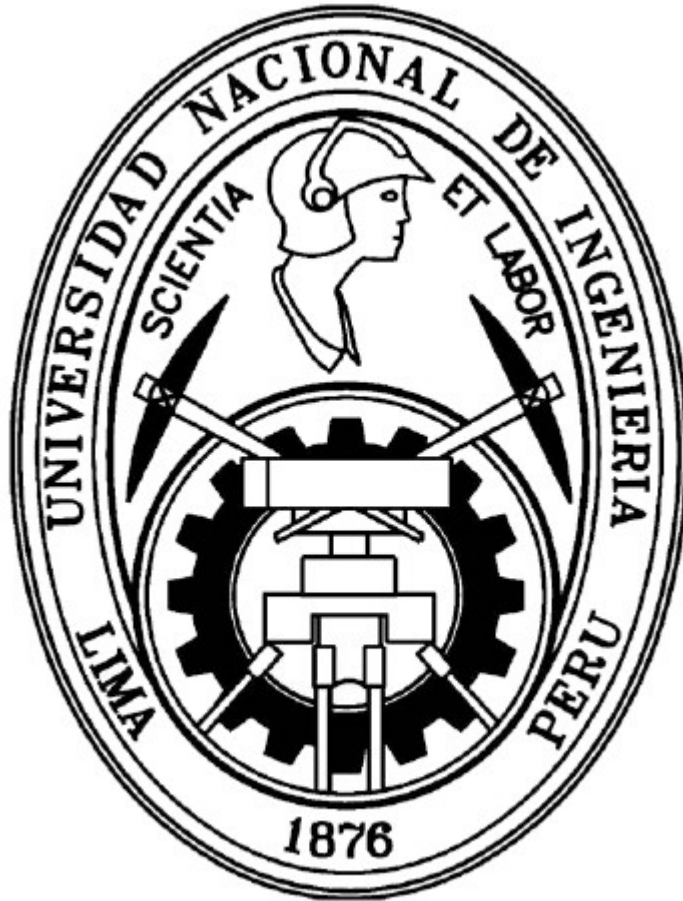


Laboratorio 8.1



Apellidos: Moreno Vera

Nombres: Felipe Adrian

Código: 20120354I

**Asignatura: Programación en Dispositivos Móviles
(CC481)**

2016 - I

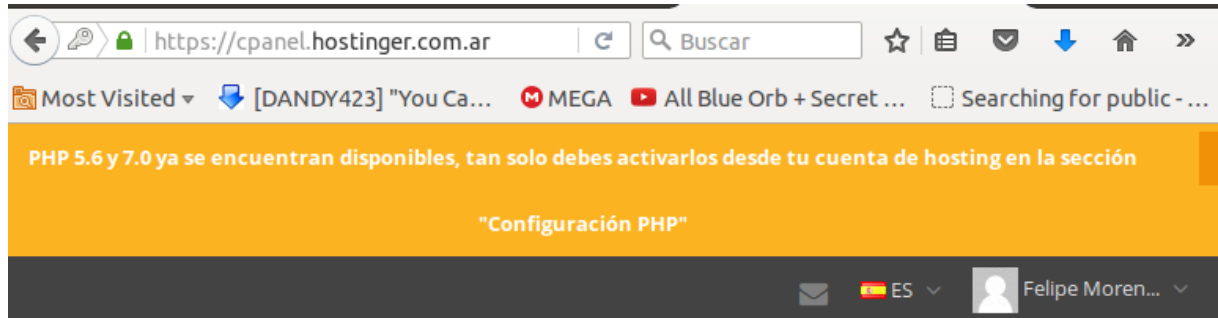
Indice

| | |
|--------------------------|------------|
| Actividad 1 | (3) |
| Actividad 2 | (7) |
| Actividad 3 | (7) |
| Actividad 4 | (7) |
| Actividad 5 | (8) |

Actividad 1

1. Entramos en www.hostinger.com.ar y creamos nuestra cuenta.

Cuenta creada:



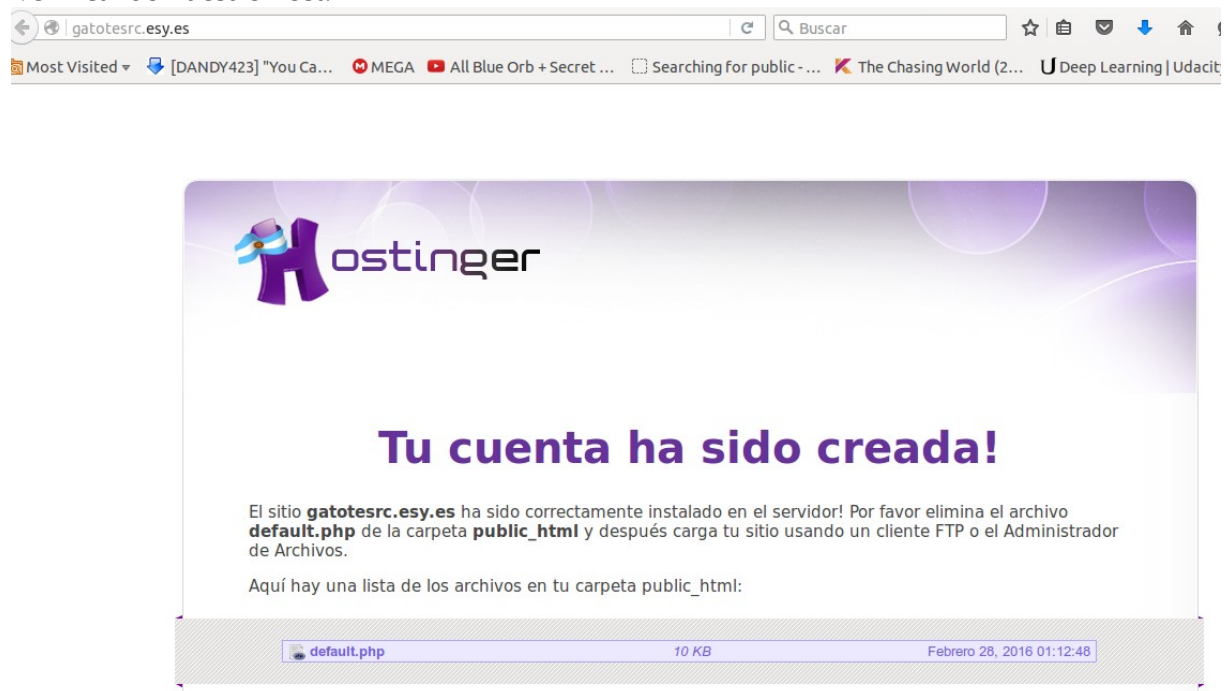
Una vez activada la cuenta, creamos un host:

| | | |
|---------------------------|--|----------------------------------|
| Subdominio | gaTotesrc | .esy.es ▼ |
| Selección de Servidor * | <input checked="" type="radio"/> Europa (UK) <input type="radio"/> Norteamérica (EEUU) <input type="radio"/> Asia (Singapur) <input type="radio"/> Sudamérica (Brasil) <input type="radio"/> Federación Rusa (RUS) | |
| Contraseña * | | Generar |
| Repetir Contraseña * | | Ingresa tu contraseña nuevamente |
| Continuar | | |

Luego vamos a nuestra lista de hosting y vemos que aparece el nuevo.

| Dominio | Plan | Expira en | Estado | Acción |
|--|--------|-----------|--------|--------------------------------|
| gatotesrc.esy.es | Gratis | - | Activo | Mejorar Cuenta |
| <div><div>Administrar</div><div>Creador de Sitios</div><div>Instalador Automático</div><div>Casillas de Correo</div></div> | | | | |

Verificando nuestro host.



2. Una vez creada vamos a “administrar” y luego en el apartado “Base de datos” pulsamos “Base de datos MySQL”. Le ponemos un nombre de BBDD, usuario y contraseña. Verifique estos apartados con el script “crearBD.php” (es un script referencial).

| | |
|-------------------------------|---|
| Nombre de base de datos MySQL | <input type="text" value="u555134733_ BBDD"/> |
| Usuario MySQL | <input type="text" value="u555134733_ root"/> |
| Contraseña | <input type="password" value="....."/> <button>⚙ Generar</button> |
| Contraseña de nuevo | <input type="password" value="....."/> |
| <button>✓ Crear</button> | |

En el fichero, el nombre es pruebaAndroid, pero en el pdf nos dice que tenga de nombre BBDD.

Ya esta creada, verificamos en la lista. (Nota: solo nos permite crear una DB a la vez)

| | Base de Datos MySQL ▲ | Usuario MySQL ▾ | Host MySQL ▾ | Uso de Disco, MB |
|---|-----------------------|-----------------|------------------------|------------------|
| ⊕ | u555134733_bbdd | u555134733_root | mysql.hostinger.com.ar | 0.02 |

3. Verificamos que está creada la BBDD.

4. Abrimos el PHPMyAdmin del lado izquierdo.

1. Creamos la tabla para nuestra práctica. Guiase del script “createTable.php”.
2. Creamos las columnas que pone el fichero siendo:

Table name:

| Name | Type ? | Length/Values ? |
|--------------------------------------|--------------------------------------|---------------------------------|
| <input type="text" value="usuario"/> | <input type="text" value="VARCHAR"/> | <input type="text" value="30"/> |
| <input type="text" value="clave"/> | <input type="text" value="VARCHAR"/> | <input type="text" value="30"/> |
| <input type="text" value="rol"/> | <input type="text" value="VARCHAR"/> | <input type="text" value="30"/> |

5. Una vez creada añadimos 1 registro de prueba que puede tomar como ejemplo el script “insertValueTable.php”.

+ Options

| | usuario | clave | rol |
|---|---------|-------|---------------|
| <input type="checkbox"/> Edit Copy Delete | admin | admin | administrador |





6. Ahora vamos a “Administracion de archivos 2” de nuestro panel y lo instalamos. (se encuentra en el panel, la sección de archivos)

No se requiere instalar nada, nos envía directo.

7. En “Public_html” creamos un directorio que se llame “laboratorio8”.

New dirNew fileUploadJava UploadInstallAdvanced

Transform selected entries: CopyMoveDeleteRenameChmodDownloadZipUnzipSizeSearch

| All | Name | Type | Size | Owner | Group | Perms | Mod Time | Actions |
|--------------------------|--|-------------------------|------|-----------|------------|-----------|--------------|--|
| |  Up.. | | | | | | | |
| <input type="checkbox"/> |  logs | Directory | 4096 | 555134733 | u555134733 | rw-r-xr-x | Feb 28 04:12 | |
| <input type="checkbox"/> |  public_html | Directory | 4096 | 555134733 | u555134733 | rw-r-xr-x | Feb 28 04:12 | |
| <input type="checkbox"/> |  DO NOT UPLOAD HERE | DO_NOT_UPLOAD_HERE File | 0 | 555134733 | u555134733 | rw-r--r-- | May 29 2014 | View Edit Open |

1. En dicha carpeta introducimos los dos archivos. Tenga en cuenta que puede que se hayan modificado según los datos de acceso y tabla que hayas creado.

New dir

New file

Upload

Java Upload

Install

Advanced

Transform selected entries:

Copy

Move

Delete

Rename

Chmod





Download

Zip

Unzip

Size

Search

| All | Name | Type | Size | Owner | Group | Perms | Mod Time | Actions |
|--------------------------|--|---------------|-------|-----------|------------|-----------|--------------|--|
| |  Up.. | | | | | | | |
| <input type="checkbox"/> |  laboratorio8 | Directory | 4096 | 555134733 | u555134733 | rw-r--r-x | Feb 28 05:11 | |
| <input type="checkbox"/> |  htaccess | HTACCESS File | 111 | 555134733 | u555134733 | rw-r--r-- | May 29 2014 | View Edit Open |
| <input type="checkbox"/> |  default.php | PHP script | 10343 | 555134733 | u555134733 | rw-r--r-- | Feb 28 04:12 | View Edit Open |

2. Explique los ficheros PHP, uno es para una conexión con método GET (methodGET.php) y el otro es para una conexión con el método POST (methodPOST.php).

methodGET.php, extrae la info de usuario y clave según el rol que desempeña, y este método si se muestra, y extrae del link, MethodPOST.php extrae la info de usuario y clave según el rol que desempeña, en este método no se muestra los id.

8. Para verificar nuestro servidor escribimos la url debiendo salir todos los registros insertados previamente.

1. www.dominio.sub/laboratorio8/get_all_empresas.php

Los creamos dentro de Labo 8

| | | | | | | | | | | | | | |
|--------------------------|----------------------|------------|------|-----------|------------|-----------|--------------|----------------------|----------------------|----------------------|--|--|--|
| All | Name | Type | Size | Owner | Group | Perms | Mod Time | Actions | | | | | |
| | Up.. | | | | | | | | | | | | |
| <input type="checkbox"/> | createTables.php | PHP script | 218 | 588311430 | u588311430 | rw-r--r-- | Feb 28 07:24 | View | Edit | Open | | | |
| <input type="checkbox"/> | insertValueTable.php | PHP script | 239 | 588311430 | u588311430 | rw-r--r-- | Feb 28 07:25 | View | Edit | Open | | | |

y verificamos si existen:



Index of /laborato

- [Parent Directory](#)
- [createTables.php](#)
- [insertValueTable.php](#)

Actividad 2

1. Ingresar los permisos de conexión

```
<uses-permission android:name="android.permission.INTERNET"/>
```

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

Actividad 3

1. La aplicación realizará una comunicación por POST y GET para conocer ambos apartados es por ello que se deja el diseño antes de la comunicación y después.



Actividad 4

1. Añadimos nuestros atributos.

```
private EditText usernameField,passwordField;  
private TextView status,role,method;
```

2. En nuestro método onCreate capturamos los atributos.

```
usernameField = (EditText)findViewById(R.id.editText1);
```

```
passwordField = (EditText)findViewById(R.id.editText2);
status = (TextView)findViewById(R.id.textView6);
role = (TextView)findViewById(R.id.textView7);
method = (TextView)findViewById(R.id.textView9);
```

3. Creamos nuestro método de login por GET.

```
public void login(View view){
String username = usernameField.getText().toString();
String password = passwordField.getText().toString();
method.setText("Get Metodo");
new SigninActivity(this,status,role,0).execute(username,password); // Clase Java a crear
}
```

4. Creamos nuestro método de login por POST.

```
public void loginPost(View view){
String username = usernameField.getText().toString();
String password = passwordField.getText().toString();
method.setText("Post Metodo");
new JSON(this,status,role,1).execute(username,password); // Clase Java a crear
}
```

Actividad 5

1. Creamos una nueva clase JAVA y le llamamos “SigninActivity” que extenderá de AsyncTask (en caso de que no se comente el código de algún método es por que este se quedará vacío) con los siguientes parámetros <String,Void,String>.

2. Aunque seguramente se importe automáticamente verificamos que se importarán las siguientes librerías.

```
import android.content.Context;
import android.os.AsyncTask;
import android.util.Log;
import android.widget.TextView;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.BufferedReader;
```



```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URI;
import java.net.URL;
import java.net.URLEncoder;
```

Bueno las clases aún no se han importado, se importan luego de escribir código usando dichas clases. Que es en los siguientes ejercicios.

3. Exponemos los siguientes atributos.

```
private TextView statusField,roleField;
private Context context;
private int byGetOrPost = 0;
```

4. El constructor.

```
public SigninActivity (Context context, TextView statusField, TextView roleField,
int flag) {
this.context = context;
this.statusField = statusField;
this.roleField = roleField;
byGetOrPost = flag;
}
```

5. Método doInBackground. De este método se pide explicar su código de manera profunda.

```
protected String doInBackground(String... arg0) {
String stream = null;
JSONObject json = new JSONObject(); //define el objeto JSON
if(byGetOrPost == 0){ //means by Get Method // es el flag sobre GET=0 y POST=1
try{
String username = (String)arg0[0]; // primer argumento usuario
String password = (String)arg0[1]; // primer argumento password
String link = "http://gatotesrcc.esy.es/laboratorio8/methodGET.php?
usuario="+username+"&clave="+password;
String link = "http://dom/subdom/baseDatos/metodoGet.php?
usuario="+username+"&clave="+password; // link con parametros usuario y clave
URL url = new URL(link); // objeto link
HttpClient client = new DefaultHttpClient(); // objeto HttpClient
HttpGet request = new HttpGet(); // objeto HttpGet
request.setURI(new URI(link)); // envía petición al url
}
```

```

HttpResponse response = client.execute(request); // objeto respuesta del cliente
if(response.getStatusLine().getStatusCode() == 200){ // Tiempo de respuesta
    HttpEntity entity = response.getEntity(); // objeto Entity que obtiene entity de la resp
    if (entity != null) {
        InputStream instream = entity.getContent(); // obtiene el contenido del entity
        String result= convertStreamToString(instream); // extrae la data del json
        Log.d("result *****", String.valueOf((result))); // manda un log con el resultado
        json.put("response_", new JSONObject(result)); // envia el response
        instream.close();
    }
    } else {
        Log.d("result ***** error", String.valueOf((0)));
    }
    return String.valueOf(json);
} catch(Exception e){
    return new String("Exception: " + e.getMessage());
}
} else {
    try{
        String username = (String)arg0[0];
        String password = (String)arg0[1];
        String link = "http://gatotesrcc.esy.es/laboratorio8/methodPOST.php";
        String link="http://dominio/subdom/baseDatos/metodoPOST.php"; // link metodo post
        String data = URLEncoder.encode("usuario", "UTF-8") + "=" +
        URLEncoder.encode(username, "UTF-8");
        data += "&" + URLEncoder.encode("clave", "UTF-8") + "=" +new // construye la data
        OutputStreamWriter(conn.getOutputStream()); //
        wr.write( data ); // escribe la data
        wr.flush(); // abre la gestion de stream buffer
        if(conn.getResponseCode() == 200){
            InputStream in = new BufferedInputStream(conn.getInputStream()); // crea el buffer con
            datos de conexion
            BufferedReader r = new BufferedReader(new InputStreamReader(in));
            StringBuilder sb = new StringBuilder(); // crea un string
            String line;
            while ((line = r.readLine()) != null) { // obtiene los datos del string
                sb.append(line);
            }
            stream = sb.toString(); // transforma la data a string
            Log.d("result *****", String.valueOf((stream))); // manda un log con la info
            json.put("response_", new JSONObject(stream)); // descompone el json, los datos
            conn.disconnect(); // desconecta
        }else {
            // something

```

```

}
return String.valueOf(json); // retorna el valor del json ( valor de data )
} catch (Exception e){
return new String("Exception: " + e.getMessage());
}
}
}
}

```

6. Método onPostExecute. Igualmente comente dicho código.

```

protected void onPostExecute(String result){
this.statusField.setText("Correct!"); // señala que hubo ingreso correcto TextView
String json_str = String.valueOf(result); // Obtiene el valor del json
try {
JSONObject my_obj = new JSONObject(json_str); // objeto json
String response = my_obj.getString("response_"); // le añade response_
JSONObject response_ = new JSONObject(response); // objeto json
String data = response_.getString("data"); // obtiene string de respuestas_
Log.d("result", String.valueOf(data)); // manda un Log con el valor de data
this.roleField.setText(data); // cambia el valor del rol textView
} catch (JSONException e) {
e.printStackTrace();
}
}
}

```

7. Finalmente el método convertStreamToString. ¿Que realiza dicho método?

```

private static String convertStreamToString(InputStream is) {
BufferedReader reader = new BufferedReader(new InputStreamReader(is));
StringBuilder sb = new StringBuilder();
String line = null;
try {
while ((line = reader.readLine()) != null) {
sb.append(line + "\n");
}
} catch (IOException e) {
e.printStackTrace();
} finally {
try {
is.close();
} catch (IOException e) {
e.printStackTrace();
} } return sb.toString(); }

```

Lee lo que hay en un buffer y lo convierte a String.

Solución:

Como yo estoy trabajando en compileSDKVersion 23, apache no me da soporte para las librerías HTTPClient. Por lo que tuve que usar.

```
useLibrary 'org.apache.http.legacy'
packagingOptions {
    exclude 'META-INF/DEPENDENCIES'
    exclude 'META-INF/NOTICE'
    exclude 'META-INF/LICENSE'
    exclude 'META-INF/LICENSE.txt'
    exclude 'META-INF/NOTICE.txt'
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.1.1'
    compile 'com.android.support:design:23.1.1'
    compile group: 'cz.msebera.android', name: 'httpclient', version: '4.4.1.1'
}
```

cz.msebera.android, los cuales si tienen las librerías.

Respondiendo a la pregunta 2, sobre las librerías:

```
import android.content.Context;
import android.os.AsyncTask;

import android.util.Log;
import android.widget.TextView;

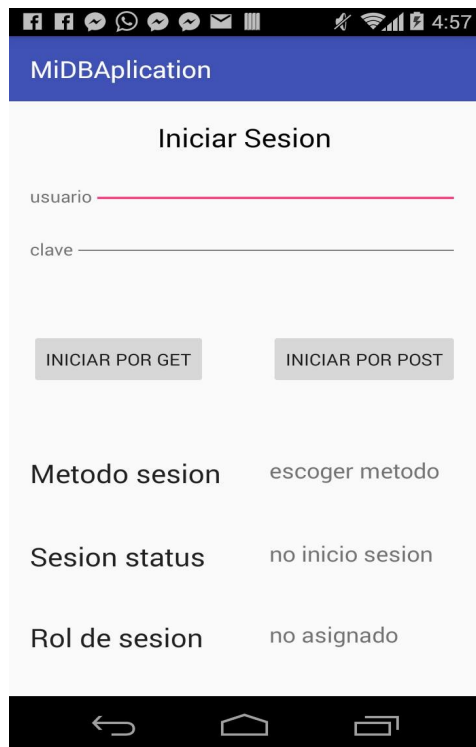
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URI;
import java.net.URL;
import java.net.URLEncoder;

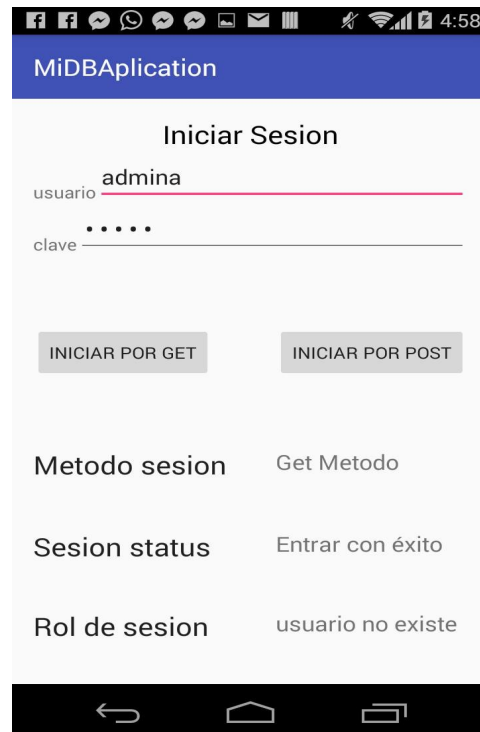
import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.client.HttpClient;
import cz.msebera.android.httpclient.client.methods.HttpGet;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
```

Mi host en hostinger.com.ar es : gaTotesrcc.esy.es :D accede a /laboratorio8 y verás el index.php

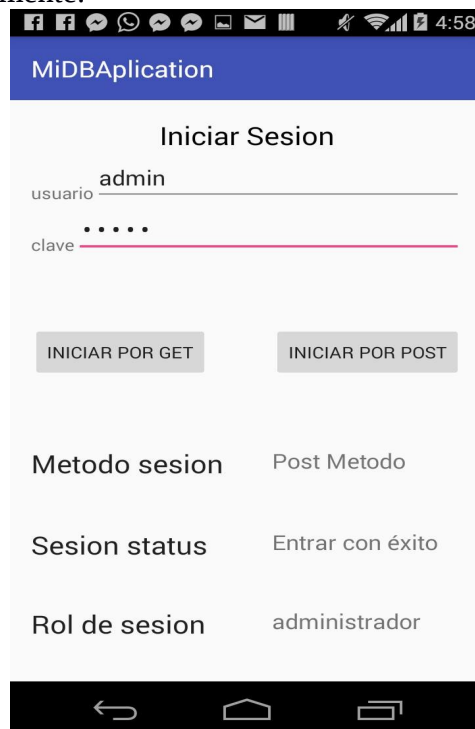
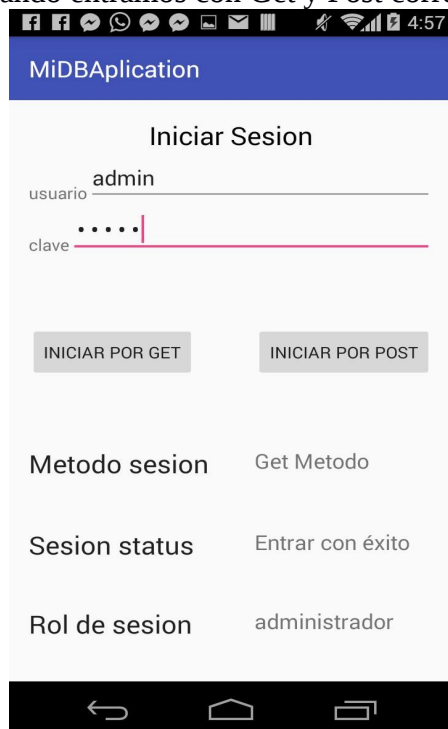
Vista del proyecto ya compilado y ejecutado en mi android:



Cuando entramos con un usuario incorrecto:



Cuando entramos con Get y Post correctamente:



Link del github con los códigos del laboratorio:

https://github.com/Jenazad/PDM/tree/master/Laboratorio_8

Referencias

<http://php.net/manual/es/reserved.variables.get.php>
<http://php.net/manual/es/reserved.variables.post.php>
<https://hc.apache.org/httpcomponents-client-4.5.x/android-port.html>
<http://stackoverflow.com/questions/15775747/how-do-i-properly-import-httpclient-from-org-apache-on-android-using-gradle-build>
<https://github.com/smarek/httpclient-android/wiki/Project-Introduction>
<http://www.hablemosdeandroid.com/2015/12/como-conectar-mysql-con-android.html>
<http://cursoandroidstudio.blogspot.pe/2015/01/base-de-datos-remotas-login.html>