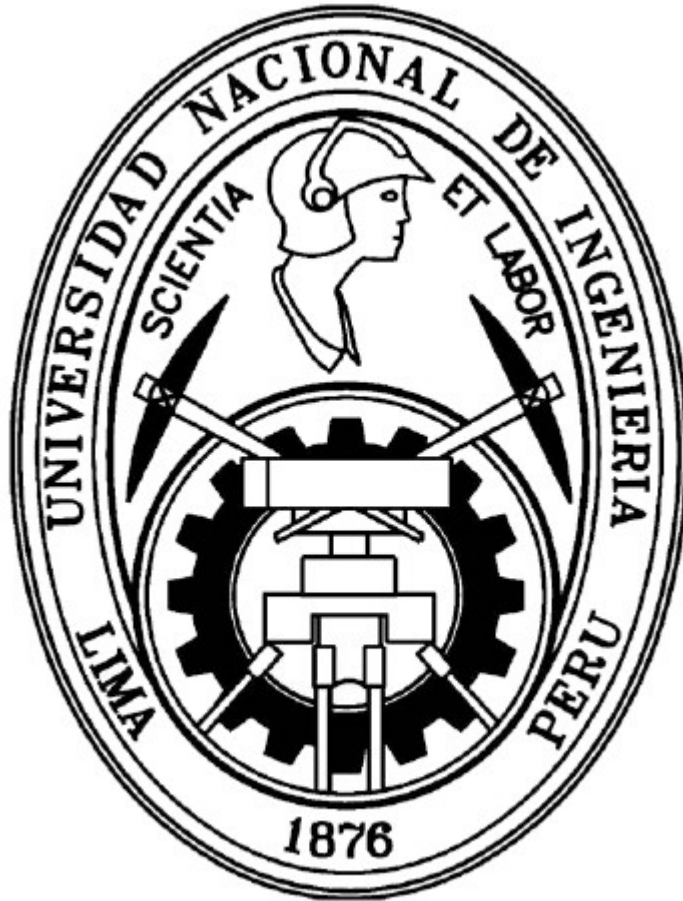


Laboratorio 8.2



Apellidos: Moreno Vera

Nombres: Felipe Adrian

Código: 20120354I

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

2016 - I










Indice

Actividad 1	(3)
Actividad 2	(4)
Actividad 3	(4)
Actividad 4	(6)
Actividad 5	(8)
Actividad 6	(8)

Actividad 1

1. Entramos en www.hostinger.com.ar y creamos nuestra cuenta.
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.
 1. Estos datos de usuario y BBDD se tendrán que poner en el fichero “db_config.php”
3. Verificamos que está creada la BDD.
4. Abrimos el PHPMyAdmin del lado izquierdo.
 1. Creamos la tabla “empresa” que puede verificarlo en el fichero “db_get_allempresas”.
 2. Creamos las columnas que pone el fichero siendo:
 1. id clave primaria y haciendo check en la columna A_1
 2. Cotejamiento de nombre ponemos “utf8_spanish...”y le damos longitud.
5. Una vez creada añadimos 3 registros de prueba.

+ Options

					id	nombre
<input type="checkbox"/>		Edit		Copy		Delete
1						MasTables
<input type="checkbox"/>		Edit		Copy		Delete
2						GeckNerds
<input type="checkbox"/>		Edit		Copy		Delete
3						MeowTron








6. Ahora vamos a “Administracion de archivos 2” de nuestro panel y lo instalamos.
7. En “Public_html” creamos un directorio que se llame “laboratorio8”.
 1. En dicha carpeta introducimos los tres archivos. Tenga en cuenta que puede que se hayan modificado según los datos de acceso y tabla que hayas creado.
 2. Explique los ficheros PHP.

Bueno, solo hay que insertar 2, en el fichero db_config, estan definidos el nombre, root, passwd y usuario de la base de datos.

En el segundo get_all_empresas.php es el programa para retornar según la consulta `SELECT * FROM empresa;`

8. Para verificar nuestro servidor escribimos la url debiendo salir todos los registros insertados previamente. 1. www.dominio.sub/laboratorio8/get_all_empresas.php

Bueno, usaremos el mismo que el del laboratorio 8.1.

All	Name	Type
	Up..	
<input type="checkbox"/>	 createTables.php	PHP script
<input type="checkbox"/>	 db_config.php	PHP script
<input type="checkbox"/>	 get_all_empresas.php	PHP script
<input type="checkbox"/>	 index.php	PHP script
<input type="checkbox"/>	 insertValueTable.php	PHP script
<input type="checkbox"/>	 methodGET.php	PHP script
<input checked="" type="checkbox"/>	 methodPOST.php	PHP script

Actividad 2

1. Ingresar los permisos de conexión

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

Actividad 3

1. Creamos una nueva clase JAVA y le llamamos “JSONParser”.

2. Creamos los atributos y el constructor.

```
static InputStream is = null;
static JSONObject jsonObj = null;
static String json = “”;
public JSONParser() {
}
```

3. Creamos el método obtener un JSON desde la url haciendo mediante el método HTTP POST o GET. No se limite a copiar pegar, comenta el código.

```
public JSONObject makeHttpRequest(String url, String method, List params) {
try {
if(method.equals("POST")){
DefaultHttpClient httpClient = new
HttpPost(httpPost = new HttpPost(url);
httpClient.execute(httpPost);
HttpEntity httpEntity = httpResponse.getEntity();
}
```

```
// Evalua si es POST
// Objeto HttpClient Default
// Objeto HttpPost
// setea Entity
// Objeto HttpResponse
// retorna la consulta
// Objeto HttpEntity
```

```

is = httpEntity.getContent();
}else if(method.equals("POST")){
DefaultHttpClient httpClient =
new DefaultHttpClient(); // Objeto DefaultHttpClient
String paramString =
URLEncodedUtils.format(params,"utf-8");
url += "?" + paramString;
HttpGet httpGet = new HttpGet(url); // Obtiene un url
HttpResponse httpResponse =
httpClient.execute(httpGet); // Ejecuta el HttpClient
HttpEntity httpEntity = httpResponse.getEntity();
is = httpEntity.getContent(); // devuelme el Contenido
}
} catch (UnsupportedEncodingException e) {
e.printStackTrace();
} catch (ClientProtocolException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}
}
try {
BufferedReader reader = new BufferedReader( // es el buffer de entrada
new InputStreamReader(is, "iso-8859-1"), 8);
StringBuilder sb = new StringBuilder(); // String para almacenar
String line = null;
while ((line = reader.readLine()) != null) { // Lee datos hasta EOF
sb.append(line + "\n"); // concatena con \n
}
is.close(); // cierra
json = sb.toString(); // transforma a json format
} catch (Exception e) {
Log.e("Buffer Error", "Error convert result" + e.toString()); // Log si es que hay error
}
try {
jObj = new JSONObject(json);
} catch (JSONException e) {
Log.e("JSON Parser","Error parsing data" + e.toString());
}
return jObj; // retorna objeto json
}

```

Vamos a comenzar a usar la segunda actividad que habiamos creado.

Actividad 4

1. Para todas las actividades no se olvide de comentar el código

2. Agregamos los atributos, tanto el objeto JSON como los que vayamos a necesitar de comunicación con el servidor.

```
private ProgressDialog pDialog; // Objeto ProgressDialog
JSONParser jParser = new JSONParser(); // objeto JSONParser
ArrayList<HashMap<String, String>> empresaList; // ArrayList
private static String url_all_empresas = // url de nuestro host y el script.
"http://midireccion/laboratorio8/get_all_empresas.php";
private static final String TAG_SUCCESS = "success"; //Strings
private static final String TAG_PRODUCTS = "empresas";
private static final String TAG_ID = "id";
private static final String TAG_NOMBRE = "nombre";
JSONArray products = null;
ListView lista;
```

3. El método onCreate tendrá:

```
public void onCreate(Bundle savedInstanceState) { // muestra las deficiones de los id
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_my);
empresaList = new ArrayList<HashMap<String, String>>();
new LoadAllProducts().execute();
lista = (ListView) findViewById(R.id.listAllProducts); // muestra la ListView
ActionBar actionBar = getSupportActionBar();
actionBar.setDisplayHomeAsUpEnabled(true);
}
```

4. Por último vamos a realizar un AsyncTask para la conexión con nuestro servidor.

1. La cabecera:

```
class LoadAllProducts extends AsyncTask<String, String, String> {
```

2. El método onPreExecute

@Override

```
protected void onPreExecute() { // crea un progressDialog
super.onPreExecute();
pDialog = new ProgressDialog(MyActivity.this); // se ejecuta en esa activity
pDialog.setMessage("Cargando comercios. Por favor espere..."); // mensaje mostrado
pDialog.setIndeterminate(false); // no termina
pDialog.setCancelable(false); // no se cancela
pDialog.show(); // mostrar
}
```

3. El método doInBackground

```
protected String doInBackground(String... args) {
    List params = new ArrayList(); // inicia la listView
    JSONObject json = jParser.makeHttpRequest( // Objeto Json que hace petición
        url_all_empresas, "GET", params); // Parámetros
    Log.d("All Products: ", json.toString()); // Lanza un Log con todo
    try {
        int success = json.getInt(TAG_SUCCESS); // intenta conectar.
        if (success == 1) {
            products=json.getJSONArray(TAG_PRODUCTS);
            for (int i = 0; i < products.length(); i++) { // extrae los ids de todos los elementos
                JSONObject c = products.getJSONObject(i);
                String id = c.getString(TAG_ID); // ID
                String name = c.getString(TAG_NOMBRE); // Nombre
                HashMap map = new HashMap(); // Crea un Hash Map apra almacenar
                map.put(TAG_ID, id); // añade TAG_ID con etiqueta id
                map.put(TAG_NOMBRE, name); // añade TAG_NOMBRE con etiqueta nombre
                empresaList.add(map); // Lo añade a listar
            }
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
    return null; }
```

4. El método onPostExecute

```
protected void onPostExecute(String file_url) {
    pDialog.dismiss(); // quita el ProgressDialog del screen
    runOnUiThread(new Runnable() {
        public void run() {
            ListAdapter adapter = new SimpleAdapter( // Creamos el ListAdapter
                MyActivity.this, empresaList, R.layout.single_post,
                new String[] {
                    TAG_ID, // se añade los TAG_NOMBRE y TAG_ID
                    TAG_NOMBRE,
                },
                new int[] {
                    R.id.single_post_tv_id, // añade los TextView que se muestran
                    R.id.single_post_tv_nombre, // de ID y nombre
                },
                null);
            //setListAdapter(adapter);
            lista.setAdapter(adapter); }
    }); }
```

Actividad 5

1. Fichero xml main

```
<LinearLayout // Crea un Linear Layout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical">
<ListView // Crea el ListView
android:id="@+id/listAllProducts"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:dividerHeight="3dp"
android:background="#fff"/>
</LinearLayout>
```

Actividad 6

1. Fichero XML

```
<LinearLayout xmlns:android="http://schemas.android.com/.." // Crea otro LinearLayout
android:layout_width="fill_parent"
android:layout_height="match_parent"
android:background="#f0f0f0"
android:orientation="vertical" >
<TextView // Crea los Text View que seran
// usados para mostrar la data
android:id="@+id/single_post_tv_nombre"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="bottom"
android:paddingBottom="2dip"
android:padding="10dp"
android:textColor="#333"
android:textSize="12dp"
android:textStyle="bold"
android:text="nombre" />
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceSmall"
android:text="id"
android:id="@+id/single_post_tv_id"
android:visibility="gone" />
</LinearLayout>
```


Para cambiar entre la aplicación de labo 8.1 y el labo 8.2, ve al manifest y descomenta el intent-filter del MainActivity y comenta el intent-filter del SedunaActivity para ejecutar el MiDBAplication o MiHostServerAplication (Labo 8.1) (lo renombre XD) y hacer en viceversa para ejecutar MiHostServerJSON (Labo 8.2)

Solución:

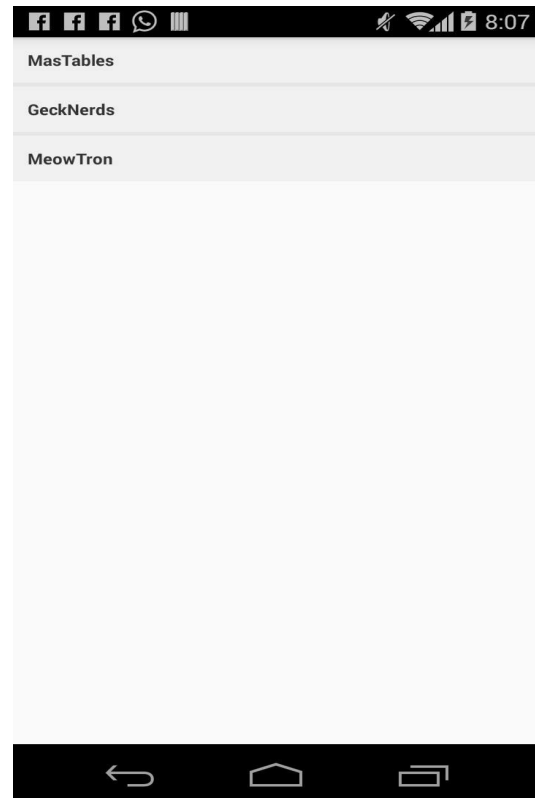
Bueno, primero en la linea:

```
JSONObject json = jParser.makeHttpRequest(url_all_empresas, "POST",  
params);
```

debe ser POST en vez de GET, pues no se esta pasando ningún parámetro en el url.

Mostrando el ProgressDialog . . .

Una vez ingresada la consulta



Link del github con los códigos del laboratorio:

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

Referencias

<http://developer.android.com/intl/es/reference/android/app/ProgressDialog.html>
<http://cursoandroidstudio.blogspot.pe/2014/08/base-de-datos-remotas.html>
<http://stackoverflow.com/questions/17743332/jsonparser-makehttprequest-error-in-json-parsing>
<http://stackoverflow.com/questions/25029861/makehttprequesturl-create-product-post-params-method-displaying-error>