
ACTIVIDAD	:	Laboratorio Nro. 04
TEMA	:	HttpClient - Consultas
DOCENTE	:	Ing. Eric Gustavo Coronel Castillo

INDICE

OBJETIVO	3
DESCRIPCIÓN	3
BASE DE DATOS	3
Diagrama E-R.....	3
Vista para consultar los movimientos.....	3
PROYECTO WEB CON PHP	4
Estructura del proyecto	4
Parámetros de conexión	4
Clase de conexión genérica.....	4
Capa DAO	5
Capa de servicios	6
Consultar los datos de un cliente.....	6
Consultar movimientos de una cuenta.....	7
CREACIÓN DEL PROYECTO.....	7
ACTIVIDAD PARA CLIENTE.....	8
ACTIVIDAD PARA MOVIMIENTOS	8
INTERFAZ DE USUARIO.....	8
activity_main.....	8
activity_cliente	9
activity_movimientos.....	9
layout_item	10
PROGRAMACIÓN	11
Permisos de Internet.....	11
MainActivity	11
Propiedad: onClick	11
Método: onCreate.....	11

Android Avanzado

Método: onClickCliente	12
Método: onClickMovimientos.....	12
Evento: onClickSalir	12
Clase: Util	12
ClienteActivity	13
Propiedad: onClick	13
Método: onClickConsultar	13
Método: onClickRetornar	14
Prueba	14
Programando el layout_item	14
Programando el adaptador	15
MovimientosActivity	16
Código.....	16
Prueba	18

OBJETIVO

En este laboratorio el participante aprende a acceder a bases de datos remotas utilizando HttpClient.

DESCRIPCIÓN

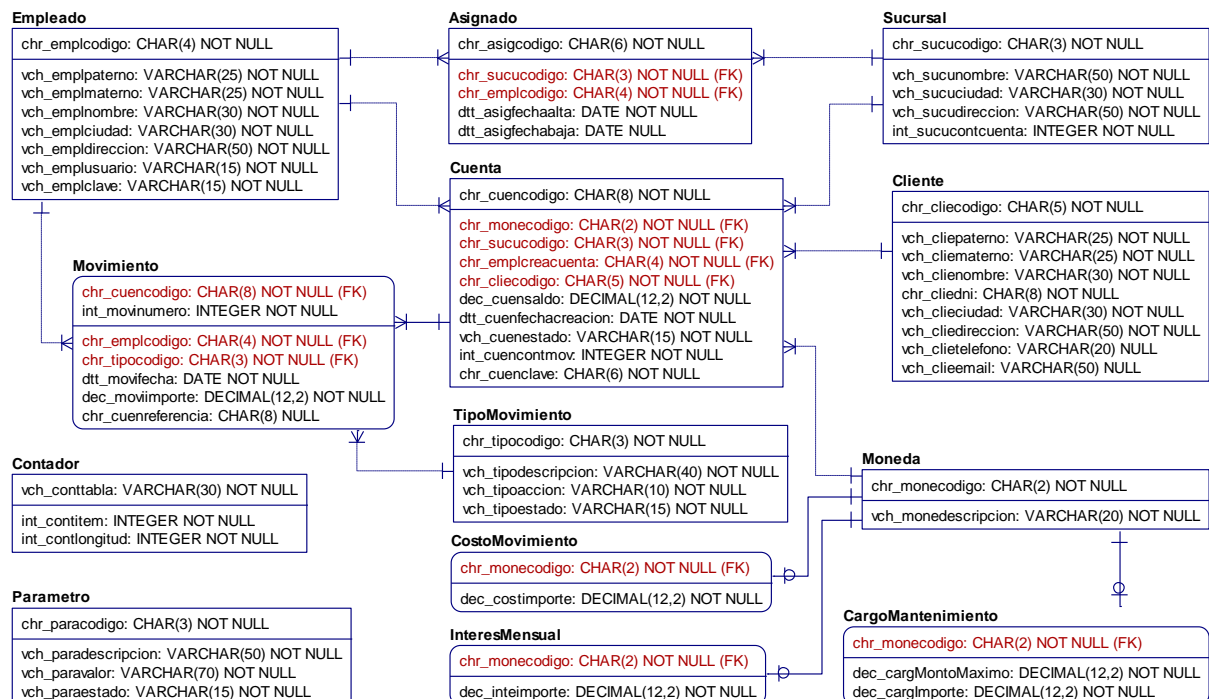
Debe desarrollar una aplicación que permita consultar los datos de un registro y una colección de registros de una base de datos MySQL utilizando HttpClient y JSON.

BASE DE DATOS

Diagrama E-R

Para el presente laboratorio utilizaremos la base de datos EUREKABANK que la puede descargar de <http://gcoronelc.blogspot.com/2011/08/base-de-datos-ejemplo.html>.

El modelo de datos es el siguiente:



Vista para consultar los movimientos

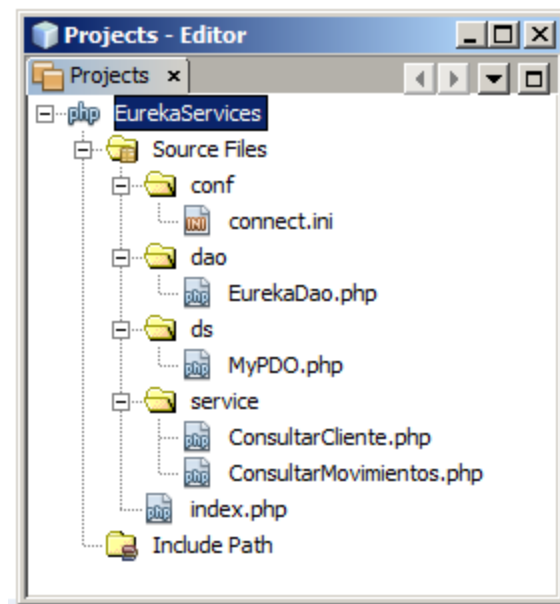
```
CREATE VIEW vMovimientos(  
    cuenta, nromov, fecha, codtipo, tipo, accion, importe, referencia  
)
```

Android Avanzado

```
AS
SELECT
    m.chr_cuencodigo, m.int_movinúmero, m.dtt_movifecha, tm.chr_tipocódigo,
    tm.vch_tipodescripción, tm.vch_tipoacción, m.dec_moviimporte,
m.chr_cuenreferencia
FROM tipomovimiento tm
JOIN movimiento m
ON tm.chr_tipocódigo = m.chr_tipocódigo;
```

PROYECTO WEB CON PHP

Estructura del proyecto



Parámetros de conexión

Los parámetros de conexión se configuran en el archivo **connect.ini**:

```
[database]
01="mysql:host=192.168.1.38;dbname=eurekabank"
02=eureka
03=admin
```

Clase de conexión genérica

En la clase **MyPDO** se configura la conexión con la base de datos de forma genérica:

Android Avanzado

```
<?php
class MyPDO {
    protected $pdo = null;
    function __construct() {
        try {
            // Parámetros
            $param = parse_ini_file("../conf/connect.ini",true);
            $dsn = $param["database"]["01"];
            $username = $param["database"]["02"];
            $passwd = $param["database"]["03"];
            // Conexión
            $this->pdo = new PDO($dsn, $username, $passwd);
            $this->pdo->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
            $this->pdo->setAttribute(PDO::ATTR_CASE,PDO::CASE_LOWER);
            $this->pdo->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE,PDO::FETCH_ASSOC);
        } catch (Exception $e) {
            throw new Exception("Error de acceso a la base de datos");
        }
    }
}
?>
```

Capa DAO

En la clase EurekaDao se programa las dos consultas:

```
<?php

require_once '../ds/MyPDO.php';

class EurekaDao extends MyPDO{

    public function consultarCliene( $codigo ) {
        $rec = null;
        try {
            $query = "select chr_cliecodigo, vch_cliepaterno, vch_cliematerno,
vch_clienombre, chr_cliedni, vch_clieciudad, vch_cliedireccion,
vch_clietelefono, vch_clieemail
from cliente where chr_cliecodigo = ?";
            $stm = $this->pdo->prepare($query);
            $stm->bindParam(1, $codigo );
            $stm->execute();
            $rec = $stm->fetch(PDO::FETCH_ASSOC);
        }
    }
}
```

Android Avanzado

```
        } catch (Exception $e) {
            throw $e;
        }
        return $rec;
    }

    public function consultarMovimientos( $cuenta ) {
        $lista = null;
        try {
            $query = "select cuenta, nromov, fecha, codtipo,
            tipo, accion, importe
            from vMovimientos where cuenta = ?";
            $stm = $this->pdo->prepare($query);
            $stm->bindParam(1, $cuenta );
            $stm->execute();
            $lista = $stm->fetchAll(PDO::FETCH_ASSOC);
        } catch (Exception $e) {
            throw $e;
        }
        return $lista;
    }
}

?>
```

Capa de servicios

Consultar los datos de un cliente

El programa **ConsultarCliente.php** permite obtener los datos de un cliente en formato JSON:

```
<?php
require_once '../dao/EurekaDao.php';

// Parámetros
$codigo = $_REQUEST["codigo"];

// Proceso
$dao = new EurekaDao();
$rec = $dao->consultarCliene($codigo);
if($rec){
    $rec["estado"] = 1; // Existe
} else {
```

Android Avanzado

```
$rec["estado"] = 0; // No existe
}
$data = json_encode($rec);

// Retorno
echo $data;
?>
```

Consultar movimientos de una cuenta

El programa **ConsultarMovimientos.php** permite obtener todos los movimientos de una cuenta:

```
<?php
require_once '../dao/EurekaDao.php';

// Parámetros
$cuenta = $_REQUEST["cuenta"];

// Proceso
$dao = new EurekaDao();
$lista = $dao->consultarMovimientos($cuenta);
$data = "";
if($lista){
    $data = json_encode($lista);
}

// Retorno
echo $data;
?>
```

CREACIÓN DEL PROYECTO

- Crear un nuevo proyecto con las siguientes especificaciones iniciales:

Application Name	HttpClient
Project Name	Lab04_HttpClient
Package Name	pe.egcc.httpclient

- El icono según su criterio.
- Una actividad en blanco con las siguientes especificaciones:

Android Avanzado

Activity Name	MainActivity
Layout Name	activity_main
Navigation Type	None

ACTIVIDAD PARA CLIENTE

Proceda a crear una actividad para consultar los datos de un cliente, los datos son los siguientes:

Activity Name	ClienteActivity
Layout Name	activity_cliente
Navigation Type	None

ACTIVIDAD PARA MOVIMIENTOS

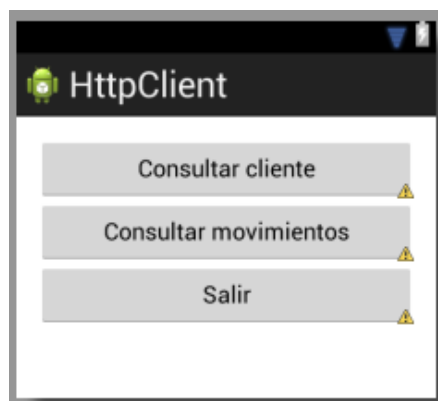
Proceda a crear una actividad para consultar los movimientos de una cuenta, los datos son los siguientes:

Activity Name	MovimientosActivity
Layout Name	activity_movimientos
Navigation Type	None

INTERFAZ DE USUARIO

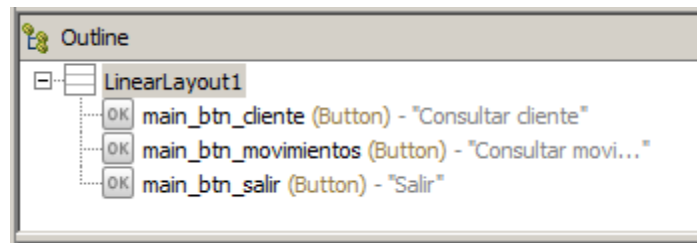
activity_main

En el layout de la actividad principal (activity_main) agregue los widgets para construir la siguiente interfaz:



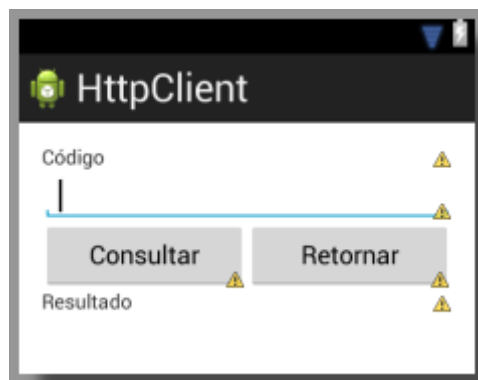
Android Avanzado

A continuación tenemos la estructura de la interfaz:

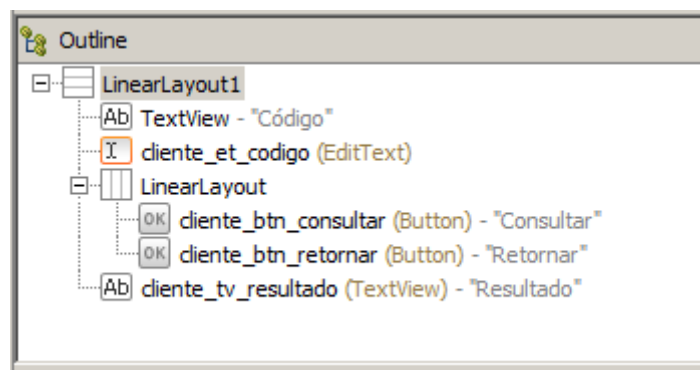


activity_cliente

En el layout de la actividad de cliente (activity_cliente) agregue los widgets para construir la siguiente interfaz:



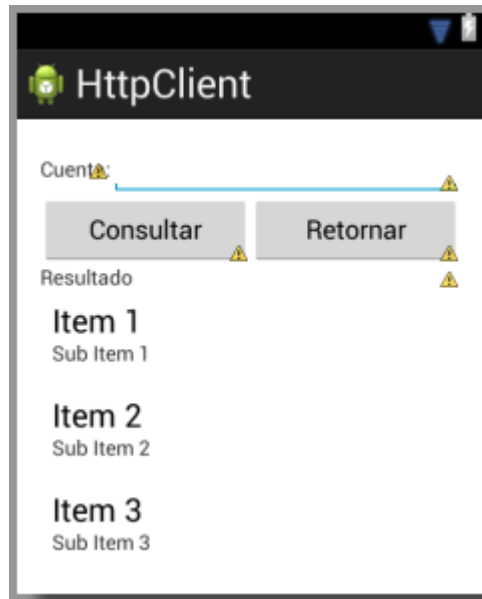
A continuación tenemos la estructura de la interfaz:



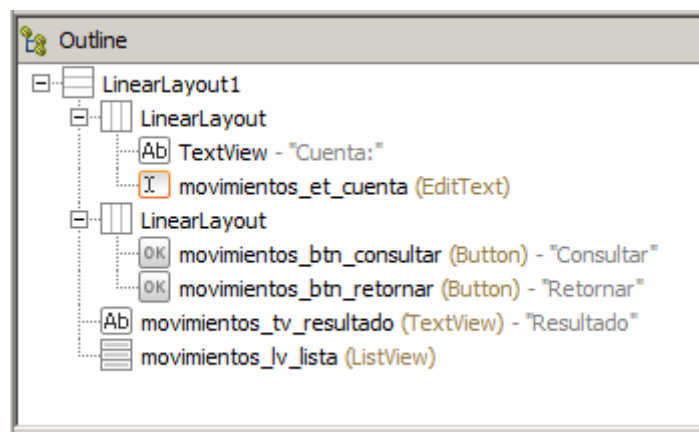
activity_movimientos

En el layout de la actividad de movimientos (activity_movimientos) agregue los widgets para construir la siguiente interfaz:

Android Avanzado



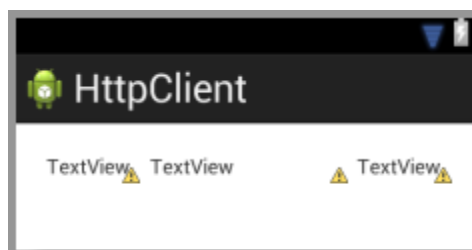
A continuación tenemos la estructura de la interfaz:



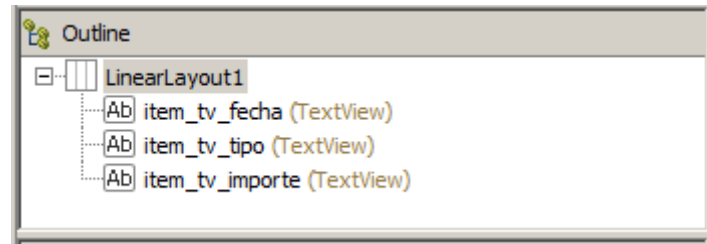
layout_item

Se trata de un layout personalizado que representa una fila del ListView donde mostraremos los movimientos de una cuenta.

Agregue los widgets para construir la siguiente interfaz:



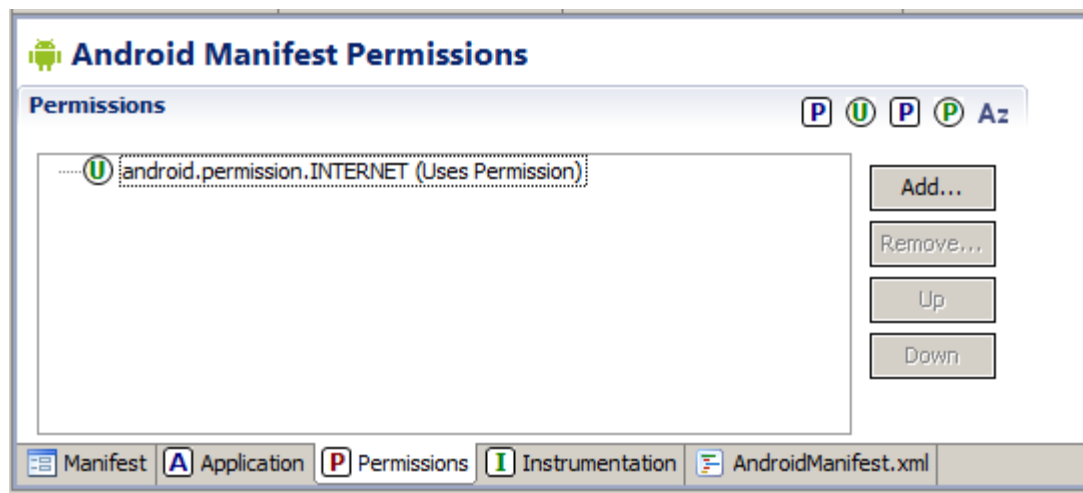
A continuación tenemos la estructura de la interfaz:



PROGRAMACIÓN

Permisos de Internet

Es necesario que se habilite el permiso de internet en el archivo de manifiesto:



MainActivity

Propiedad: onClick

En la propiedad **onClick** de los botones del layout activity_main especifique el método a ejecutar para cada uno de los botones:

Botón	Propiedad	Valor
main_btn_cliente	onClick	onClickCliente
main_btn_movimientos	onClick	onClickMovimientos
main_btn_salir	onClick	onClickSalir

Método: onCreate

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
}
```

Android Avanzado

```
setContentView(R.layout.activity_main);
// Si se trata de Android Versión 3 o superior
StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder()
    .permitAll().build();
StrictMode.setThreadPolicy(policy);
}
```

Método: onClickCliente

```
public void onClickCliente(View view) {
    Intent intent = new Intent(getApplicationContext(), Cliente.class);
    startActivity(intent);
    this.finish();
}
```

Método: onClickMovimientos

```
public void onClickMovimientos(View view) {
    Intent intent = new Intent(getApplicationContext(), Movimientos.class);
    startActivity(intent);
    this.finish();
}
```

Evento: onClickSalir

```
public void onClickSalir(View view) {
    this.finish();
}
```

Clase: Util

```
package pe.egcc.httpclient;

import java.io.IOException;

import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
```

Android Avanzado

```
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.util.EntityUtils;

public class Util {

    public static final String URL_APP =
        "http:// 192.168.1.38:80/androidnivel2/EurekaServices/service/";

    private Util() {
    }

    public static String execJsonGetRequest(String url)
        throws ClientProtocolException, IOException {
        HttpClient httpClient = new DefaultHttpClient();
        HttpGet httpGet = new HttpGet(url);
        httpGet.setHeader("content-type", "application/json");
        HttpResponse response = httpClient.execute(httpGet);
        String jsonResult = EntityUtils.toString(response.getEntity());
        return jsonResult;
    }
}
```

ClienteActivity

Propiedad: onClick

En la propiedad **onClick** de los botones del layout activity_main especifique el método a ejecutar para cada uno de los botones:

Botón	Propiedad	Valor
cliente_btn_consultar	onClick	onClickConsultar
cliente_btn_retornar	onClick	onClickRetornar

Método: onClickConsultar

```
public void onClickConsultar(View view) {
    // Controles
    EditText etCodigo = (EditText) findViewById(R.id.cliente_et_codigo);
    TextView tvResultado = (TextView) findViewById(R.id.cliente_tv_resultado);
    try {
        // Dato
        String codigo = etCodigo.getText().toString();
```

Android Avanzado

```
// Lanzar consulta
String url = Util.URL_APP + "ConsultarCliente.php?codigo=" + codigo;
String jsonResult = Util.execJsonGetRequest(url);
// Procesar el resultado
JSONObject object = new JSONObject(jsonResult);
String estado = object.getString("estado");
if (estado.equals("0")) {
    throw new Exception("Dato incorrecto.");
}
String resultado = "Código: " + object.getString("chr_cliecodigo")
    + "\nPaterno: " + object.getString("vch_cliepaterno")
    + "\nMaterno: " + object.getString("vch_cliematerno")
    + "\nNombre: " + object.getString("vch_clienombre");
tvResultado.setText(resultado);
} catch (Exception e) {
    tvResultado.setText(e.getMessage());
}
}
```

Método: onClickRetornar

```
public void onClickRetornar(View view) {
    Intent intent = new Intent(getApplicationContext(), MainActivity.class);
    startActivity(intent);
    this.finish();
}
```

Prueba



The screenshot shows a mobile application interface. At the top, there is a text input field labeled 'Código' containing the value '00001'. Below the input field are two buttons: 'Consultar' (highlighted in blue) and 'Retornar' (grey). Below the buttons, the search results are displayed in a list format: 'Código: 00001', 'Paterno: CORONEL', 'Materno: CASTILLO', and 'Nombre: ERIC GUSTAVO'.

Programando el layout_item

```
package pe.egcc.httpclient;
```

```
import java.util.HashMap;

import android.content.Context;
import android.widget.LinearLayout;
import android.widget.TextView;

public class ItemView extends LinearLayout {

    private TextView tvFecha;
    private TextView tvTipo;
    private TextView tvImporte;

    public ItemView(Context context) {
        super(context);
        inflate(context, R.layout.layout_item, this);
        // Controles
        tvFecha = (TextView) findViewById(R.id.item_tv_fecha);
        tvTipo = (TextView) findViewById(R.id.item_tv_tipo);
        tvImporte = (TextView) findViewById(R.id.item_tv_importe);
    }

    public void setItem(HashMap<String, String> item) {
        tvFecha.setText(item.get("fecha").toString());
        tvTipo.setText(item.get("tipo").toString());
        tvImporte.setText(item.get("importe").toString());
    }
}
```

Programando el adaptador

```
package pe.egcc.httpclient;

import java.util.ArrayList;
import java.util.HashMap;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;

public class ItemAdapter extends BaseAdapter {

    private ArrayList<HashMap<String, String>> datos;
```

Android Avanzado

```
ItemAdapter(ArrayList<HashMap<String, String>> datos) {
    this.datos = datos;
    notifyDataSetChanged();
}

@Override
public int getCount() {
    return datos.size();
}

@Override
public Object getItem(int position) {
    return datos.get(position);
}

@Override
public long getItemId(int position) {
    return 0;
}

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    ItemView view;
    if (convertView == null) //NO existe, creamos uno
        view = new ItemView(parent.getContext());
    else //Existe, reutilizamos
        view = (ItemView) convertView;
    view.setItem((HashMap<String, String>) datos.get(position));
    return view;
}
```

MovimientosActivity

Código

```
package pe.egcc.httpclient;

import java.util.ArrayList;
import java.util.HashMap;

import org.json.JSONArray;
```


Android Avanzado

```
import org.json.JSONObject;

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;

public class MovimientosActivity extends Activity {

    // JSON Node names
    private static final String TAG_DATE = "fecha";
    private static final String TAG_TYPE = "tipo";
    private static final String TAG_AMOUNT = "importe";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_movimientos);
    }

    public void onClickConsultar(View view) {
        // Controles
        EditText etCuenta = (EditText) findViewById(R.id.movimientos_et_cuenta);
        TextView tvResultado = (TextView)
            findViewById(R.id.movimientos_tv_resultado);
        ListView lvDatos = (ListView) findViewById(R.id.movimientos_lv_lista);
        try {
            // Dato
            String cuenta = etCuenta.getText().toString();
            // Lanzar consulta
            String url = Util.URL_APP + "ConsultarMovimientos.php?cuenta="
                + cuenta;
            String jsonResult = Util.execJsonGetRequest(url);
            if (jsonResult.isEmpty()) {
                throw new Exception("Cuenta no existe.");
            }
            // Procesar el resultado
            JSONObject json = new JSONObject(jsonResult);
            // Obtener el elemento que contiene los movimientos
            JSONArray jArray = json.getJSONArray("movimientos");
            // Hashmap for ListView
            ArrayList<HashMap<String, String>> movimientos =
```

Android Avanzado

```
        new ArrayList<HashMap<String, String>>();
        // Loop the Array
        for (int i = 0; i < jsonArray.length(); i++) {
            HashMap<String, String> map = new HashMap<String, String>();
            JSONObject e = jsonArray.getJSONObject(i);
            map.put(TAG_DATE, e.getString(TAG_DATE));
            map.put(TAG_TYPE, e.getString(TAG_TYPE));
            map.put(TAG_AMOUNT, String.valueOf(e.getDouble(TAG_AMOUNT)));
            movimientos.add(map);
        }
        // Adaptandolo al ListView
        lvDatos.setAdapter(new ItemAdapter(movimientos));
    } catch (Exception e) {
        tvResultado.setText(e.getMessage());
        // throw e;
    }
}

public void onClickRetornar(View view) {
    Intent intent = new Intent(getApplicationContext(), MainActivity.class);
    startActivity(intent);
    this.finish();
}
}
```

Prueba

