

MySQL, PHP y otras historias



# MySQL, PHP y otras historias

## Planteamiento

*Un cliente desea consultar sus cuentas del banco.*

*Pero no tiene cerca una sucursal bancaria.*

*Decide usar su dispositivo móvil para hacer la consulta.*

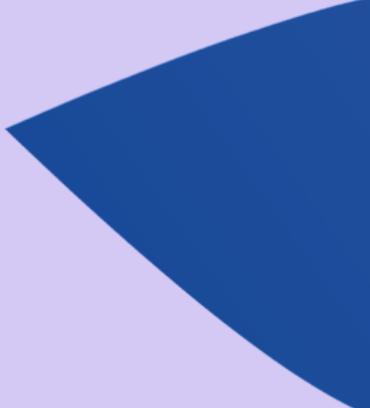
*Voy a usar este dispositivo móvil para consultar los numeros de mis cuentas*

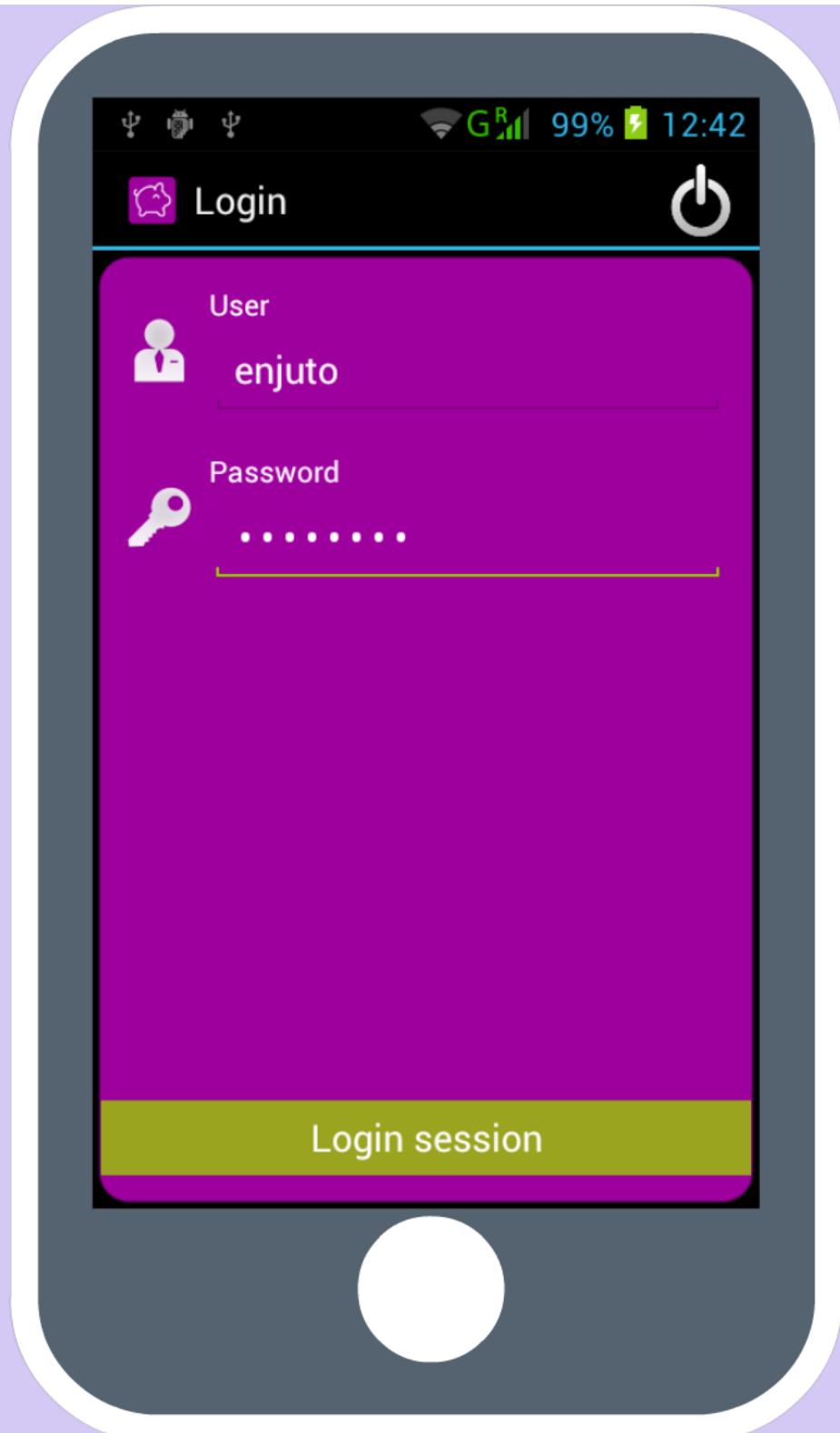


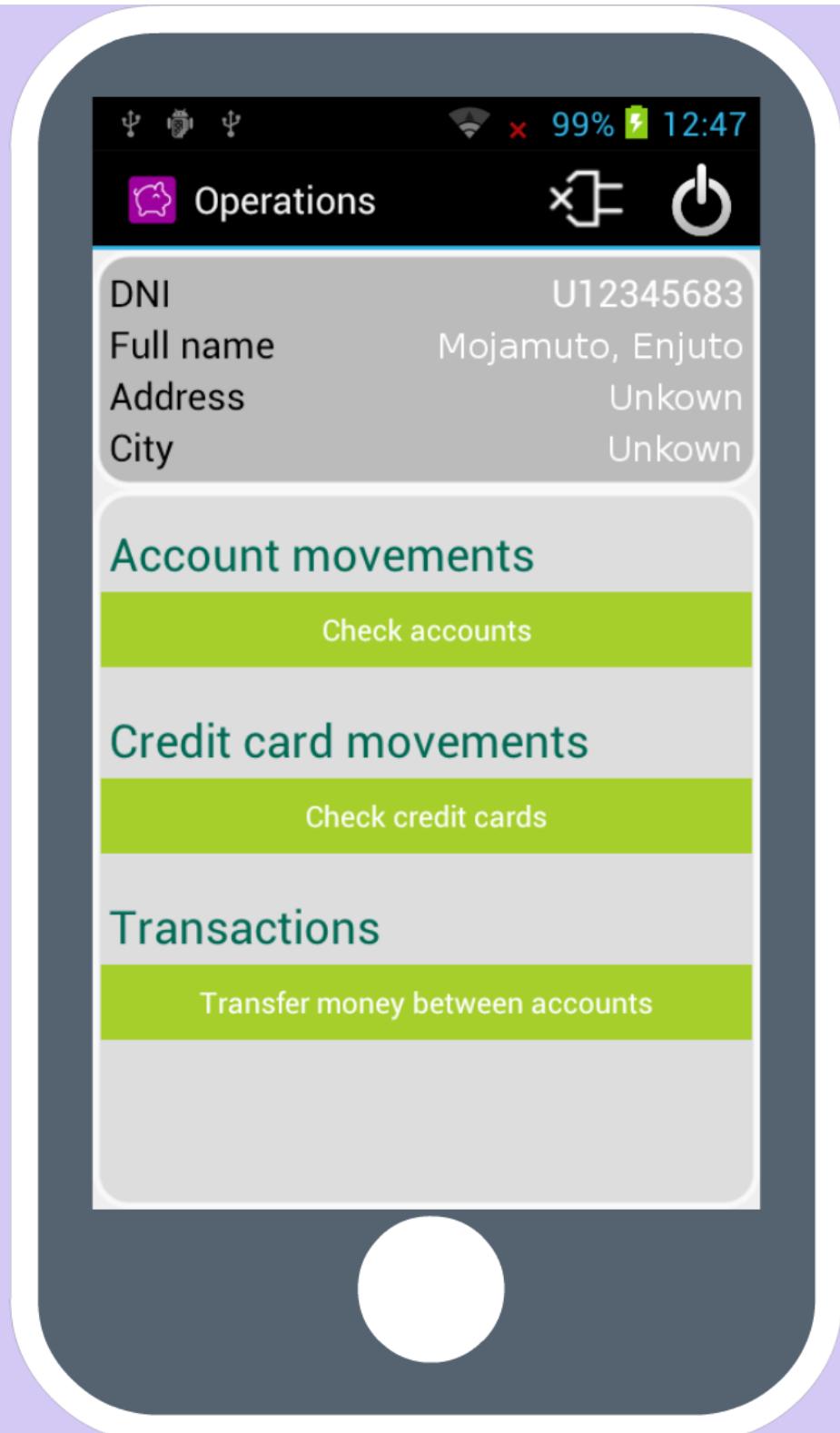
# BankAPP

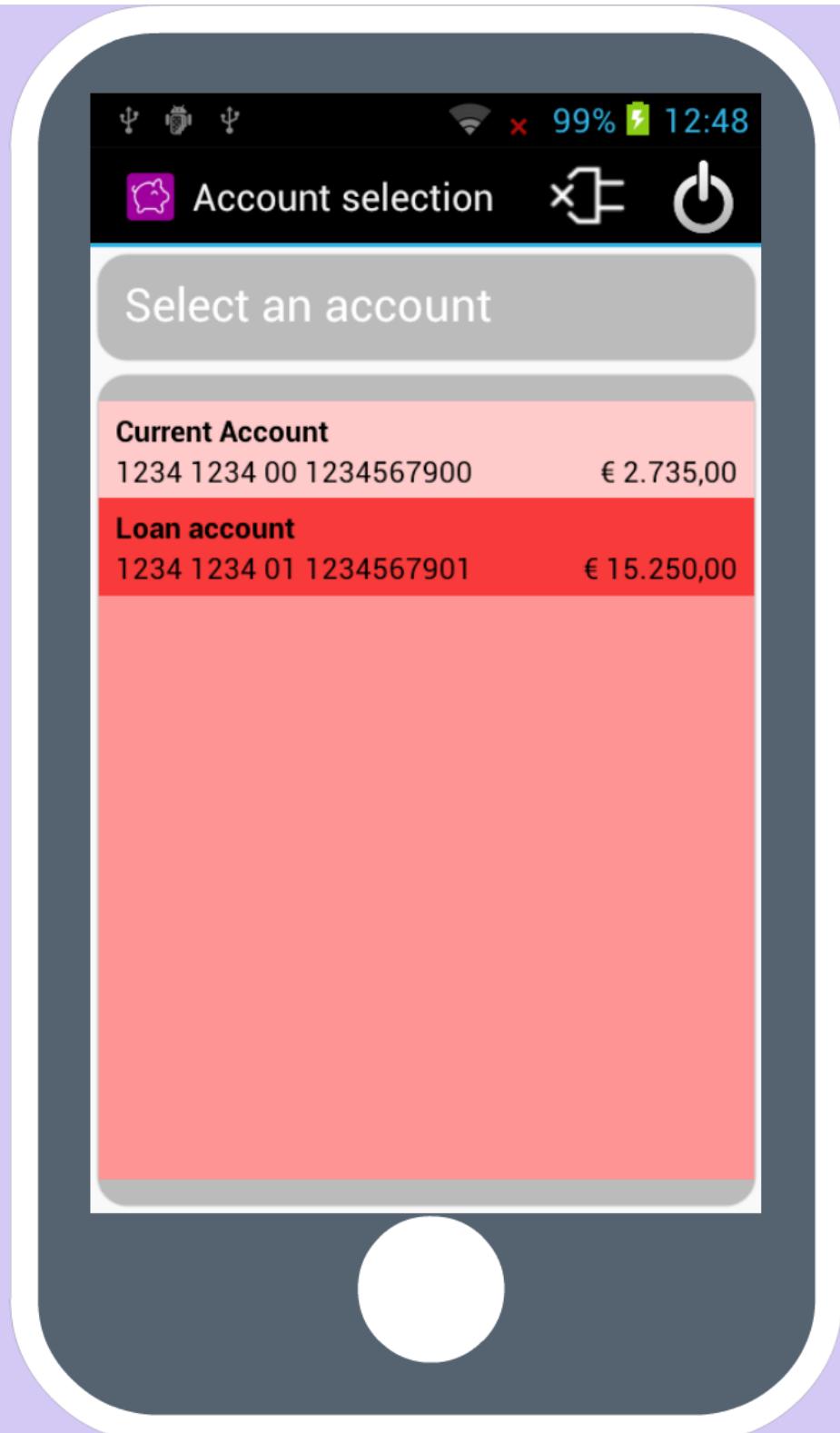


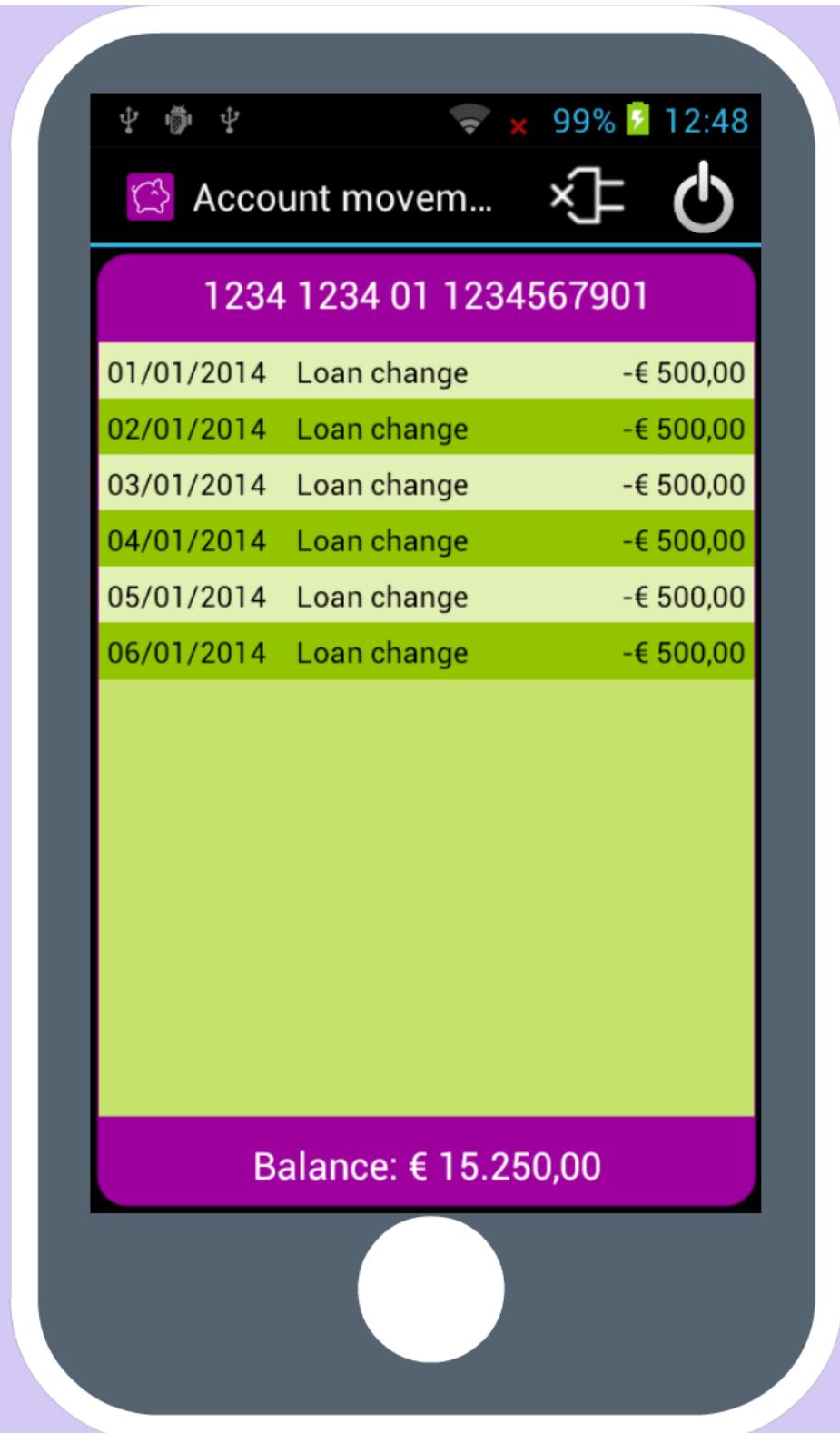
banca electrónica









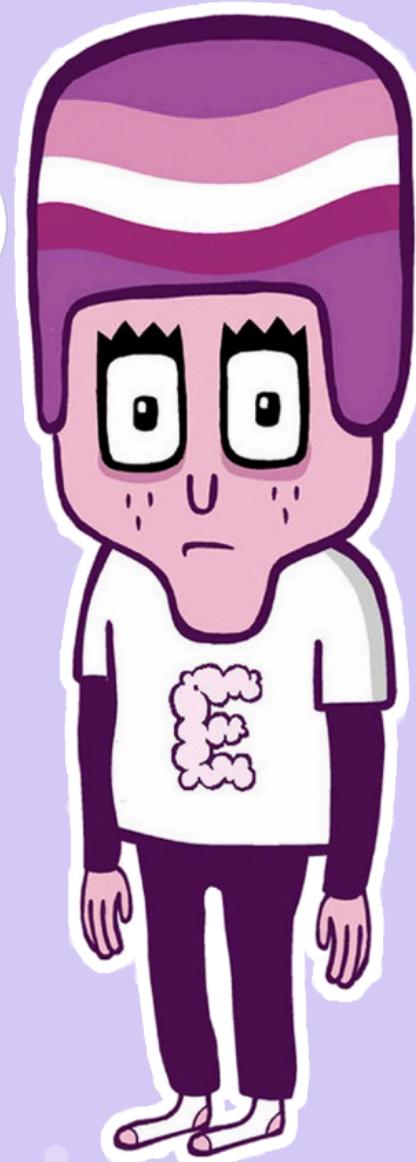


*Madre mía  
to lo que he hecho y  
no me he enterao de na...*

*Me voy a comprar un router que  
estoy podridico de  
pasta*

*Efectivamente Enjuto ha consultado con su banco  
varios datos sobre él y sus cuentas pero realmente  
no sabe lo que hay detrás de ese simple gesto.*

*Ahora viene la explicación técnica...*



# Para la consulta anterior necesitamos

- Un cliente
  - \* *APP Android en Java*
- Una red de conexión
- Un servidor
  - \* *Una base de datos (MySQL)*
  - \* *Un servidor web (Apache)*
  - \* *Scripts de acceso a datos (PHP)*

# Cliente

*Realiza una petición GET/POST*

# Red

*Transmite la información al servidor*

# Servidor

*Recibe la petición GET / POST*

*Ejecuta el script PHP de consulta a la base de datos*

*Devuelve la información, normalmente en forma de JSON*

# Red

*Transmite la información al cliente*

# Cliente

*Recibe la información, en formato JSON*

*Interpreta el JSON*

*Lo presenta en la forma adecuada  
o realiza operaciones con él*

Account selection		
Select an account		
<b>Current Account</b>	1234 1234 16 1234567916	€ 9.876,00
<b>Loan account</b>	1234 1234 17 1234567917	€ 357.256,00

# Formación de una petición GET

*Es necesario lo siguiente*

*La IP del servidor*

<http://10.2.253.131/>

*El script que se ejecutará*

*phps/c\_u\_p.php*

*En su caso, el paso de argumentos*

*?user=user18&psw=psw*

# ¿ Cómo hacerlo desde Android ?

```
102    /**
103     * Comprueba si el usuario y la contraseña dados son correctos. En caso de
104     * serlo devuelve el ID del usuario electrónico en la base de datos, en caso
105     * contrario devuelve -1
106     *
107     * @param user
108     *          - cadena con el nombre del usuario
109     * @param psw
110     *          - cadena con la contraseña
111     * @return - un entero ID
112     */
113 private int check_usr_psw(String user, String psw) {
114     int id_usuario = -1;
115
116     ArrayList<NameValuePair> argumentos = new ArrayList<NameValuePair>();
117
118     argumentos.add(new BasicNameValuePair("user", user));
119     argumentos.add(new BasicNameValuePair("psw", psw));
120
121     JSONParser jSp = new JSONParser();
122     JSONObject jSo = new JSONObject();
123     jSo = jSp.makeHttpRequest(URL_connect, "GET", argumentos);
124
125     try {
126         id_usuario = jSo.getInt("id_usr");
127     } catch (JSONException e) {
128         // TODO Auto-generated catch block
129         e.printStackTrace();
130     }
131
132     return id_usuario;
133 }
```

Creo un ArrayList de tipo Nombre-Valor  
Añado las parejas de nombre valor  
Declaro el interpretador y un objeto JSON que va a recibir la respuesta del servidor  
Hago la petición con la dirección, el método y los argumentos  
Finalmente recojo el id del usuario en el objeto JSON declarado antes

```
1 <?php
2
3     /*
4      * Compruebo el usuario y la contraseña dados contra la base de datos
5     */
6
7     // array para volcar el resultado
8     $respuesta = array();
9     // Fijo estos dos valores
10    $respuesta["id_usr"] = -1;
11    $respuesta["exito"] = 0;
12    $respuesta["mensaje"] = "Contraseña y/o usuario incorrectos";
13
14    // llamo a la conexión de la base de datos
15    require_once __DIR__ . '/db_connect.php';
16
17    // realizo la conexión
18    $db = new DB_CONNECT();
19
20    // compruebo los parámetros recibidos
21    if (isset($_GET["user"]) && isset($_GET["psw"])) {
22        $user = $_GET['user'];
23        $psw = $_GET['psw'];
24
25        $resultado = mysql_query("SELECT * FROM Electronic_user "
26                                . "WHERE User='{$user}' and Password='{$psw}'") or die(mysql_error());
27
28        // Compruebo si la consulta devuelve algo
29        if (!empty($resultado)) {
30            // Compruebo si ha devuelto una fila
31            if (mysql_num_rows($resultado) == 1) {
32                // Vuelco el resultado sobre la misma variable
33                $resultado = mysql_fetch_array($resultado);
34
35                // Extraigo el valor de la columna
36                $id_usr = $resultado["ID_Client"];
37
38                $respuesta["id_usr"] = (int) $id_usr;
39                $respuesta["exito"] = 1;
40                $respuesta["mensaje"] = "OK";
41
42                echo json_encode($respuesta);
43            } else {
44                // contraseña y/o usuario incorrectos
45                echo json_encode($respuesta);
46            }
47        } else {
48            // contraseña y/o usuario incorrectos
49            echo json_encode($respuesta);
50        }
51    } else {
52        // falta un parámetro
53        $respuesta["mensaje"] = "Falta un argumento en la consulta";
54
55        // echoing JSON respuesta
56        echo json_encode($respuesta);
57    }
58 ?>
```

```
// Fijo estos dos valores
$respuesta["id_usr"] = -1;
$respuesta["exito"] = 0;
$respuesta["mensaje"] = "Contraseña y/o usuau

// llamo a la conexión de la base de datos
require_once __DIR__ . '/db_connect.php';

// realizo la conexión
$db = new DB_CONNECT();

// compruebo los parámetros recibidos
if (isset($_GET["user"]) && isset($_GET["psw"]))
    $user = $_GET['user'];
    $psw = $_GET['psw'];
```

```
// llamo a la conexión de la base de datos
require_once __DIR__ . '/db_connect.php';

// realizo la conexión
$db = new DB_CONNECT();

// compruebo los parámetros recibidos
if (isset($_GET["user"]) && isset($_GET["psw"])) {
    $user = $_GET['user'];
    $psw = $_GET['psw'];

    $resultado = mysql_query("SELECT * FROM Electrodomesticos
        . WHERE User='{$user}' and Password='{$psw}'");

    // Compruebo si la consulta devuelve algo
    if (!empty($resultado)) {
        // Compruebo si ha devuelto una fila
        if (mysql_num_rows($resultado) == 1) {
            // Si es correcto, redirijo al login
            header('Location: login.php');
        } else {
            // Si no es correcto, muestro un error
            echo "Error: Los datos no son correctos";
        }
    }
}
```

```
spuesta["mensaje"] = "Contraseña y/o usuario incorrectos";
```

llamo a la conexión de la base de datos

```
uire_once __DIR__ . '/db_connect.php';
```

realizo la conexión

```
= new DB_CONNECT();
```

compruebo los parámetros recibidos

```
(isset($_GET["user"]) && isset($_GET["psw"])) {
```

```
$user = $GET['user'];
```

```
$psw = $_GET['psw'];
```

```
$resultado = mysql_query("SELECT * FROM Electronic_user '");
```

```
." WHERE User='".$user."' and Password='".$psw."') or die(mysql_error());
```

Compruebo si la consulta devuelve algo

```
if (!empty($resultado)) {
```

Compruebo si ha devuelto una fila

```
if (mysql_num_rows($resultado) == 1) {
```

Vuelco el resultado sobre la misma variable

```
$resultado = mysql_fetch_array($resultado)
```

Extraigo el valor de la columna

```
$id_usr = $resultado["ID Client"];
```

```
$respuesta["id_usr"] = (int) $id_usr;
```

```
$respuesta["exito"] = 1;
```

```
3 // Compruebo si la consulta devuelve algo
4     if (!empty($resultado)) {
5 // Compruebo si ha devuelto una fila
6         if (mysql_num_rows($resultado) == 1) {
7 // Vuelco el resultado sobre la misma variable
8             $resultado = mysql_fetch_array($resultado);
9
10 // Extraigo el valor de la columna
11     $id_usr = $resultado["ID_Client"];
12
13     $respuesta["id_usr"] = (int) $id_usr;
14     $respuesta["exito"] = 1;
15     $respuesta["mensaje"] = "OK";
16
17     echo json_encode($respuesta);
18 } else {
19 // contraseña y/o usuario incorrectos
20     echo json_encode($respuesta);
21 }
22 } else {
23 // contraseña y/o usuario incorrectos
24     echo json_encode($respuesta);
```

# Formato JSON que devuelve el script PHP

```
{  
    "id_usr":18, ← clave: id_usr,     valor: (int) 18  
    "exito":1,   ← clave: exito,      valor: (int) 1  
    "mensaje":"OK" ← clave: mensaje, valor: (String) OK  
}  
  
{  
    "exito":1,  
    "mensaje":"El cliente tiene cuentas",  
    "cuentas":  
        [  
            {  
                "id_cuenta":29,  
                "iban":"12341234181234567918",  
                "balance":9837,  
                "tipo_cuenta":"Current Account"},  
            {  
                "id_cuenta":30,  
                "iban":"12341234191234567919",  
                "balance":12312,  
                "tipo_cuenta":"Loan account"  
            }]  
}
```

## Otro ejemplo

# Interpretación del objeto JSON en Android

```
112*   /**
113 * Recupera la información de las cuentas del cliente, concretamente el IBAN
114 * y el saldo
115 */
116 private AccountManager accounts_data() {
117     AccountManager acc_man_temp = new AccountManager();
118     Account account = null;
119     ArrayList<NameValuePair> argumentos = new ArrayList<NameValuePair>();
120
121     argumentos.add(new BasicNameValuePair("id_client", String
122         .valueOf(id_usuario)));
123
124     JSONParser jSp = new JSONParser();
125     JSONObject jSo = new JSONObject();
126     // Array para volcar los datos del cliente
127     JSONArray jSa = null;
128     jSo = jSp.makeHttpRequest(URL_connect, "GET", argumentos);
129
130     try {
131         // Vuelco los datos del cliente
132         jSa = jSo.getJSONArray("cuentas");
133         int num_cuentas = jSa.length();
134
135         for (int i = 0; i < num_cuentas; i++) {
136             // Creo un objeto a partir de este para extraer los datos
137             JSONObject jSAccount = jSa.getJSONObject(i);
138             // Creo un nuevo objeto Account con los datos extraidos
139             account = new Account(jSAccount.getInt("id_cuenta"),
140                 jSAccount.getString("iban"),
141                 jSAccount.getDouble("balance"),
142                 jSAccount.getString("tipo_cuenta"));
143
144             // Lo añado a la lista de cuentas del cliente
145             acc_man_temp.addAccount(account);
146         }
147
148     } catch (JSONException e) {
149         // Log.e("ERROR", "Problema con el JSON");
150         e.printStackTrace();
151     }
152     return acc_man_temp;
153 }
```

```
112     /**
113      * Recupera la información de las cuentas del cliente, concretamente el IBAN
114      * y el saldo
115      */
116     private AccountManager accounts_data() {
117         AccountManager acc_man_temp = new AccountManager();
118         Account account = null;
119         ArrayList<NameValuePair> argumentos = new ArrayList<NameValuePair>();
120
121         argumentos.add(new BasicNameValuePair("id_client", String
122                                         .valueOf(id_usuario)));
123
124         JSONParser jSp = new JSONParser();
125         JSONObject jSo = new JSONObject();
126         // Array para volcar los datos del cliente
127         JSONArray jSa = null;
128         jSo = jSp.makeHttpRequest(URL_connect, "GET", argumentos);
129
130         try {
131             // Vuelco los datos del cliente
132             jSa = jSo.getJSONArray("cuentas");
133             int num_cuentas = jSa.length();
134
135             for (int i = 0; i < num_cuentas; i++) {
136                 // Creo un objeto a partir de este para extraer los datos
137                 JSONObject jSAccount = jSa.getJSONObject(i);
138                 // Creo un nuevo objeto Account con los datos extraidos
139                 account = new Account(jSAccount.getInt("id_cuenta"),
140                                       jSAccount.getString("iban"),
141                                       jSAccount.getDouble("balance"),
142                                       jSAccount.getString("tipo_cuenta"));
143             }
144         } catch (JSONException e) {
145             e.printStackTrace();
146         }
147     }
```

```
122     .valueOf(id_usuario));
123
124     JSONParser jSp = new JSONParser();
125     JSONObject jSo = new JSONObject();
126     // Array para volcar los datos del cliente
127     JSONArray jSa = null;
128     jSo = jSp.makeHttpRequest(URL_connect, "GET", argumentos);
129
130    try {
131        // Vuelco los datos del cliente
132        jSa = jSo.getJSONArray("cuentas");
133        int num_cuentas = jSa.length();
134
135        for (int i = 0; i < num_cuentas; i++) {
136            // Creo un objeto a partir de este para extraer los datos
137            JSONObject jSAccount = jSa.getJSONObject(i);
138            // Creo un nuevo objeto Account con los datos extraidos
139            account = new Account(jSAccount.getInt("id_cuenta"),
140                jSAccount.getString("iban"),
141                jSAccount.getDouble("balance"),
142                jSAccount.getString("tipo_cuenta"));
143
144            // Lo añado a la lista de cuentas del cliente
145            acc_man_temp.addAccount(account);
146        }
147
148    } catch (JSONException e) {
149        // Log.e("ERROR", "Problema con el JSON");
150        e.printStackTrace();
151    }
152    return acc_man_temp;
}
```

# Diferencias entre GET y POST

*El concepto **GET** es obtener información del servidor. Traer datos que están en el servidor, ya sea en un archivo o base de datos, al cliente. Independientemente de que para eso tengamos que enviar (request) algún dato que será procesado para luego devolver la respuesta (response) que esperamos, como por ejemplo un identificador para obtener una noticia de la base de datos.*

***POST** sin embargo es enviar información desde el cliente para que sea procesada y actualice o agregue información en el servidor, como sería la carga o actualización en sí de una noticia. Cuando enviamos (request) datos a través de un formulario, estos son procesados y luego a través de una redirección, por ejemplo, devolvemos (response) alguna página con información.*

# Breve explicación sobre AsyncTask

*Permite ejecutar una tarea en segundo plano, es decir podemos ejecutar tareas asíncronas.*

*Se divide en 4 etapas:*

## *onPreExecute()*

*Normalmente se usa para configurar la tarea.*

## *dolnBackground(Tipo... args)*

*Es el bloque principal, ejecuta la tarea encomendada. En el ejemplo acceder al servidor, realizar la consulta y devolver los datos obtenidos.*

## *onProgressUpdate(Tipo... progreso)*

*Se dispara cuando se hace una llamada publishProgress(Progress...) desde dolnBackground, permite actualizar partes de la UI. Por ejemplo, una barra de progreso.*

## *onPostExecute(Resultado)*

*Recibe el resultado de la tarea ejecutada en dolnBackground como argumento*

# Prueba libre de la app

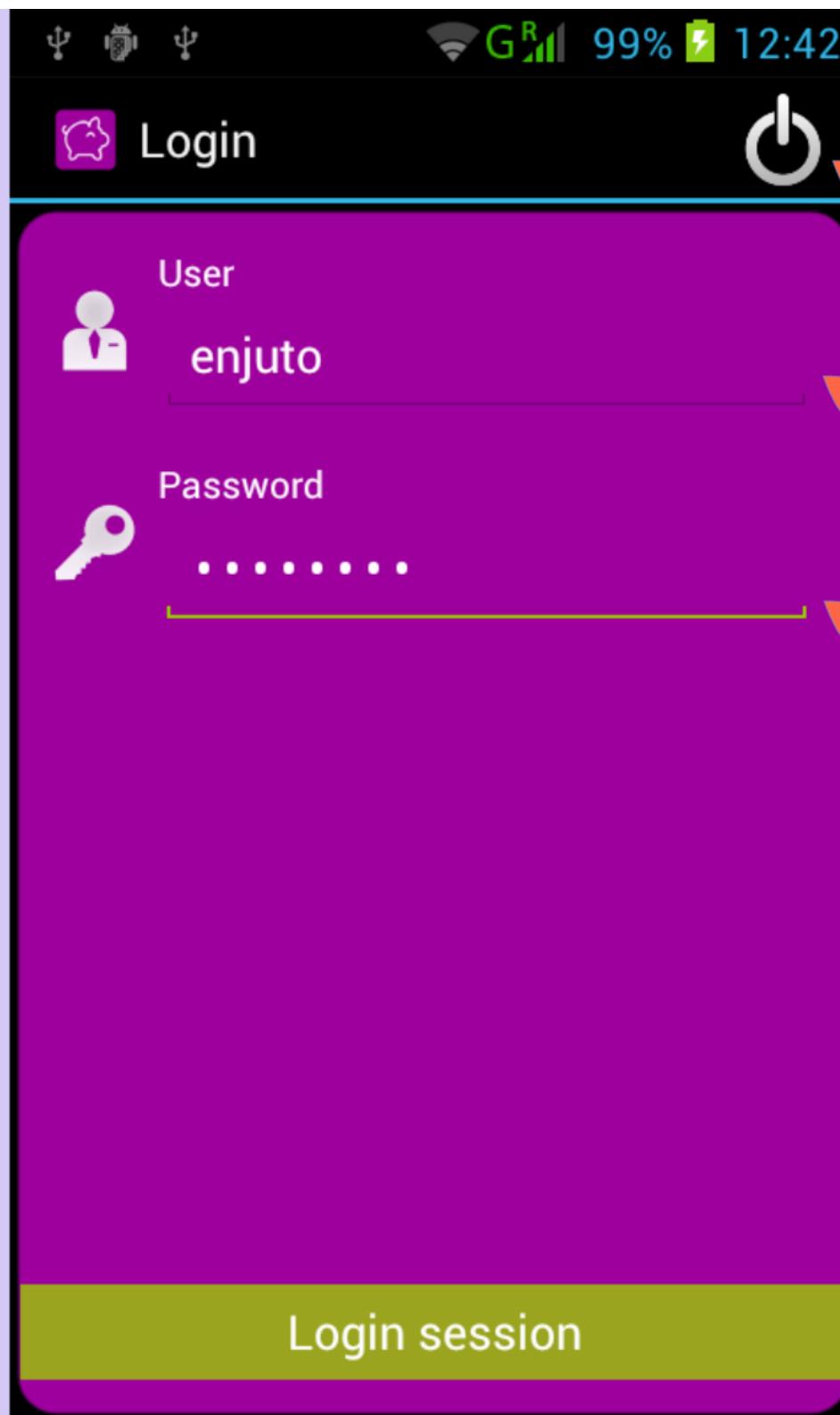
Usuarios

user + número del 1 al 20

Contraseña

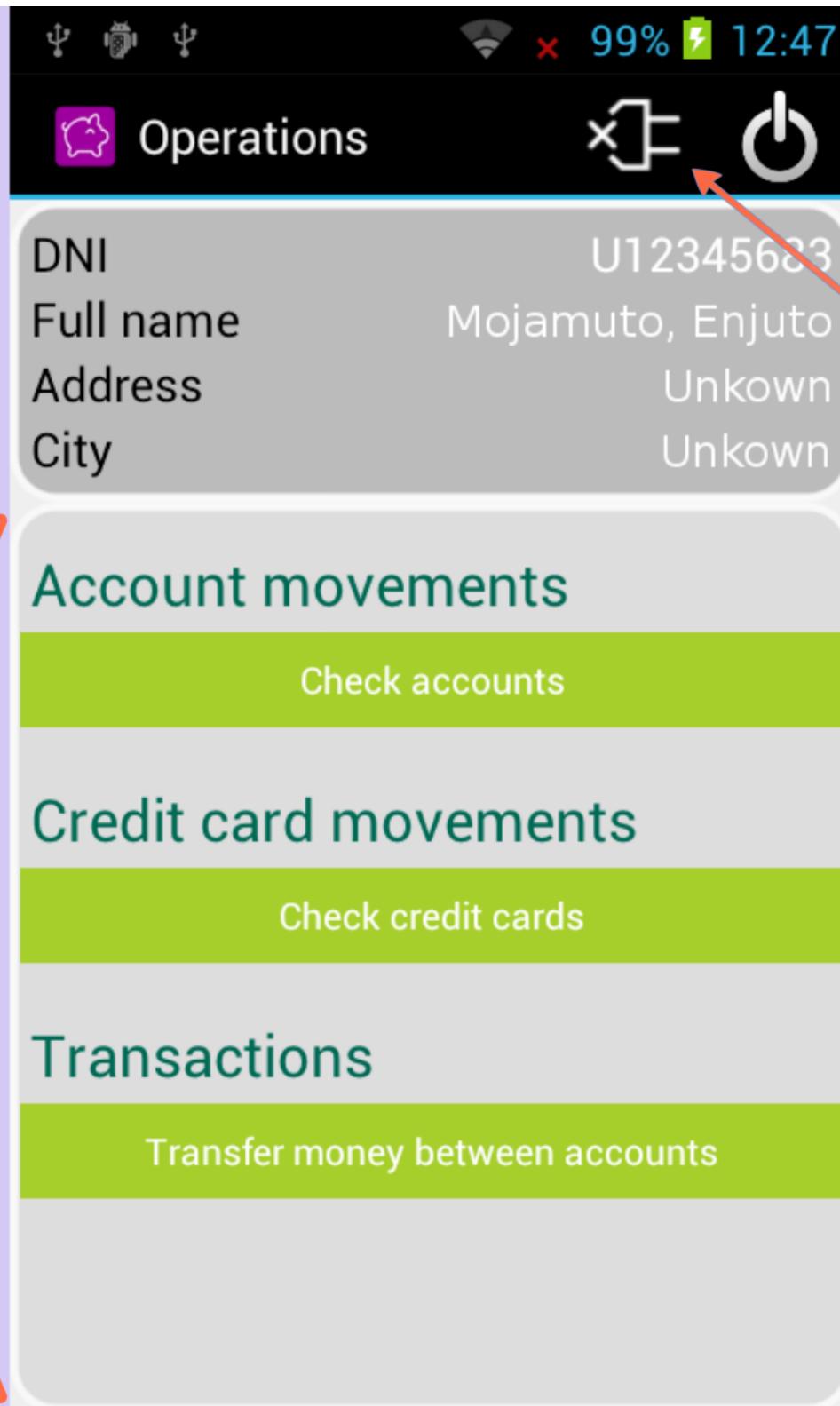
psw

nota: el usuario enjuto tiene restringido el acceso  
por un problema de phising



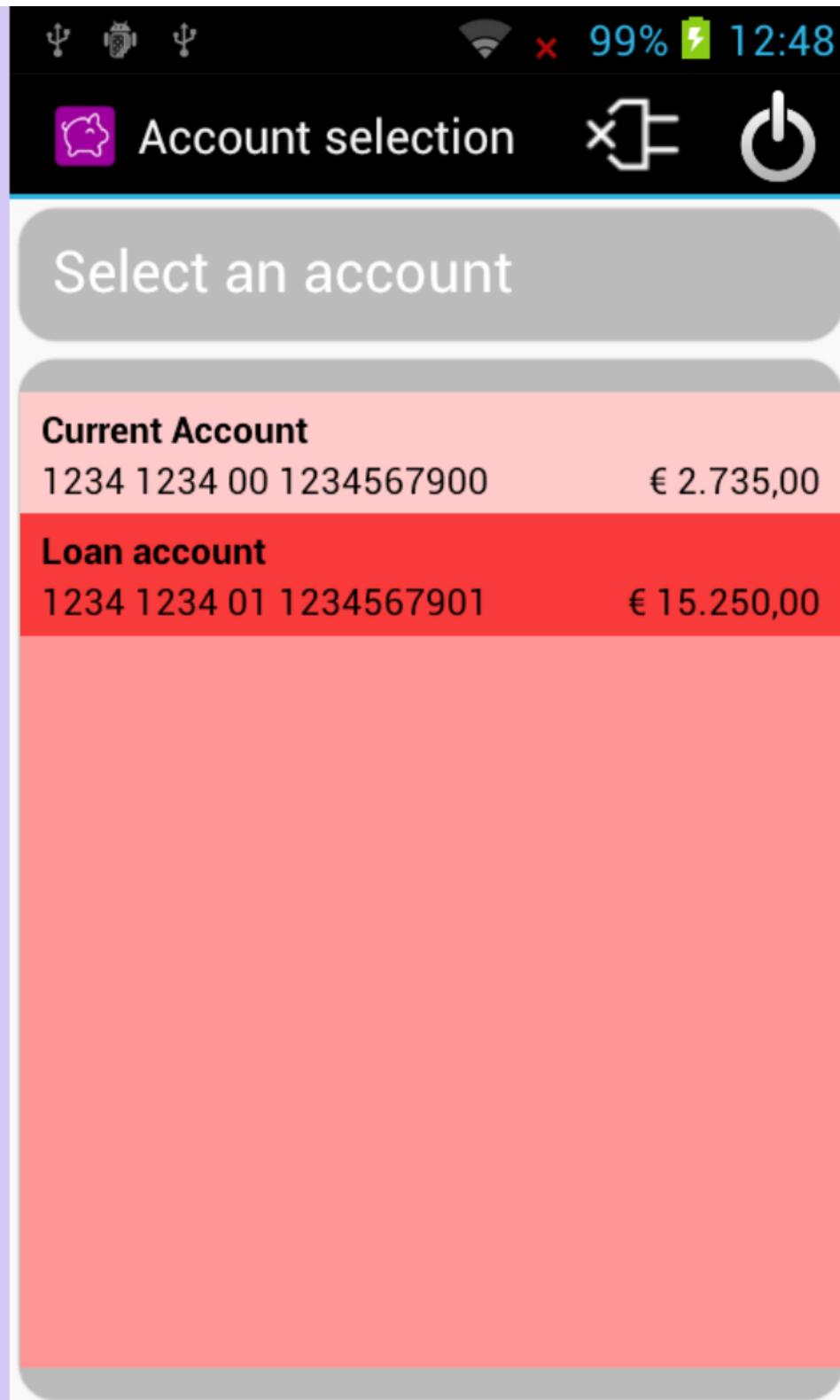
Información  
del usuario

Operaciones



Cierre de sesión

Listado de  
cuentas  
del usuario



Nº de cuenta

1234 1234 01 1234567901

Detalles de  
movimientos  
de la cuenta  
seleccionada

01/01/2014	Loan change	-€ 500,00
02/01/2014	Loan change	-€ 500,00
03/01/2014	Loan change	-€ 500,00
04/01/2014	Loan change	-€ 500,00
05/01/2014	Loan change	-€ 500,00
06/01/2014	Loan change	-€ 500,00

Saldo

Balance: € 15.250,00



99%



12:48



Account movem...



# Fuentes consultadas

## *Instalacion LAMP*

<https://www.digitalocean.com/community/articles/how-to-install-linux-apache-mysql-php-lamp-stack-on-centos-6>

## *MySQL, JSON Parser, PHP, conexión desde Android*

<http://www.androidhive.info/2012/05/how-to-connect-android-with-php-mysql/>

## *Estilos, duda y fuente principal*

<http://developer.android.com/>

## *PHP*

<http://www.php.net/>

## *AsyncTask*

<http://developer.android.com/reference/android/os/AsyncTask.html>

## *Diferencias entre GET y POST*

[http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=581:request-php-diferencias-ventajas-y-problemas-de-get-post-request-ejercicios-ejemplos-cu00832b-&catid=70:tutorial-basico-programador-web-php-desde-cero&Itemid=193](http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=581:request-php-diferencias-ventajas-y-problemas-de-get-post-request-ejercicios-ejemplos-cu00832b-&catid=70:tutorial-basico-programador-web-php-desde-cero&Itemid=193)

# Software utilizado

## *VirtualBox*

<https://www.virtualbox.org/>

## *MySQL*

<http://www.mysql.com/>

## *Máquina virtual CentOS*

<https://s3-eu-west-1.amazonaws.com/virtualboxes.org/CentOS-6.5-i386-minimal.ova.torrent>

## *Android Developer Tools*

<http://developer.android.com/tools/index.html>

## *Apache server*

<http://httpd.apache.org/>

## *php*

<http://www.php.net/>

## *LibreOffice*

<http://es.libreoffice.org/>

## *Iconos de la aplicación*

<http://www.icojam.com/blog/?p=589>