

Introducción a la programación de Dispositivos Móviles.

Dr. Mauricio Arroqui

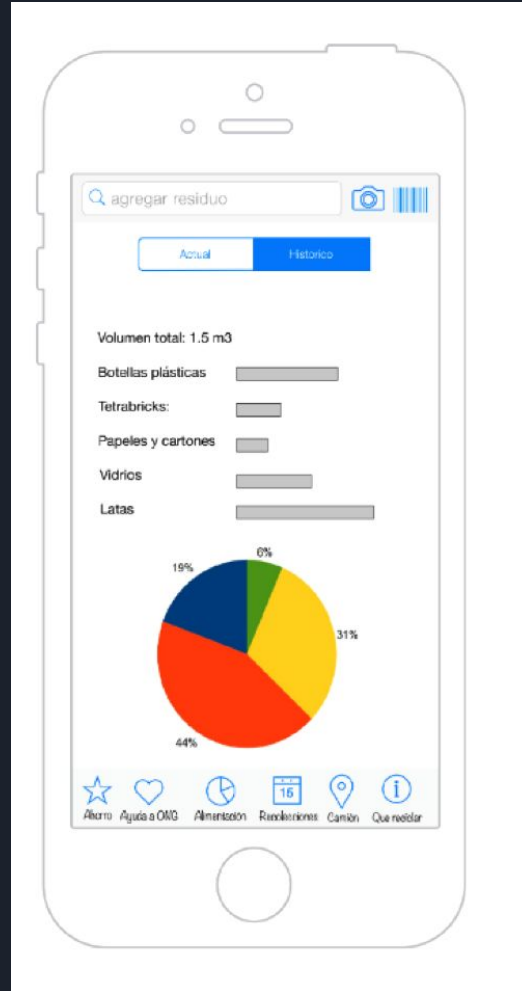
Dr. Juan Manuel Rodriguez

Ing. Juan Maximiliano Rodriguez



FINAL

- Registro de un usuario
- Almacenar reciclados
 - Botellas, tetrabricks, papeles, vidrios, latas.
- Submitear un reciclado
- Mostrar listas de reciclados
- Mostrar un total de todo lo reciclado.



Conectarse a Internet



Pasos para conectarse

- Agregar permisos a Manifiesto de Android
- Verificar la conexión de red
- Crear un WorkerThread
- Implementar tarea en background
 - Crear URI
 - Hacer una conexión HTTP
 - Conectar y GET Datos
- Procesar resultados
- Parsear resultados



Permiso en el AndroidManifest

Internet

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

Check Network State

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



Obtener información de la Red

- ConnectivityManager
 - Responde sobre el estado de la conexión
 - Notifica cuando la conectividad cambia
- NetworkInfo
 - Describe el estado de una interfaz de red
 - Mobile or Wi-Fi



Chequear si la red está disponible

```
ConnectivityManager connMgr = (ConnectivityManager)
    getSystemService(Context.CONNECTIVITY_SERVICE);

NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();

if (networkInfo != null && networkInfo.isConnected()) {
    // Create background thread to connect and get data
    new DownloadWebpageTask().execute(stringUrl);
} else {
    textView.setText("No network connection available.");
}
```



WiFi y Mobile

```
NetworkInfo networkInfo =  
    connMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI);  
boolean isWifiConn = networkInfo.isConnected();  
  
networkInfo =  
    connMgr.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);  
boolean isMobileConn = networkInfo.isConnected();
```


Worker Thread





Use Worker Thread

- [AsyncTask](#)—tareas cortas, no resultados a la UI
- [AsyncTaskLoader](#)—tareas largas, resultados a la UI
- [Background Service](#)



En Background

En la background task (ejemplo `doInBackground()`)

- Crear URI
- Hacer HTTP Connection
- Download Datos



URI = Uniform Resource Identifier

String that names or locates a particular resource

- file://
- http:// and https://
- content://



Ejemplo URL para Google Books API

[https://www.googleapis.com/books/v1/volumes?
q=pride+prejudice&maxResults=5&printType=books](https://www.googleapis.com/books/v1/volumes?q=pride+prejudice&maxResults=5&printType=books)

Constantes como Parametros

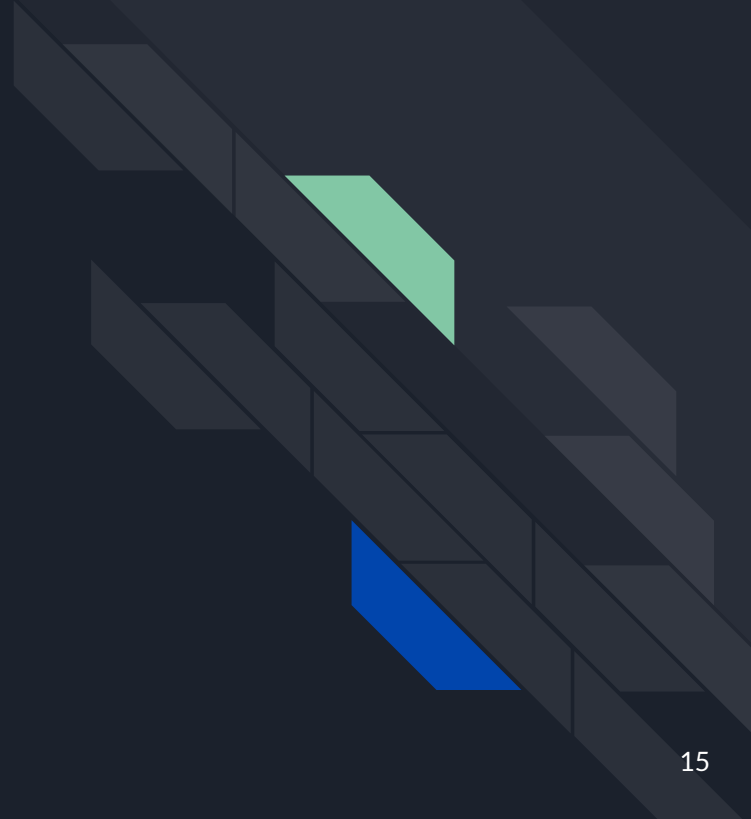
```
final String BASE_URL =  
    "https://www.googleapis.com/books/v1/volumes?";  
  
final String QUERY_PARAM = "q";  
  
final String MAX_RESULTS = "maxResults";  
  
final String PRINT_TYPE = "printType";
```



Crear a URI programaticamente

```
Uri builtURI = Uri.parse(BASE_URL).buildUpon()  
    .appendQueryParameter(QUERY_PARAM, "pride+prejudice")  
    .appendQueryParameter(MAX_RESULTS, "10")  
    .appendQueryParameter(PRINT_TYPE, "books")  
    .build();  
  
URL requestURL = new URL(builtURI.toString());
```

HTTP Client Connection





Conectarse a una HTTP

- Usar [HttpURLConnection](#)
- Tiene que ser en otro Thread
- Requiere InputStreams y try/catch blocks



Crear una HttpURLConnection

```
HttpURLConnection conn =  
    (HttpURLConnection) requestURL.openConnection();
```



Configurar connection

```
conn.setReadTimeout(10000 /* milliseconds */);  
conn.setConnectTimeout(15000 /* milliseconds */);  
conn.setRequestMethod("GET");  
conn.setDoInput(true);
```



Connect y get response

```
conn.connect();  
int response = conn.getResponseCode();  
  
InputStream is = conn.getInputStream();  
String contentAsString = convertIsToString(is, len);  
return contentAsString;
```



Cerrar la conexión y el stream

```
} finally {  
    conn.disconnect();  
    if (is != null) {  
        is.close();  
    }  
}
```

Convertir la respuesta
en un String





Convertir input stream en un string

```
public String convertIsToString(InputStream stream, int len)
    throws IOException, UnsupportedEncodingException {

    Reader reader = null;
    reader = new InputStreamReader(stream, "UTF-8");
    char[] buffer = new char[len];
    reader.read(buffer);
    return new String(buffer);
}
```



BufferedReader

```
StringBuilder builder = new StringBuilder();  
BufferedReader reader =  
    new BufferedReader(new  
    InputStreamReader(inputStream));  
String line;  
while ((line = reader.readLine()) != null) {  
    builder.append(line + "\n");  
}
```



Libraries HTTP Client Connection



Hacer una conexión utilizando librerías

- Usando librerías de terceras partes [OkHttp](#) or [Volley](#)
- Pueden ser llamadas desde el main thread
- Mucho menos código



Volley

```
RequestQueue queue = Volley.newRequestQueue(this);
String url = "http://www.google.com";

StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
        new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        // Do something with response
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {}
});
queue.add(stringRequest);
```

OkHttp

```
OkHttpClient client = new OkHttpClient();
Request request = new Request.Builder()
    .url("http://publicobject.com/helloworld.txt").build();

client.newCall(request).enqueue(new Callback() {
    @Override
    public void onResponse(Call call, final Response response)

        throws IOException {
        try {
            String responseData = response.body().string();
            JSONObject json = new JSONObject(responseData);
            final String owner = json.getString("name");
        } catch (JSONException e) {}
    }
});
```

Parsear los resultados





Parsear los resultados

- Implementar método para recibir y manejar los resultados (`onPostExecute()`)
- La “response” es en general JSON or XML

Parsear Usando

- [JSONObject](#), [JSONArray](#)
- [XMLPullParser](#)—pares XML



JSON basico

```
{  
  "population":1,252,000,000,  
  "country":"India",  
  "cities":["New Delhi","Mumbai","Kolkata","Chennai"]  
}
```



JSONObject basics

```
JSONObject jsonObject = new JSONObject(response);  
String nameOfCountry = (String) jsonObject.get("country");  
long population = (Long) jsonObject.get("population");  
JSONArray listOfCities = (JSONArray) jsonObject.get("cities");  
Iterator<String> iterator = listOfCities.iterator();  
while (iterator.hasNext()) {  
    // do something  
}
```



Otro ejemplo JSON

```
{ "menu": {  
  "id": "file",  
  "value": "File",  
  "popup": {  
    "menuitem": [  
      { "value": "New", "onclick": "CreateNewDoc()" },  
      { "value": "Open", "onclick": "OpenDoc()" },  
      { "value": "Close", "onclick": "CloseDoc()" }  
    ]  
  }  
}
```

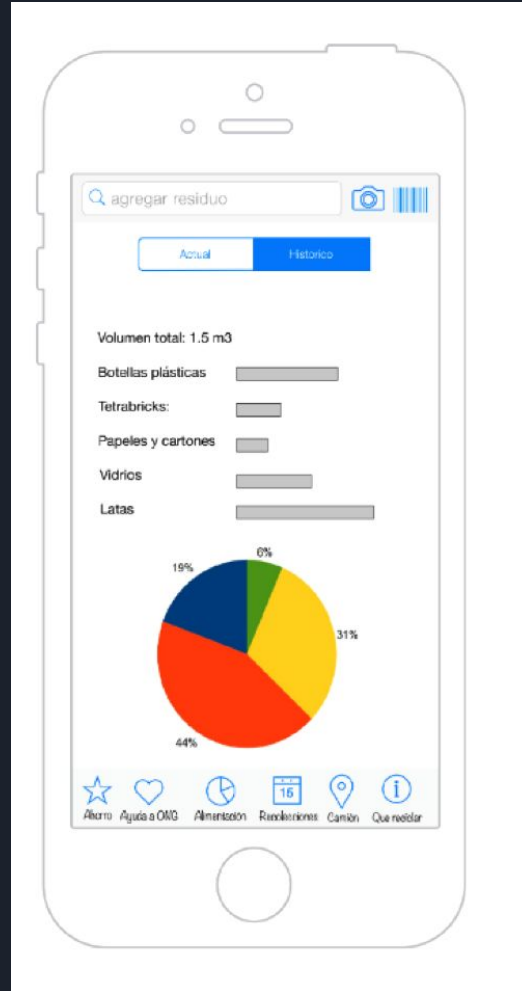



Otra ejemplo JSON

```
JSONObject data = new JSONObject(responseString);  
  
JSONArray menuItemArray =  
    data.getJSONArray("menuItem");  
  
JSONObject thirdItem =  
    menuItemArray.getJSONObject(2);  
  
String onClick = thirdItem.getString("onclick");
```

FINAL

- Registro de un usuario
- Almacenar reciclados
 - Botellas, tetrabricks, papeles, vidrios, latas.
- Submitear un reciclado
- Mostrar listas de reciclados
- Mostrar un total de todo lo reciclado.





Registrar un usuario

- POST con
- URL: `http://localhost:8080/api/users`
- Body:

```
{"firstName":"Mauri","lastName":"Arroqui","mail":"mauriarroqui@gmail.com","username":"marroqui2","address":{"department":"Tandil","number":874,"streetAddress":"Alberdi","city":"Tandil","state":"Buenos Aires","zipCode":"7000"}}
```



Agregar un reciclado

- POST
- URL: `http://localhost:8080/api/users/marroqui2/recycling/`
- Body:
`{"bottles":1,"tetrabriks":5,"glass":3,"paperboard":4,"cans":2,"date":"2018-11-28"}`

IMPORTANTE: marroqui2 es parametro!!!



Obtener la lista de reciclados

- GET
- URL <http://localhost:8080/api/users/marroqui2/recycling/>



Obtener el total de todo lo reciclado

- GET
- <http://localhost:8080/api/users/marroqui2/total/>



Base de datos en memoria

- <http://localhost:8080/h2-console>
 - Driver class: org.h2.Driver
 - JDBC URL: jdbc:h2:mem:testdb
 - User Name: sa
 - Password:

FIN!!!

