

Cliente API REST con LungoJS

Adolfo Sanz De Diego

Octubre 2013

1 Acerca de

1.1 El GUL

- El GUL es el **Grupo de Usuarios de Linux de la UC3M**.
- Grupo de personas con inquietudes en torno a la informática.
- Con la idea común de la utilización y promoción del **Software Libre**.
- Quedamos de vez en cuando y organizamos actividades sobre todo esto.
- El punto de unión es la **lista de correo** que está abierta a todo el mundo.

1.2 ¿Dónde encontrarnos?

- Twitter: <http://twitter.com/guluc3m>
- Lista: gul@gul.uc3m.es
- Ftp: <ftp://ftp.gul.uc3m.es>
- Web: <http://www.gul.uc3m.es>
- Podcast: <http://holamundo.gul.es/>
- Blog: <http://planeta.gul.uc3m.es/>
- Linkedin: <http://www.linkedin.com/groups?gid=3451836>

1.3 Adolfo Sanz De Diego

- **Antiguo programador web JEE**
- Hoy en día:
 - **Profesor de FP de informática:**
 - Hardware, Sistemas Operativos
 - Redes, Programación
 - **Formador Freelance:**
 - Java, Android
 - JavaScript, jQuery
 - JSF, Spring, Hibernate
 - Groovy & Grails
 - **Me gusta programar**

1.4 Hackalover



- **Para los amantes de los hackathones**
 - **Meetup:** <http://www.meetup.com/Hackathon-Lovers/>
 - **Twitter:** <http://twitter.com/HackathonLovers>
 - **Blog:** <http://hackathonlovers.tumblr.com/>
 - **LinkedIn:**
<http://www.linkedin.com/groups/Hackathon-Lovers-6510465>
 - **YouTube:**
<http://www.youtube.com/channel/UCRwSe7jK-y62BMvliNBV1qw>

1.5 Tweets Sentiment



- Es un **analizador de tweets** que extrae información semántica para conocer si el sentimiento general de los tweets de un determinado tema es positivo o negativo.
 - **Web:** <http://tweetssentiment.com/>
 - **Twitter:** <http://twitter.com/TweetsSentiment>

1.6 ¿Dónde encontrarme?

- Mi nick: **asanzdiego**
 - AboutMe: <http://about.me/asanzdiego>
 - GitHub: <http://github.com/asanzdiego>
 - Twitter: <http://twitter.com/asanzdiego>
 - Blog: <http://asanzdiego.blogspot.com.es>
 - LinkedIn: <http://www.linkedin.com/in/asanzdiego>
 - Google+:
<http://plus.google.com/+AdolfoSanzDeDiego>

1.7 Créditos

- Estas **transparencias** están hechas con:
 - <https://github.com/asanzdiego/markdownslides>

1.8 Licencia

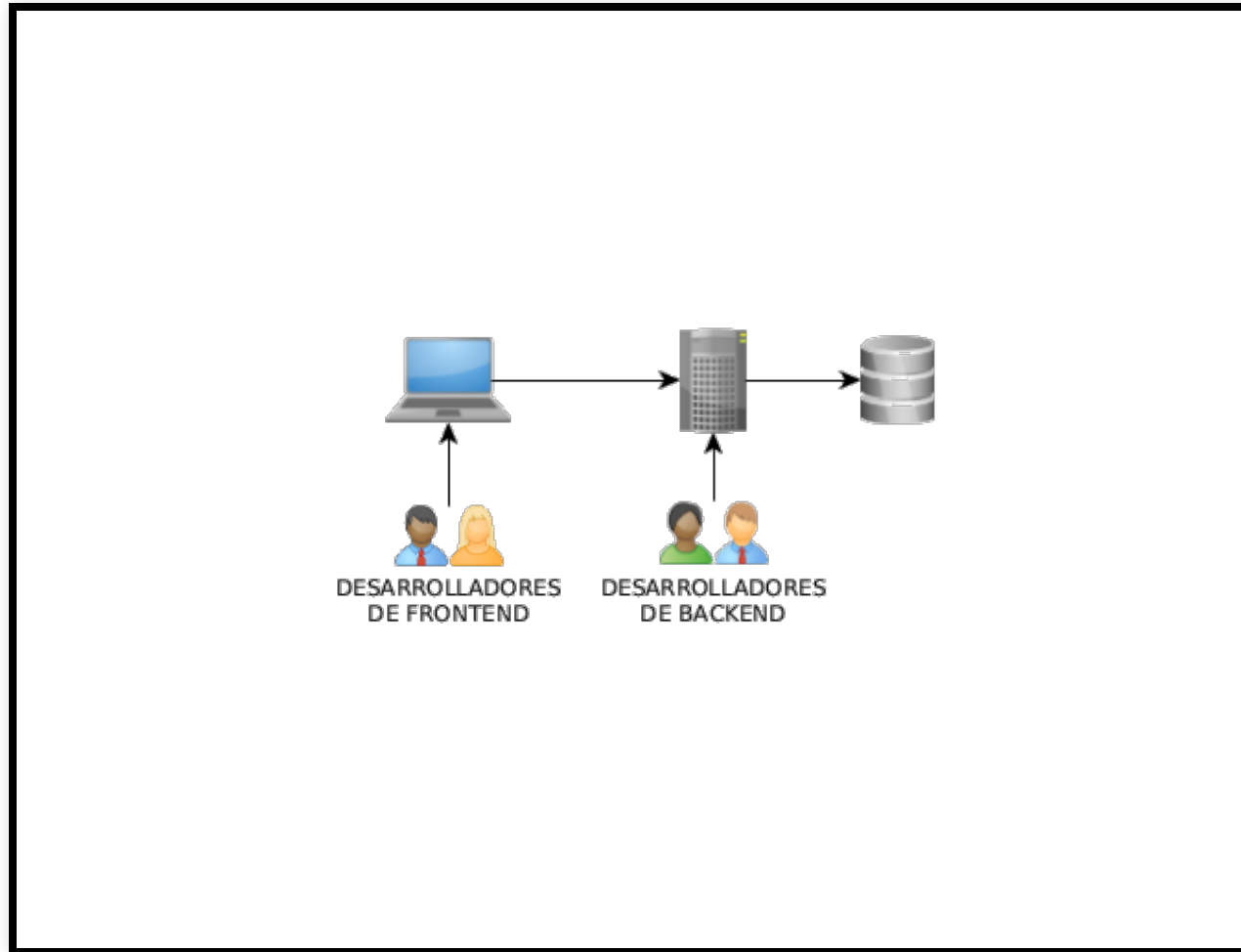
- Estas **transparencias** están bajo una licencia:
 - Creative Commons Reconocimiento-CompartirIgual 3.0
- El **código** de los programas están bajo una licencia:
 - GPL 3.0

1.9 Fuentes

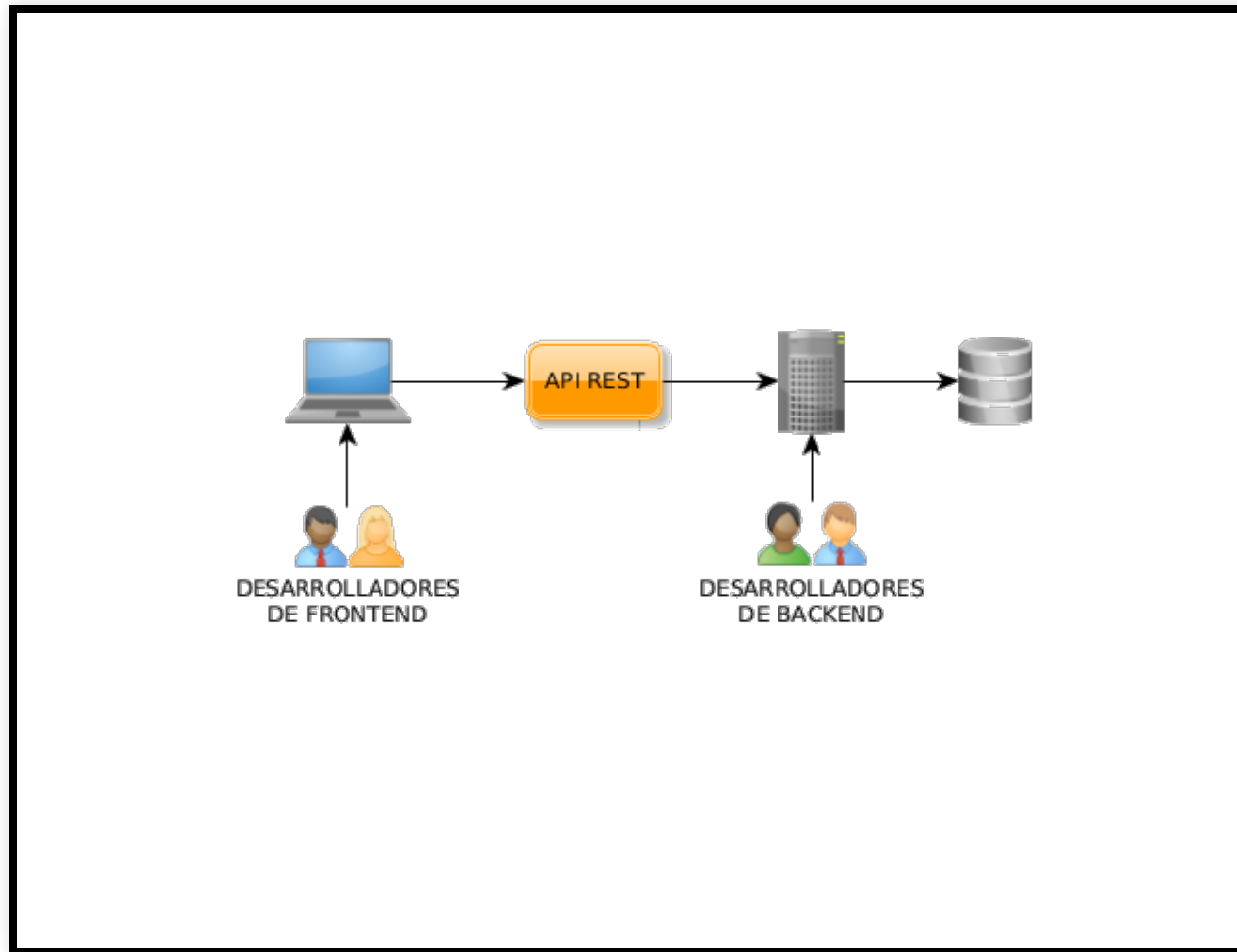
- Transparencias
 - SlideShare
 - Deck Slides
 - Reveal Slides
 - Plain HTML
- Código
 - <https://github.com/asanzdiego/curso-api-restful-lungojs-client-2013/tree/master/src>

2 APIs ¿Para
qué?

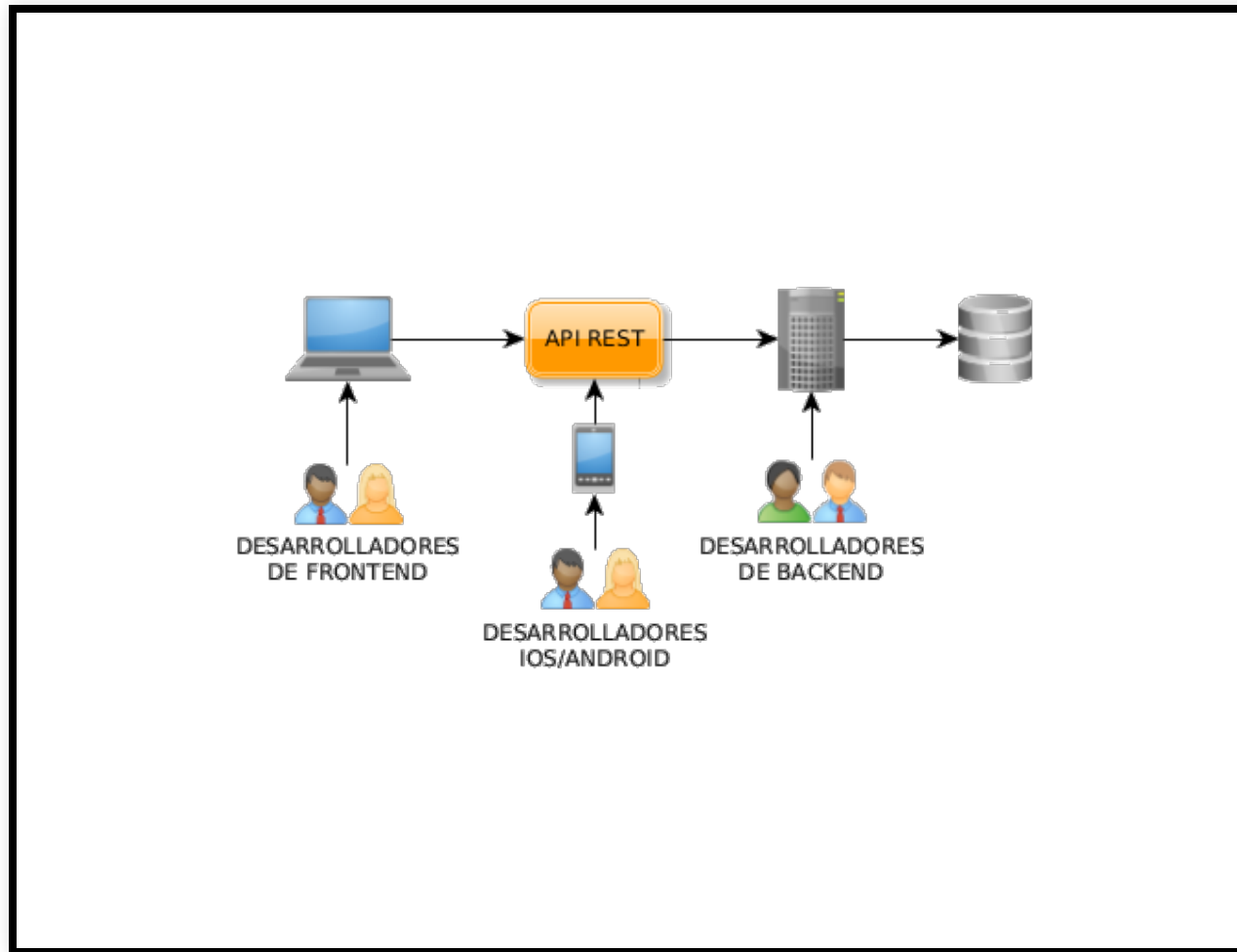
2.1 Aplicación estándar



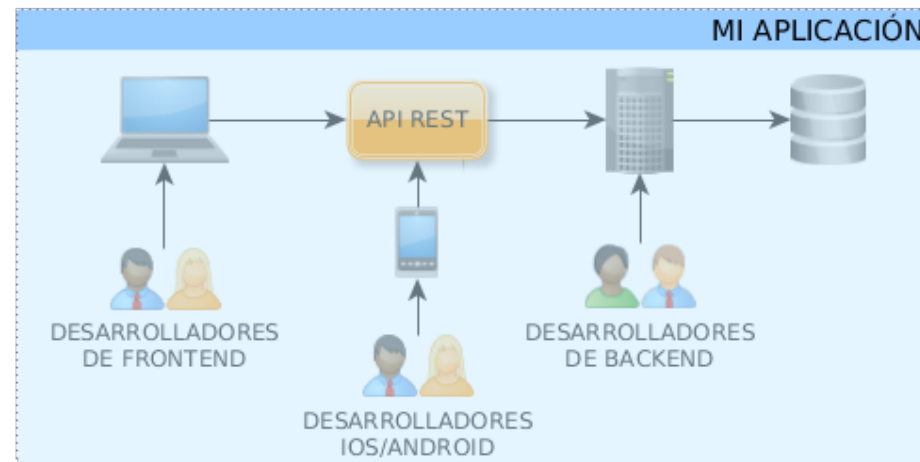
2.2 Introducimos API



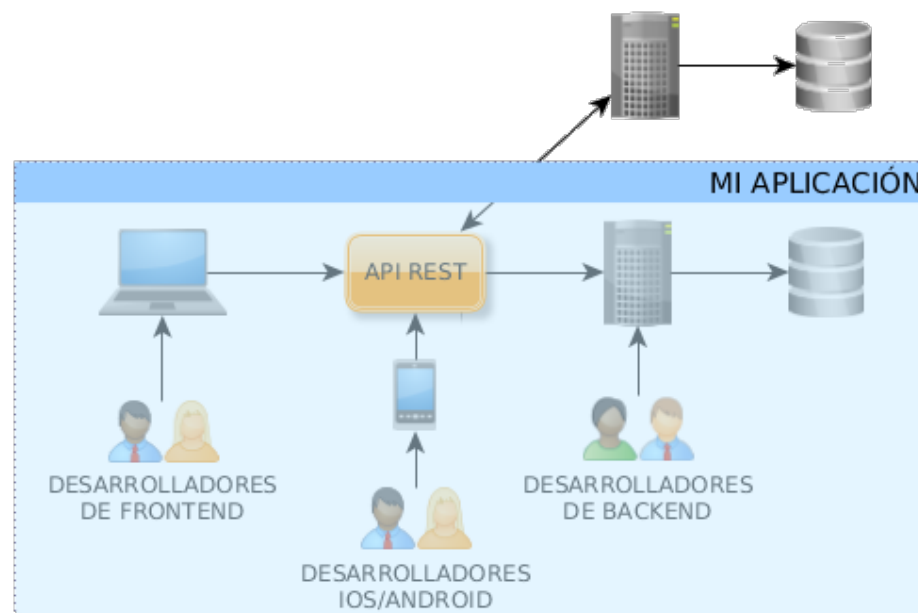
2.3 Separación Roles



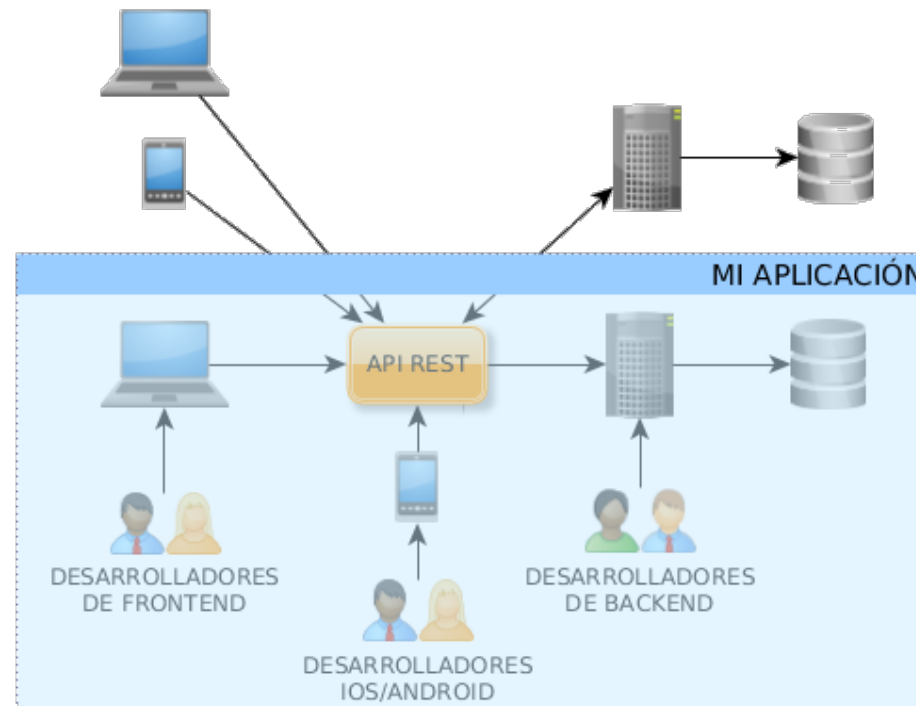
2.4 ¿Y ahora qué?



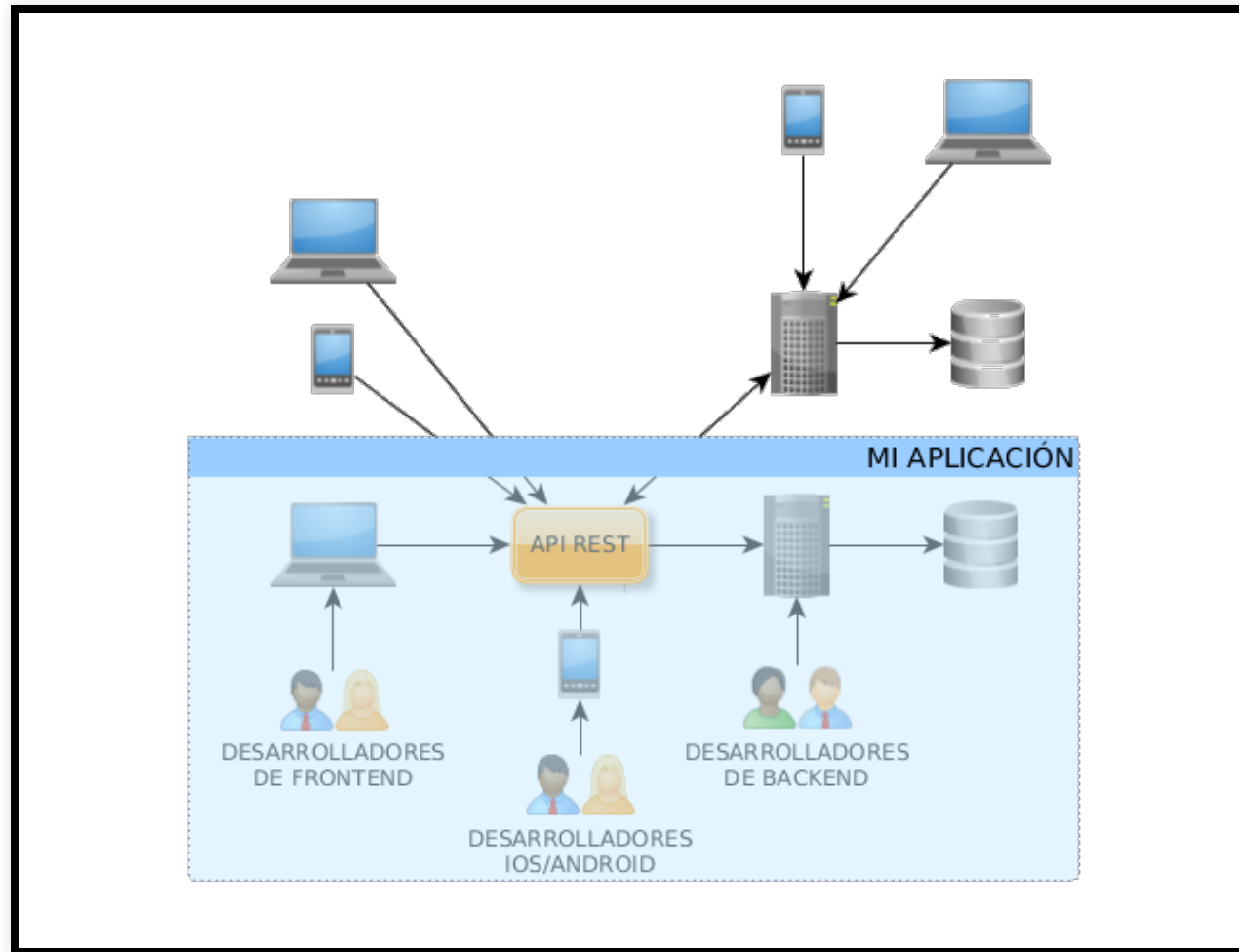
2.5 Servicios externos



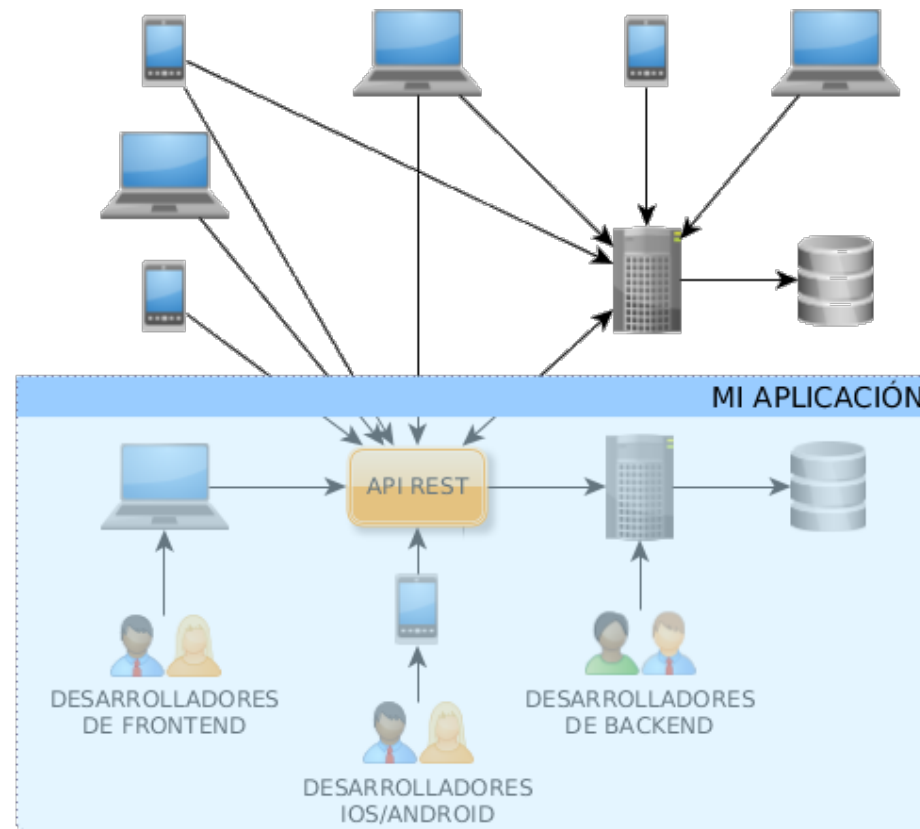
2.6 Apps clientes



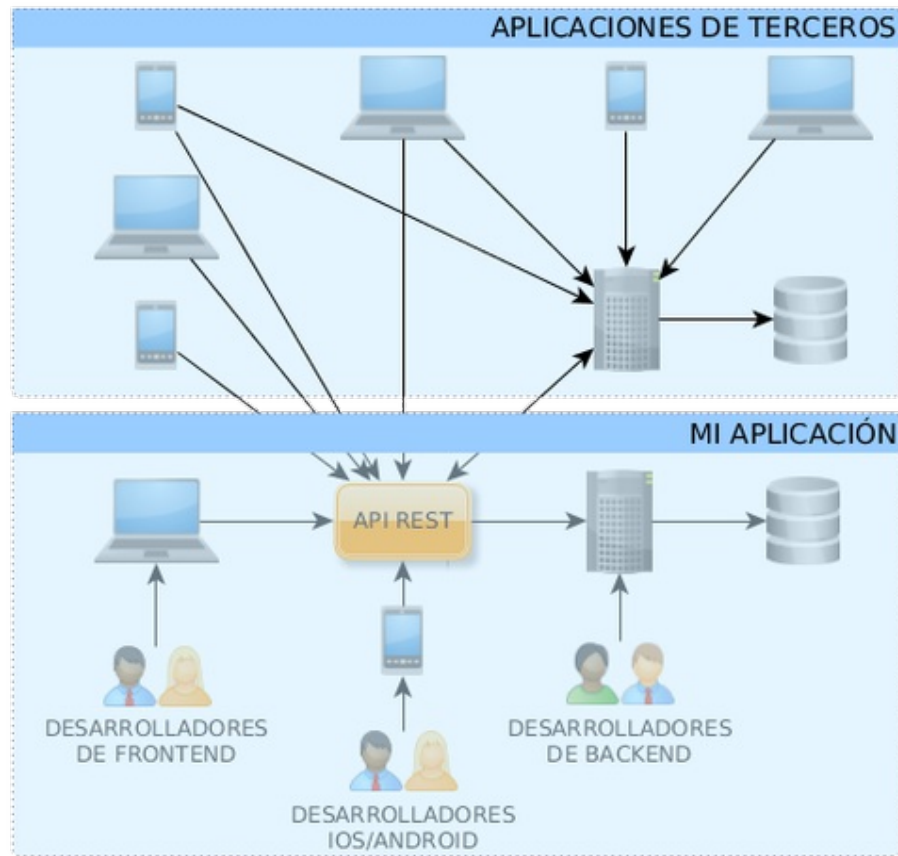
2.7 Apps de servicios



2.8 Apps mixtas



2.9 Plataforma



2.10 ¿Quien expone APIs?



2.11 ¿Quien expone APIs?



2.12 ¿Quien expone APIs?



2.13 Exponlas tú



2.14 Exponlas tú



2.15 Exponlas tú



3 APIs RESTful

3.1 ¿Qué es REST?

- REST (**Representational State Transfer**) es una técnica de arquitectura de software para sistemas hipermedia distribuidos como la World Wide Web.
- En REST una **URL** (Uniform Resource Locator) representa un **recurso**.
- Se puede acceder al recurso o modificarlo mediante los **métodos del protocolo HTTP**:

GET, POST, PUT, DELETE

3.2 Ejemplo API

- **http://myhost.com/talk**
 - GET > Devuelve todas las charlas.
 - POST > Crear una nueva charla.
- **http://myhost.com/talk/123**
 - GET > Devuelve la charla con id=123
 - PUT > Actualiza la charla con id=123
 - DELETE > Borra la charla con id=123

3.3 Manejo de errores

- **Se pueden utilizar los errores del protocolo HTTP:**
 - 200 Successful
 - 201 Created
 - 202 Accepted
 - 301 Moved Permanently
 - 400 Bad Request
 - 401 Unauthorised
 - 402 Payment Required
 - 403 Forbidden
 - 404 Not Found
 - 405 Method Not Allowed
 - 500 Internal Server Error
 - 501 Not Implemented

3.4 ¿Por qué REST?

- Es **más sencillo** (tanto la API como la implementación).
- Es **más rápido** (peticiones más ligeras que se pueden cachear).
- Es **multiformato** (HTML, XML, JSON, etc.).
- Se complementa muy bien con **AJAX**.

3.5 REST vs RESTful

- REST se refiere a un tipo de arquitectura de software
 - Se utiliza como **nombre**
 - Se utiliza como por ejemplo: success = éxito.
- Si un servicio web es REST**ful** indica que implementa dicha arquitectura.
 - Se utiliza como **adjetivo**
 - Se utiliza como por ejemplo: success**ful** = éxito**so**).

3.6 REST vs RESTful

- A veces el **ful** se confunde con **full** = completo.
 - Y se refiere a los servicios web REST**full**

Aquellos que implementan una API con todos los métodos del protocolo HTTP.

- Y se refiere a los servicios web REST (**sin el full**)

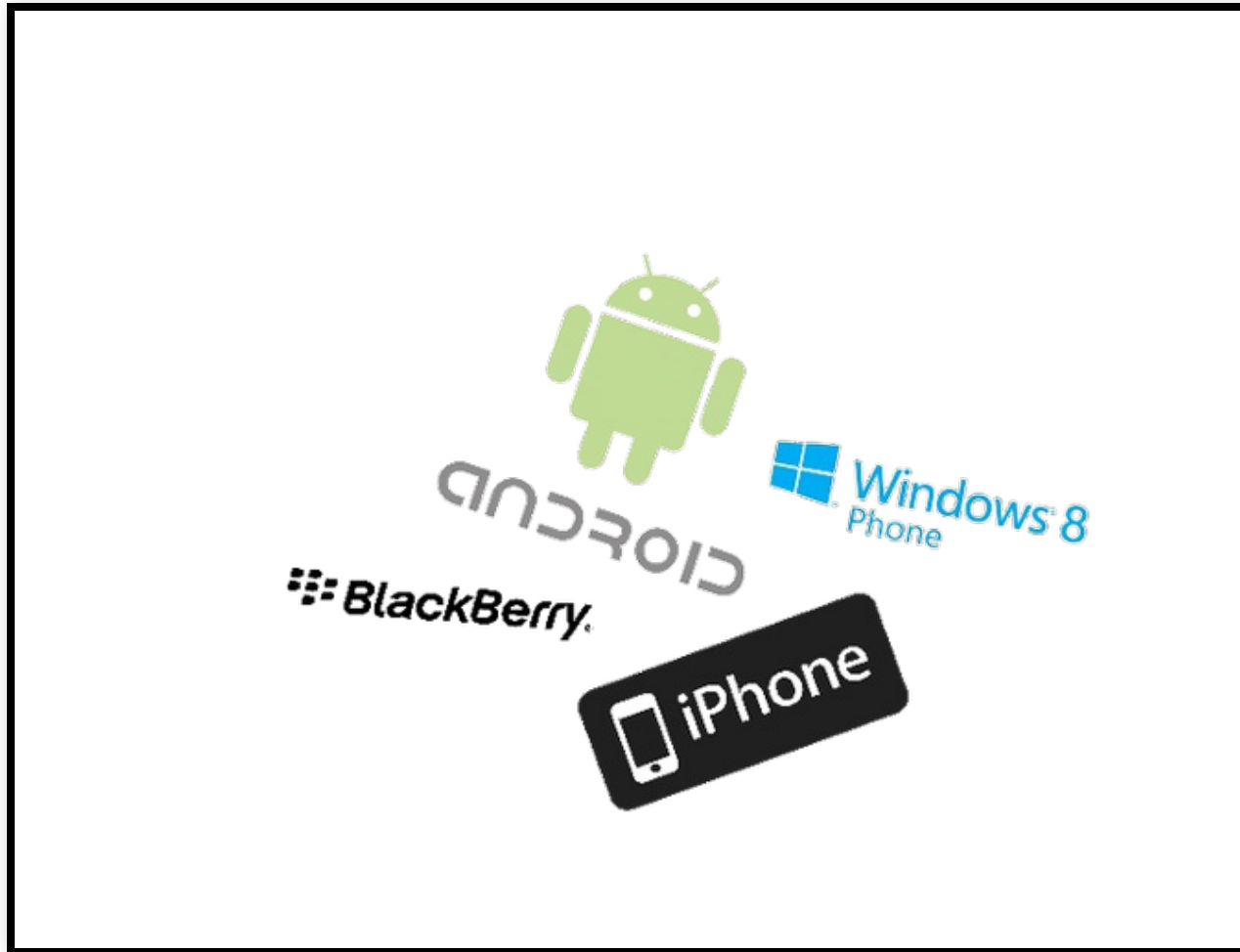
Aquellos que NO implementan una API con todos los métodos del protocolo HTTP.

4 Clientes
multidispositi
vo

4.1 Distintos dispositivos



4.2 Distintos OS



4.3 Distintos OS



4.4 Distintos navegadores



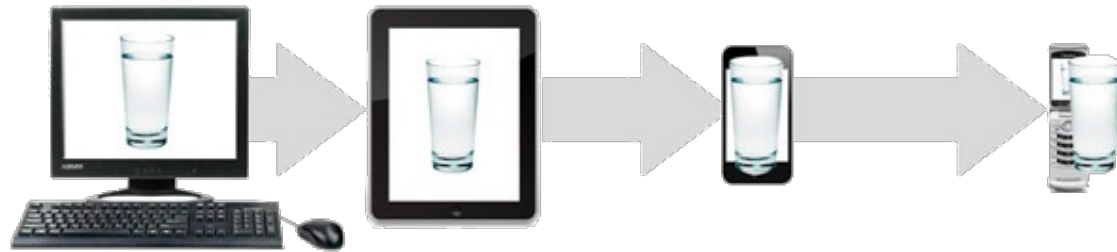
4.5 iiiSocorro!!!



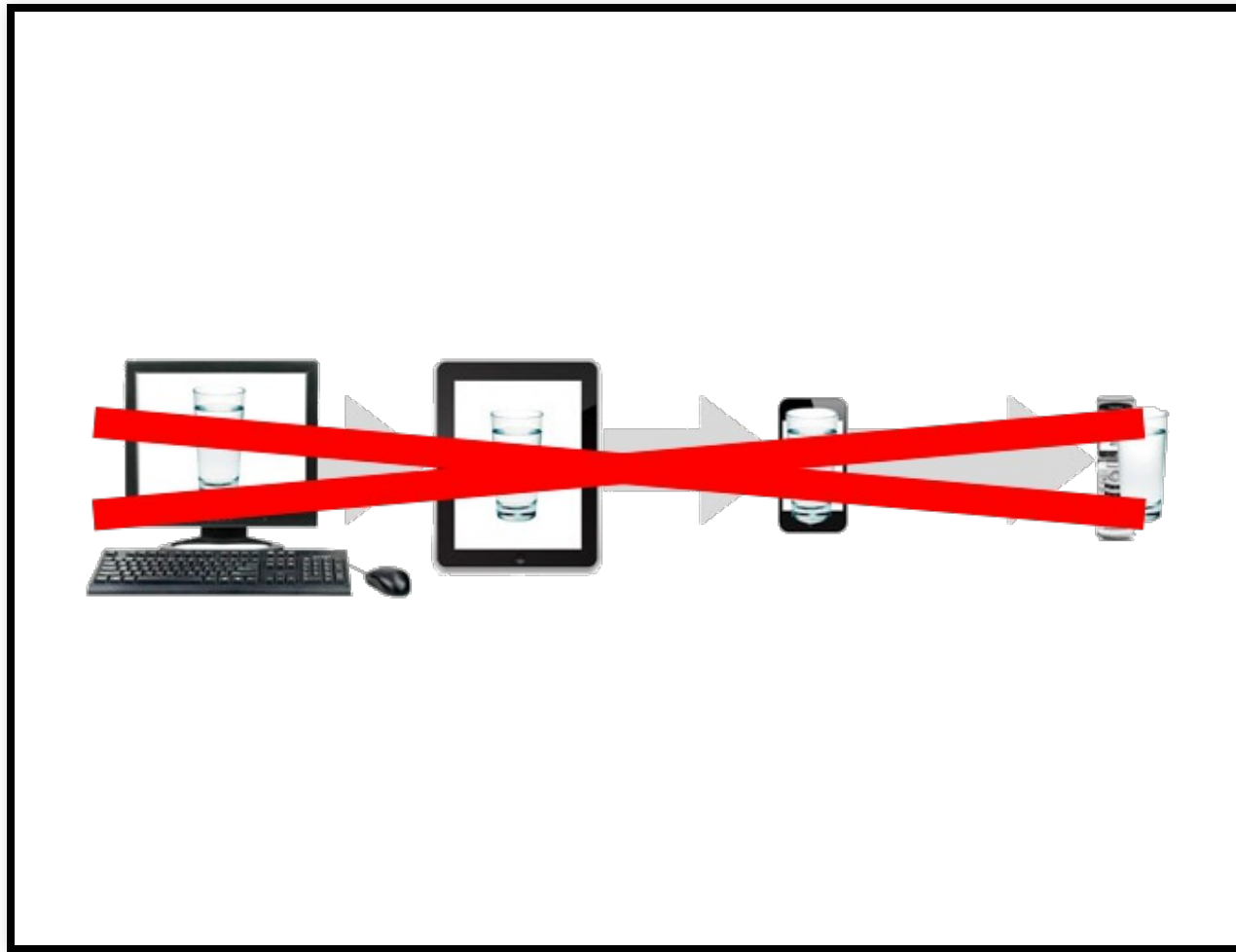
4.6 AI rescue



4.7 Graceful Degradation



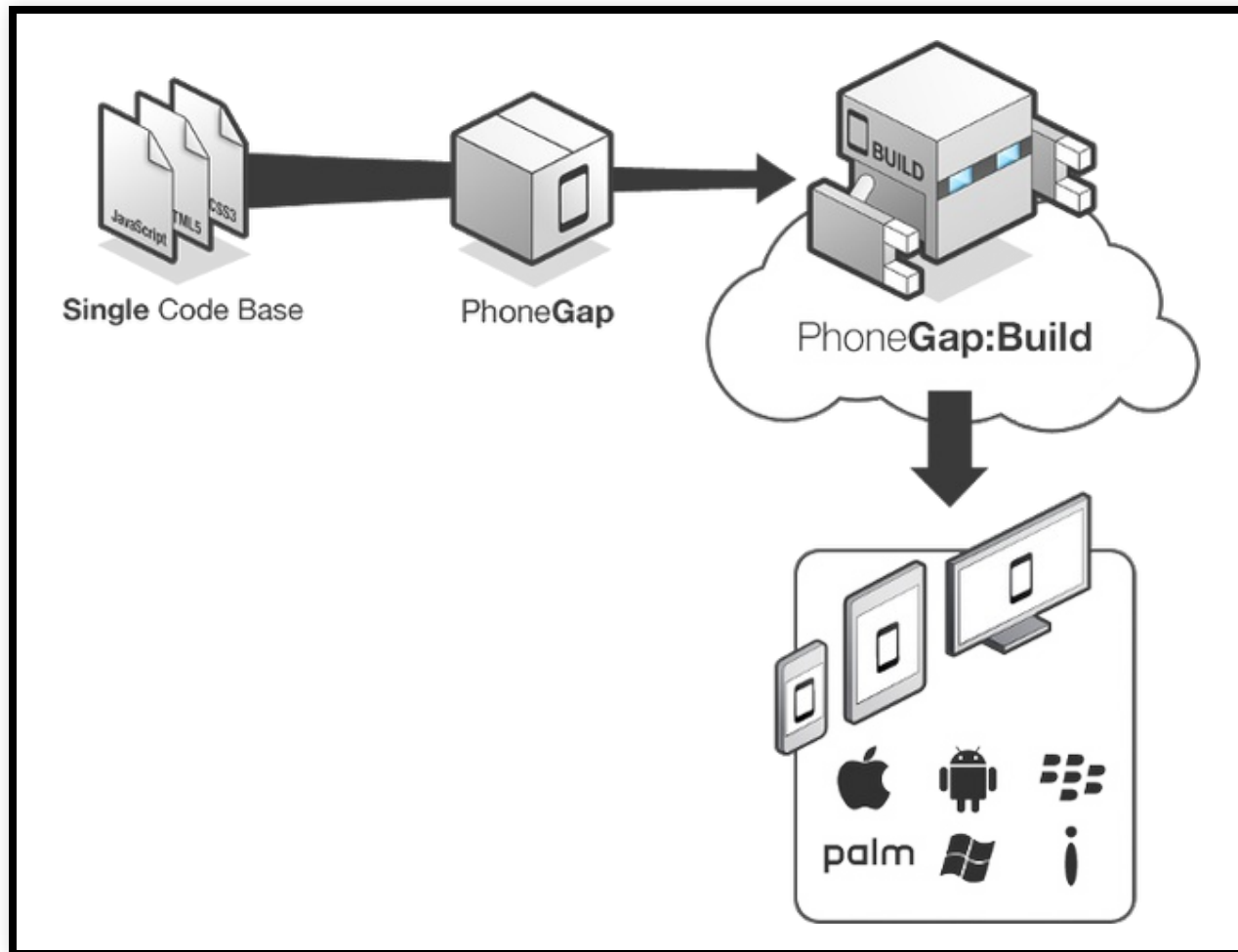
4.8 Graceful Degradation



4.9 Progressive Enhancement



4.10 Phonegap



5 Lungo.js

5.1 Competidores

- jQTouch: <http://jqtouch.com/>
- Sencha Touch:
<http://www.sencha.com/products/touch/>
- jQueryMobile: <http://jquerymobile.com/>
- hammer.js: <http://eightmedia.github.io/hammer.js/>

5.2 ¿Por qué Lungo?



**¿Por qué
usar lungo?**

5.3 ¿Licencia GPL?



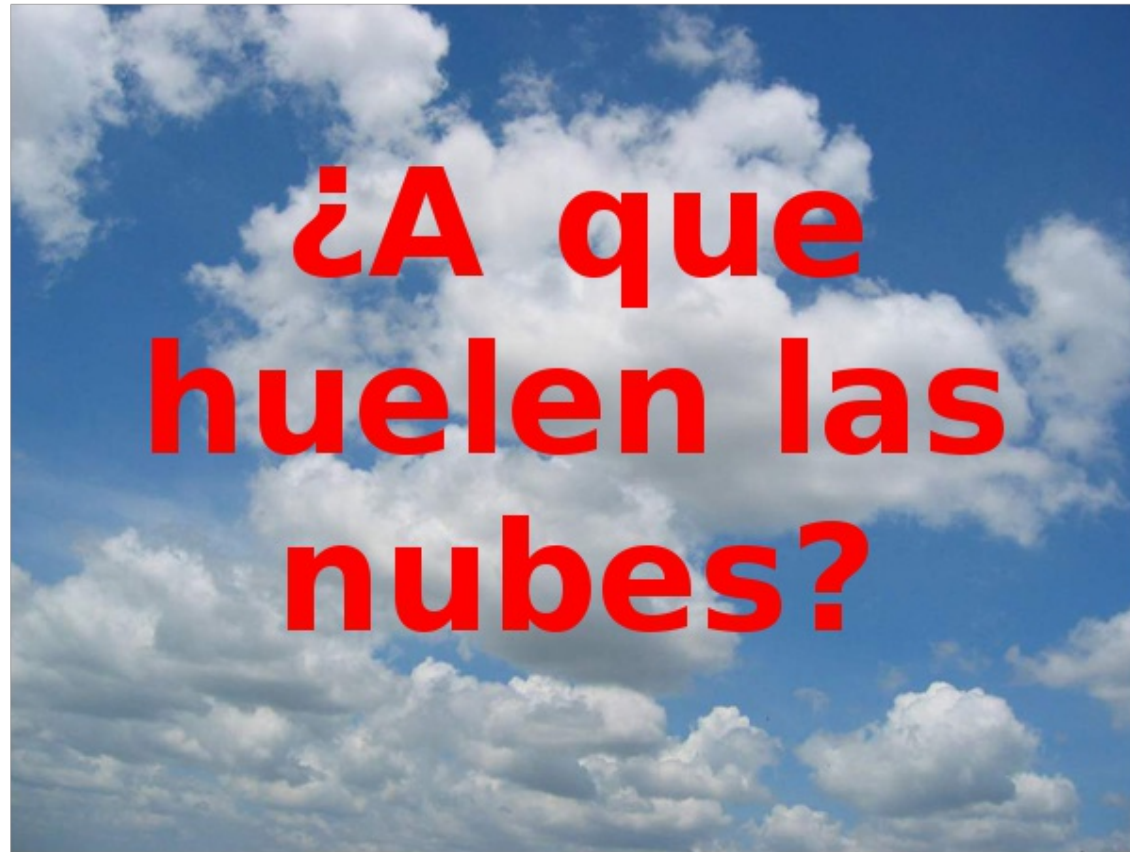
5.4 ¿Español?



5.5 Pues no



5.6 Me gusta

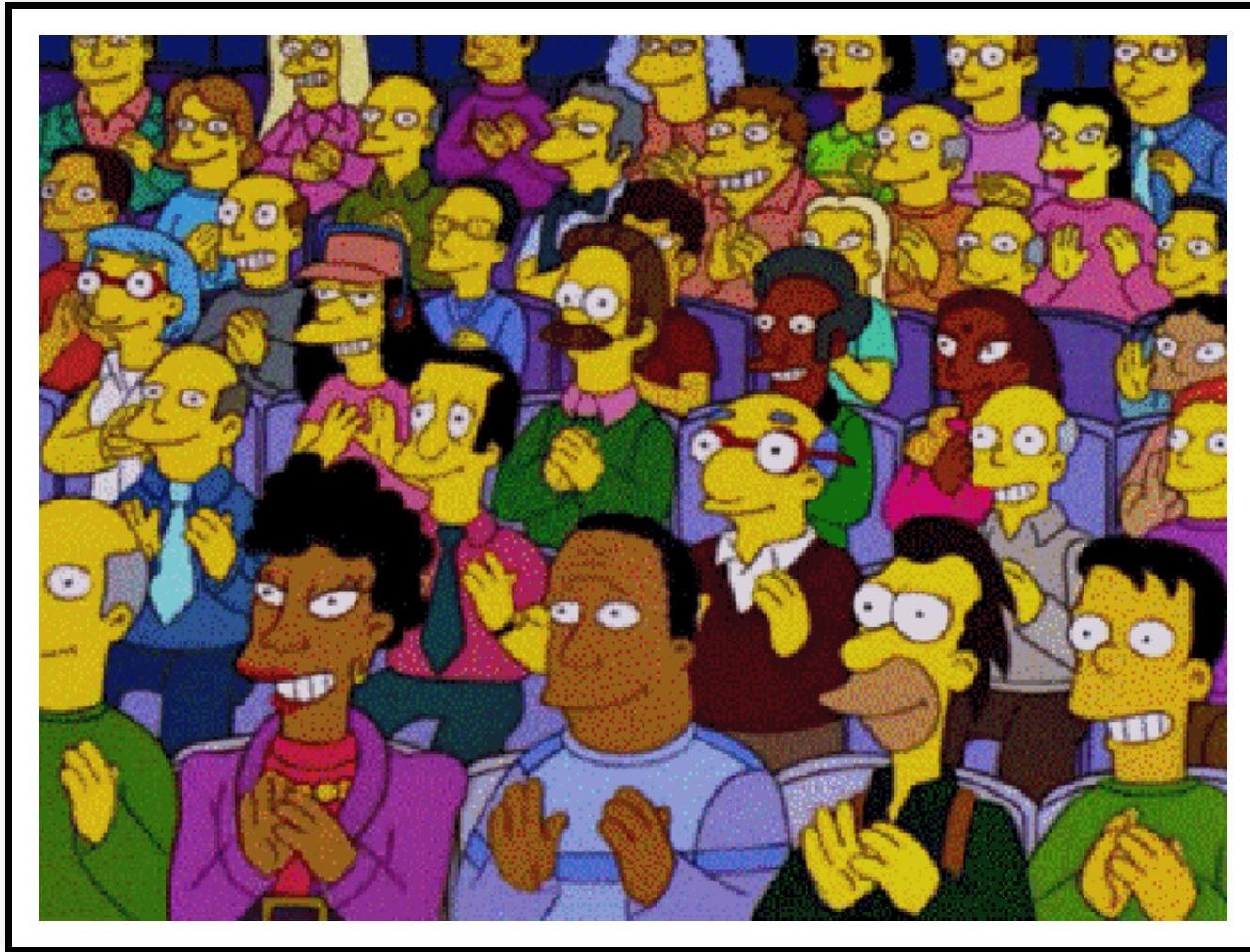


5.7 Aburrido



6 Código

6.1 Aplausos



6.2 Directorios

- Puedes usar la estructura de directorios que quieras.
- Yo he usado esta:
 - **html**: los HTML con una carpeta para cada entidad del domino
 - **js**: los JS con una carpeta para cada entidad del domino
 - **lib**: carpeta con las dependencias a otros proyectos
 - Lungo depende de Quo que es una librería de los mismos autores de 'tipo' jQuery enfocada a móviles

6.3 js/util.js

```
// Search
$$('document').ready(function(){
    Lungo.dom('input[type=search]').on('keyup', ...);
});

// Server URL
var util_server_url = "http://localhost:3000";

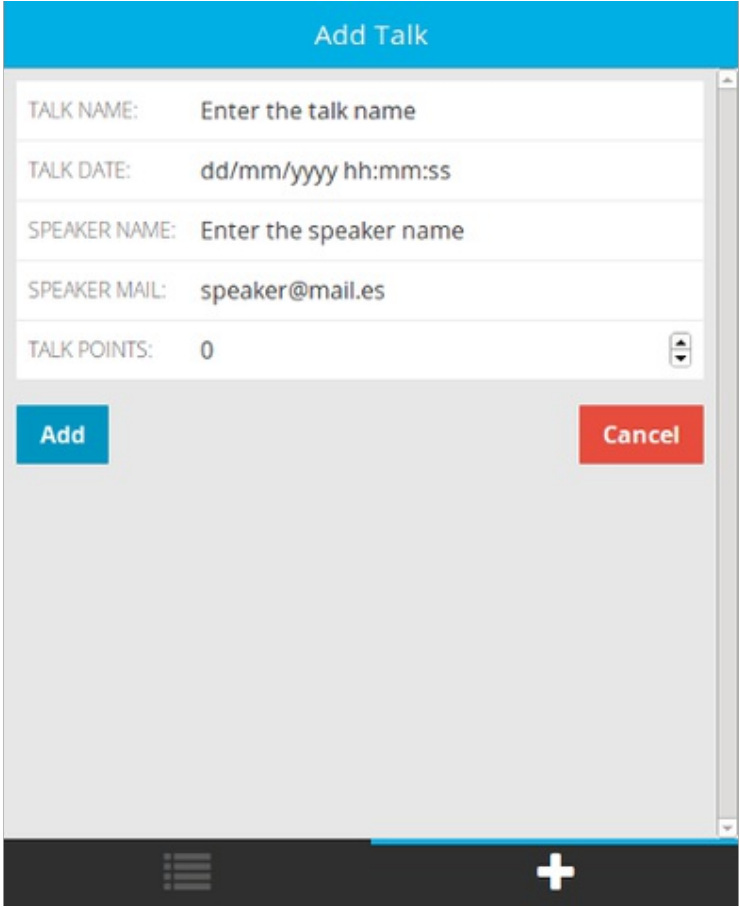
// Error Notification
var util_errorNotification = function(message, error) {
    Lungo.Notification.error(message, "", "warning-sign", 2);
};

// Métodos REST
var util_ajaxGet = function(url, data, callback) {
    $.get(util_server_url+url, data, ..., 'json');
};
var util_ajaxPost = function(url, data, callback) {...};
var util_ajaxPut = function(url, data, callback) {...};
var util_ajaxDelete = function(url, data, callback) {...};
```

6.4 html/talk/talk-add.html

```
<body class="app">
<section id="main" data-transition="">
  <header data-title="Add Talk" class="extended"></header>
  <footer>
    <nav>
      <a href="talk-list.html" data-icon="list"></a>
      <a href="talk-add.html" data-icon="plus" class="active"></a>
    </nav>
  </footer>
  <article id="main-article" class="active list">
    <div class="form">
      <fieldset>
        <label>NAME:</label><input type="text" id="talkName" />
      </fieldset>...
    </div>
    <div>
      <a href="#addTalk" id="addTalk" class="button">Add</a>
      <a href="talk-list.html" class="button cancel">Cancel</a>
    </div>
  </article>
</section>
</body>
```

6.5 html/talk/talk-add.html



The screenshot shows a mobile application interface for adding a talk. The form is titled "Add Talk" and contains several input fields. The "TALK NAME" field has a placeholder "Enter the talk name". The "TALK DATE" field has a placeholder "dd/mm/yyyy hh:mm:ss". The "SPEAKER NAME" field has a placeholder "Enter the speaker name". The "SPEAKER MAIL" field has a placeholder "speaker@mail.es". The "TALK POINTS" field has a value of "0" and a vertical scroll indicator. Below the form are two buttons: "Add" (blue) and "Cancel" (red). The bottom of the screen shows a dark navigation bar with a white plus sign.

Add Talk

TALK NAME: Enter the talk name

TALK DATE: dd/mm/yyyy hh:mm:ss

SPEAKER NAME: Enter the speaker name

SPEAKER MAIL: speaker@mail.es

TALK POINTS: 0

Add **Cancel**

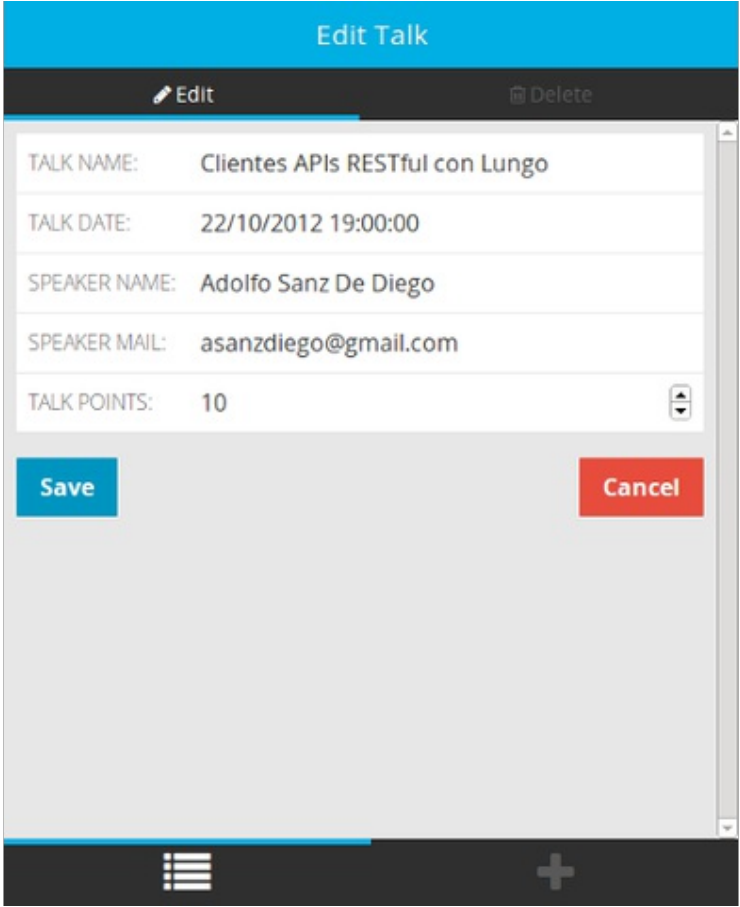
6.6 js/talk/talk-add.js

```
$('#addTalk').tap(function(){  
  
  var data = {  
    talkName:    $('#talkName').val(),  
    talkDate:    util_stringToDate($('#talkDate').val()),  
    talkSpeaker: $('#talkSpeaker').val(),  
    talkSpeakerMail: $('#talkSpeakerMail').val(),  
    talkPoints:  $('#talkPoints').val()  
  };  
  
  // send data to server  
  util_ajaxPost('/talk', data, function(json) {  
    if(!json || json.error) {  
      util_errorNotification('ERROR adding talk', json.error);  
    } else {  
      util_successNotification('Talk saved', function() {  
        window.location.replace('talk-list.html');  
      });  
    }  
  });  
});
```

6.7 html/talk/talk-edit.html

```
<body class="app">
<section id="main" data-transition="">
  <header data-title="Edit Talk" class="extended"></header>
  <nav data-control="groupbar">
    <a href="#editTalkArticle" ...>Edit</a>
    <a href="#deleteTalkArticle" ...>Delete</a>
  </nav>
  <footer>
    ...
  </footer>
  <article id="editTalkArticle" class="list indented scroll">
    ...
  </article>
  <article id="deleteTalkArticle" class="list indented scroll">
    ...
  </article>
</section>
</body>
```

6.8 html/talk/talk-edit.html

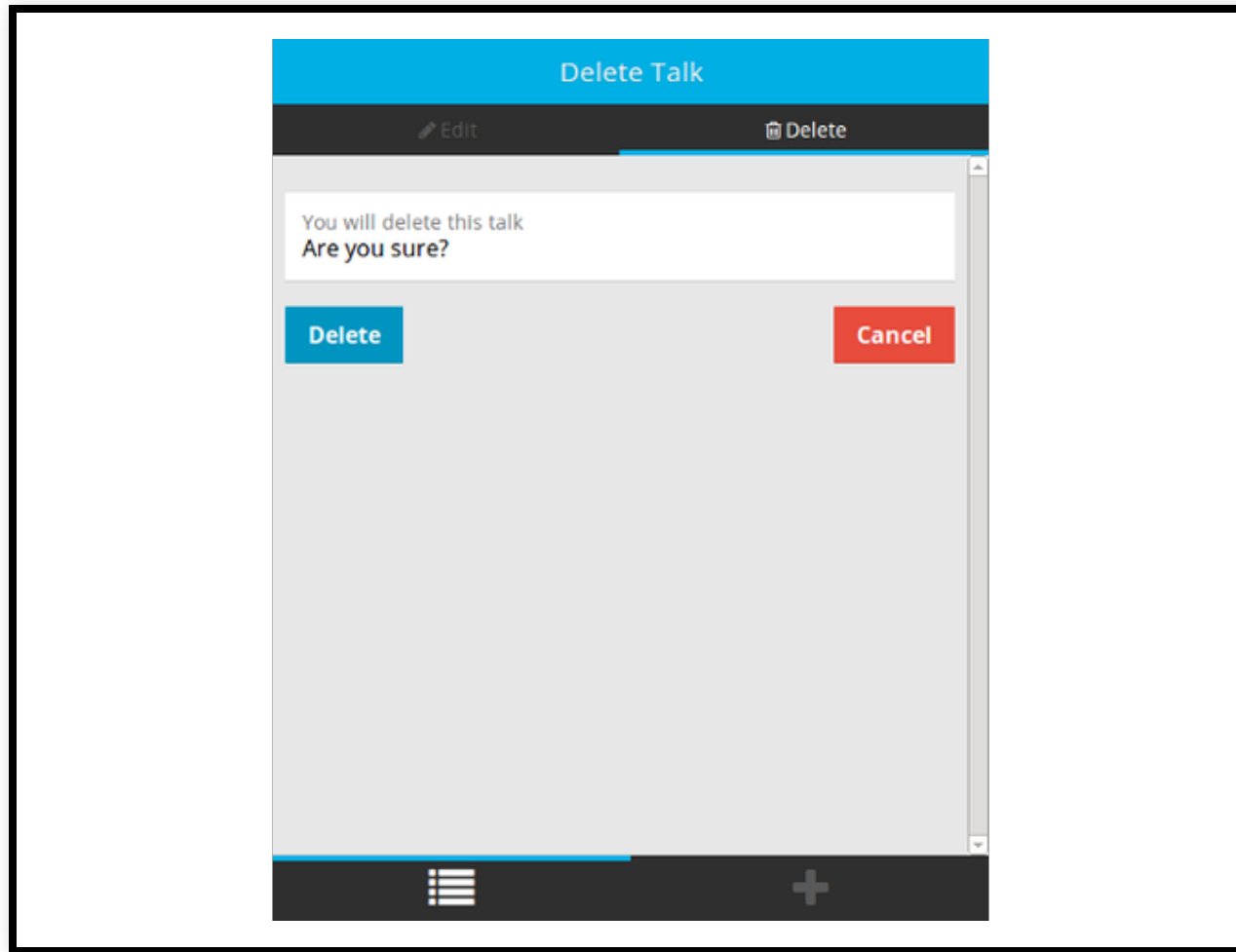


The screenshot displays a mobile application interface for editing a talk. The title bar is blue and labeled "Edit Talk". Below it, a dark grey navigation bar contains two options: "Edit" (with a pencil icon) and "Delete" (with a trash can icon). The main content area is a light grey form with the following fields:

TALK NAME:	Clientes APIs RESTful con Lungo
TALK DATE:	22/10/2012 19:00:00
SPEAKER NAME:	Adolfo Sanz De Diego
SPEAKER MAIL:	asanzdiego@gmail.com
TALK POINTS:	10

At the bottom of the form, there are two buttons: a blue "Save" button on the left and a red "Cancel" button on the right. The bottom of the screen features a dark grey navigation bar with a hamburger menu icon on the left and a plus sign icon on the right.

6.9 html/talk/talk-edit.html



6.10 js/talk/talk-get.js

```
$$('document').ready(function(){  
  
    var talkId = util_urlParams["talkId"];  
  
    // get data from server  
    util_ajaxGet('/talk/'+talkId, {}, function(json) {  
        if(!json || json.error) {  
            util_errorNotification('ERROR retrieving talk', json.error);  
        } else {  
            var talk = json;  
            console.log('Talk retrieved');  
            drawTalk(talk);  
        }  
    });  
  
    // draw data  
    var drawTalk = function(talk) {  
        $$("#talkName").val(talk.talkName);  
        ...  
    };  
});
```


6.11 js/talk/talk-update.js

```
$('#saveTalk').tap(function(){  
  
    var talkId = util_urlParams["talkId"];  
  
    var data = {  
        talkName:    $('#talkName').val(),  
        ...  
    };  
  
    // send data to server  
    util_ajaxPut('/talk/'+talkId, data, function(json) {  
        if(!json || json.error) {  
            util_errorNotification('ERROR saving talk', json.error);  
        } else {  
            util_successNotification('Talk saved', function() {  
                window.location.replace('talk-list.html');  
            });  
        }  
    });  
});  
});
```

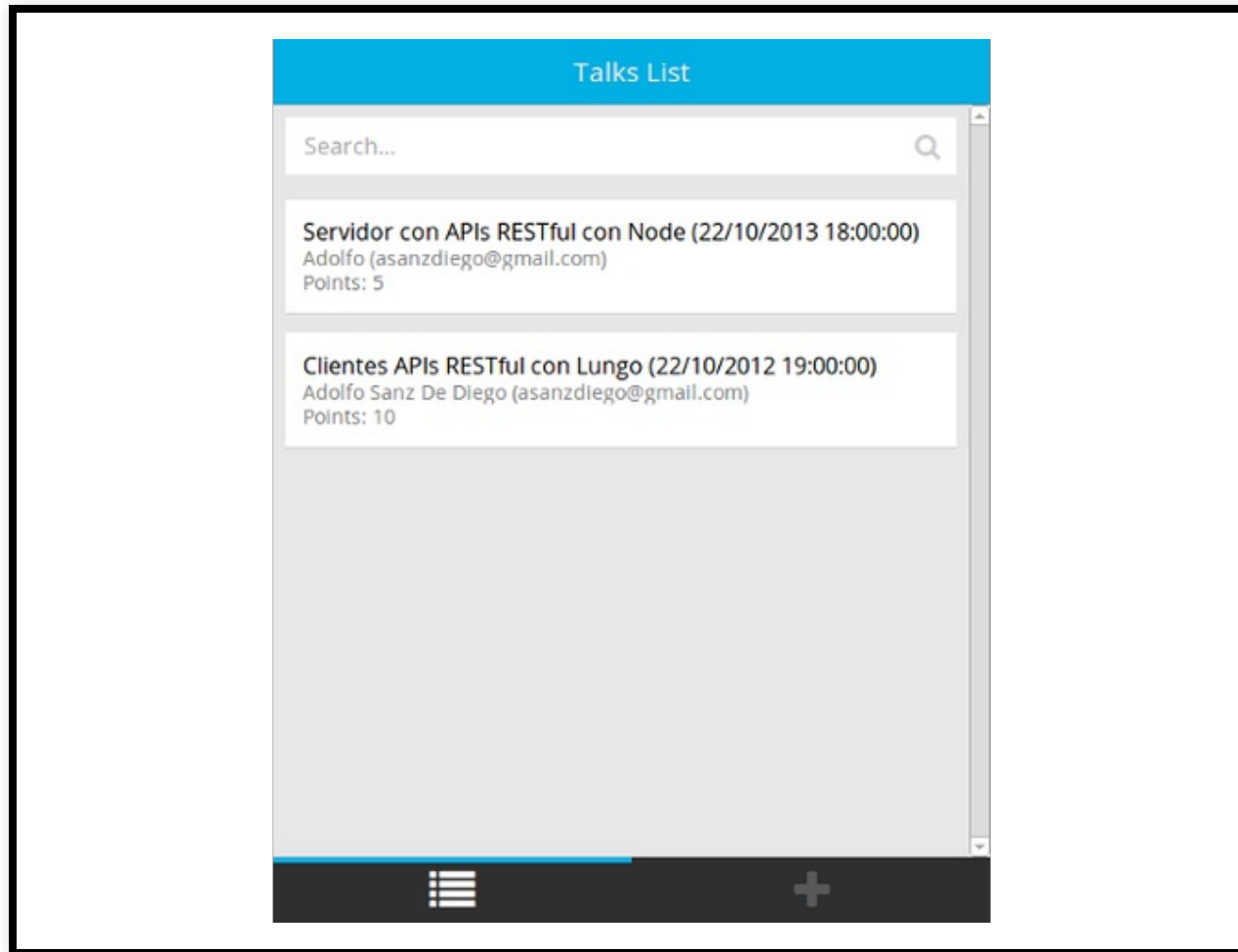
6.12 js/talk/talk-delete.js

```
$('#deleteTalk').tap(function(){  
  
    var talkId = util_urlParams["talkId"];  
  
    // send data to server  
    util_ajaxDelete('/talk/'+talkId, {}, function(json) {  
        if(!json || json.error) {  
            util_errorNotification('ERROR deleting talk', json.error);  
        } else {  
            util_successNotification('Talk deleted', function() {  
                window.location.replace('talk-list.html');  
            });  
        }  
    });  
});
```

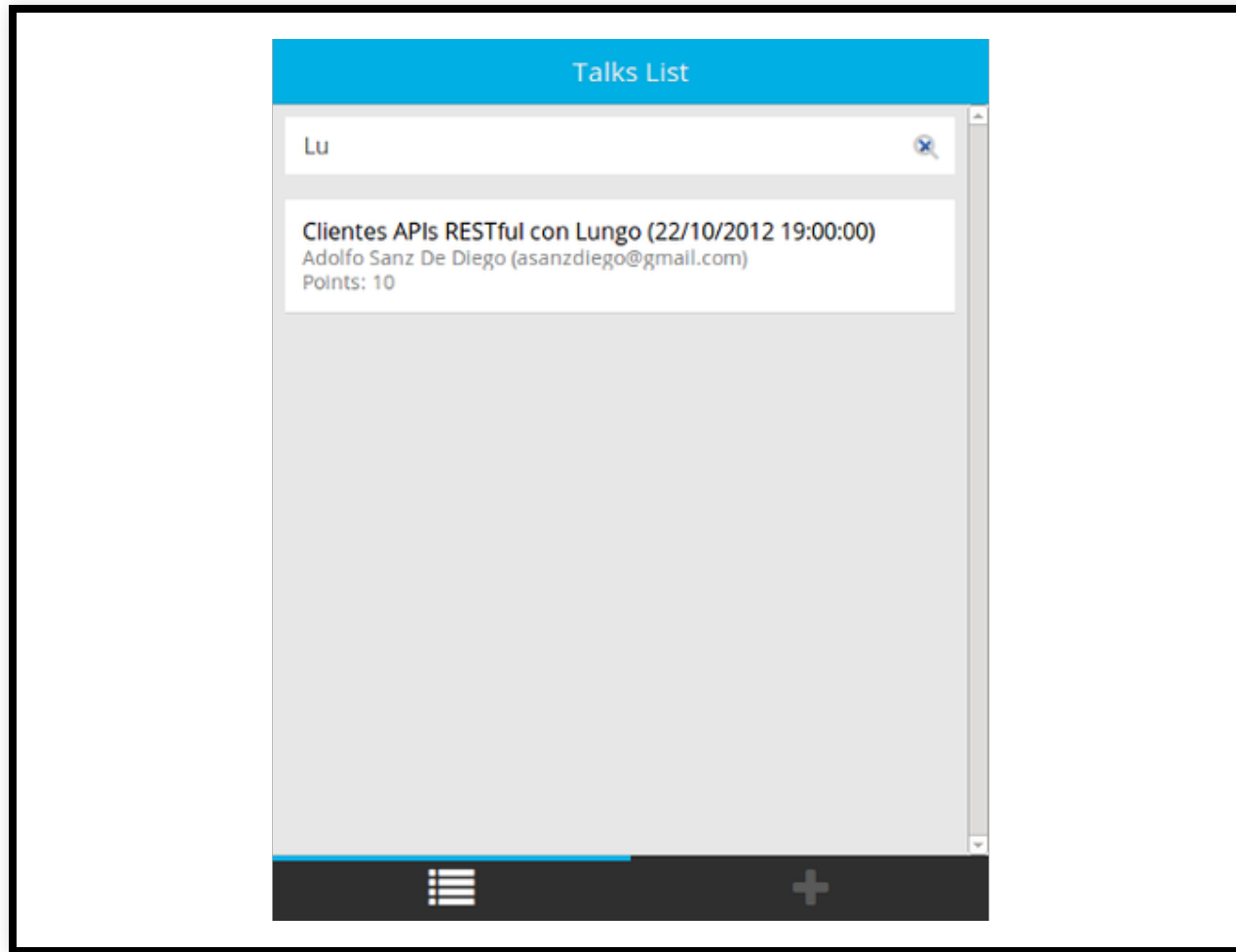
6.13 html/talk/talk-list.html

```
<body class="app">
<section id="main" data-transition="">
  <header data-title="Talks List" class="extended"></header>
  <footer>
    <nav>
      <a href="talk-list.html" data-icon="list" class="active"></a>
      <a href="talk-add.html" data-icon="plus"></a>
    </nav>
  </footer>
  <article id="main-article" class="active list indented scroll">
    <div class="form">
      <fieldset data-icon="search">
        <input type="search" placeholder="Search...">
      </fieldset>
    </div>
    <ul id="talks"></ul>
  </article>
</section>
</body>
```

6.14 html/talk/talk-list.html



6.15 html/talk/talk-list.html



6.16 js/talk/talk-list.js

```
$$('document').ready(function(){  
  
  // get data from server  
  util_ajaxGet('/talk', {}, function(json) {  
    if(!json || json.error) {  
      util_errorNotification('ERROR retrieving talks', json.error);  
    } else {  
      drawTalks(json);  
    }  
  });  
  
  // draw data  
  var drawTalks = function(talks) {  
    for (var i = 0; i < talks.length; i++) {  
      var talk = talks[i];  
      $$("#talks").append(  
        '<li data-action="search" class="selectable">'+  
        '<a href="talk-edit.html?talkId='+talk._id+'">'+  
        '<strong>'+talk.talkName+'</strong></a></li>');  
    }  
  };  
});
```

7 Demo

8 ¿Alguna
pregunta?