

CLIENTE API REST CON LUNGOJS

ADOLFO SANZ DE DIEGO

OCTUBRE 2013

1 ACERCA DE

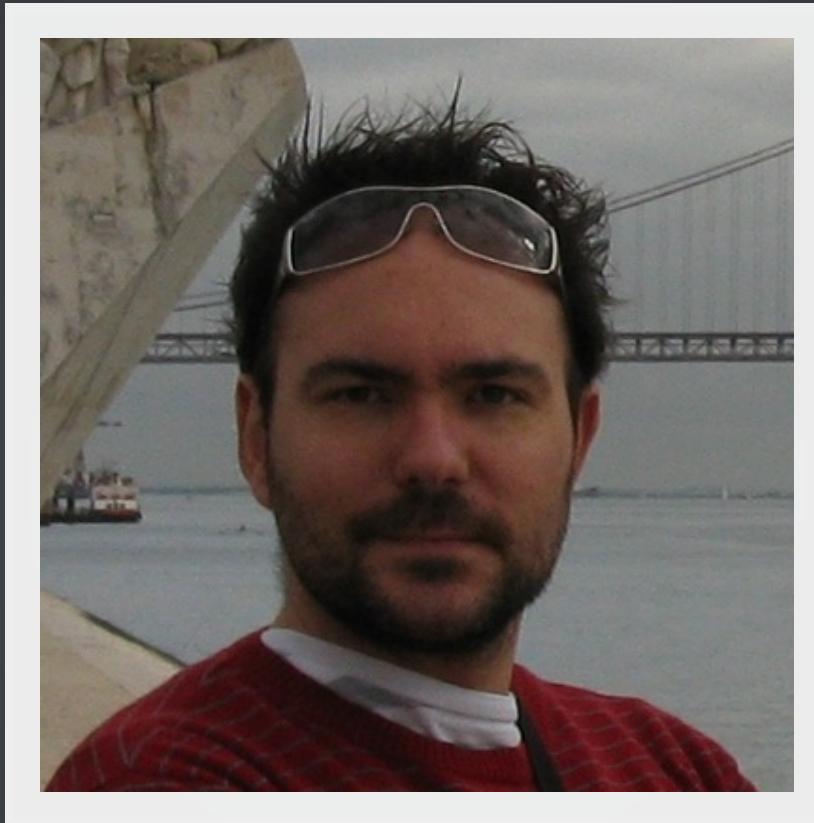
1.1 EL GUL

- 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



1.4 MI CV RESUMIDO

- **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**

1.5 HACKALOVER



- **Para los amantes de los hackathones**
- **Meetup:** <http://www.meetup.com/Hackathon-Lovers/>
- **Twitter:** <http://twitter.com/HackathonLovers>

1.6 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>

1.7 ¿DONDE 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>

1.8 CRÉDITOS

- Estas **transparencias** están hechas con:
 - <https://github.com/asanzdiego/reveal.js-pandoc-slides>

1.9 LICENCIA

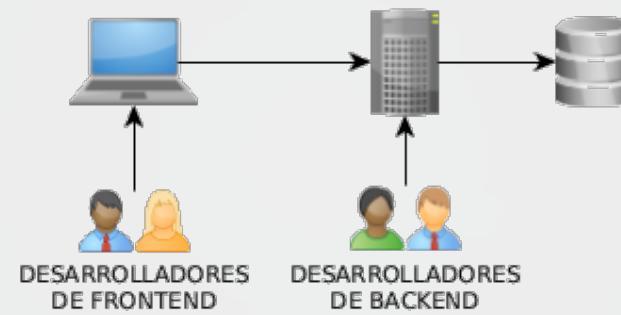
- Estas **transparencias** están bajo una licencia:
 - Creative Commons Reconocimiento-CompartirIgual 3.0
- El **código** de los programas están bajo una licencia:
 - GPL 3.0

1.10 FUENTES

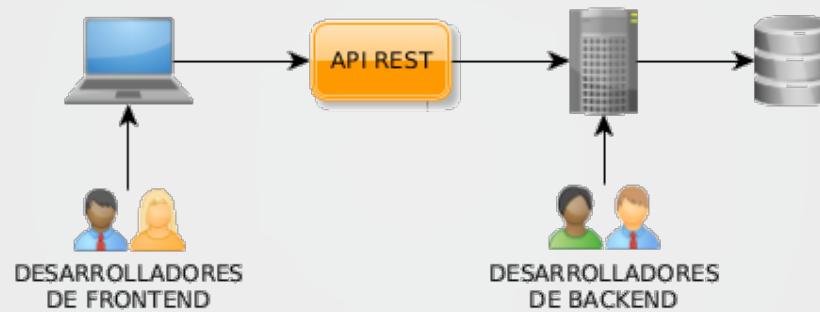
- Transparencias:
 - SlideShare:
 - <http://www.slideshare.net/asanzdiego/cliente-api-rest-con-lungojs>
 - GitHub Pages:
 - <http://asanzdiego.github.io/curso-api-restful-lungojs-client-2013/slides/export/api-restful-lungojs-client-reveal-slides>
- Código:
 - GitHub:
 - <http://github.com/asanzdiego/curso-api-restful-lungojs-client-2013/tree/master/src>

2 APIs ¿PARA QUÉ?

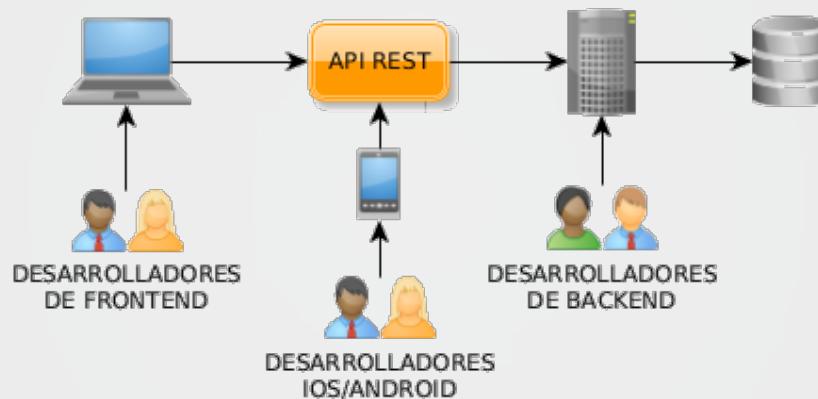
2.1 APLICACIÓN ESTÁNDAR



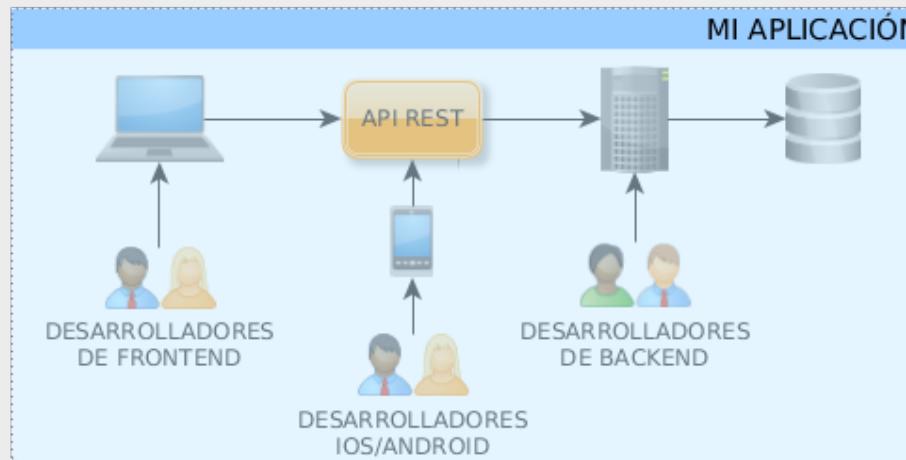
2.2 INTRODUCIMOS API



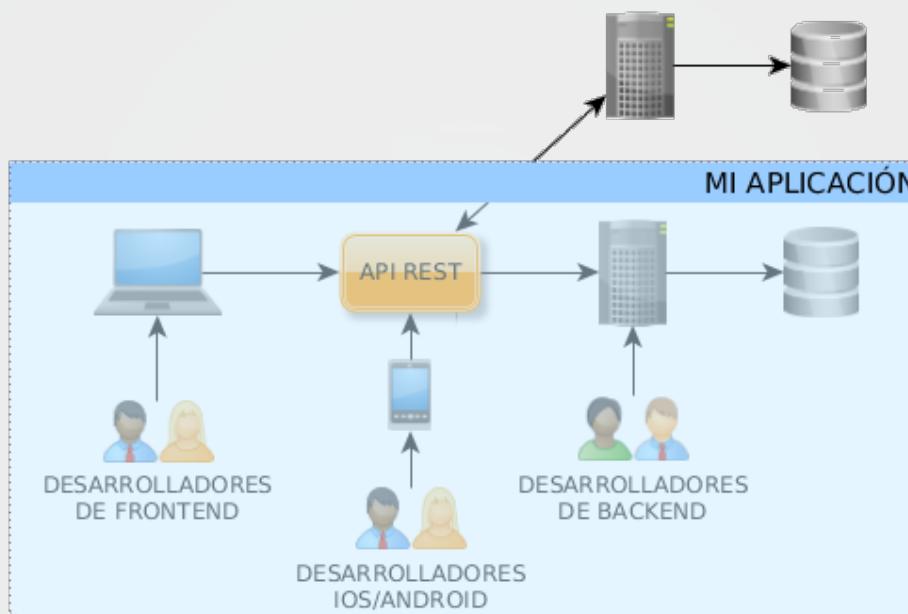
2.3 SEPARACIÓN ROLES



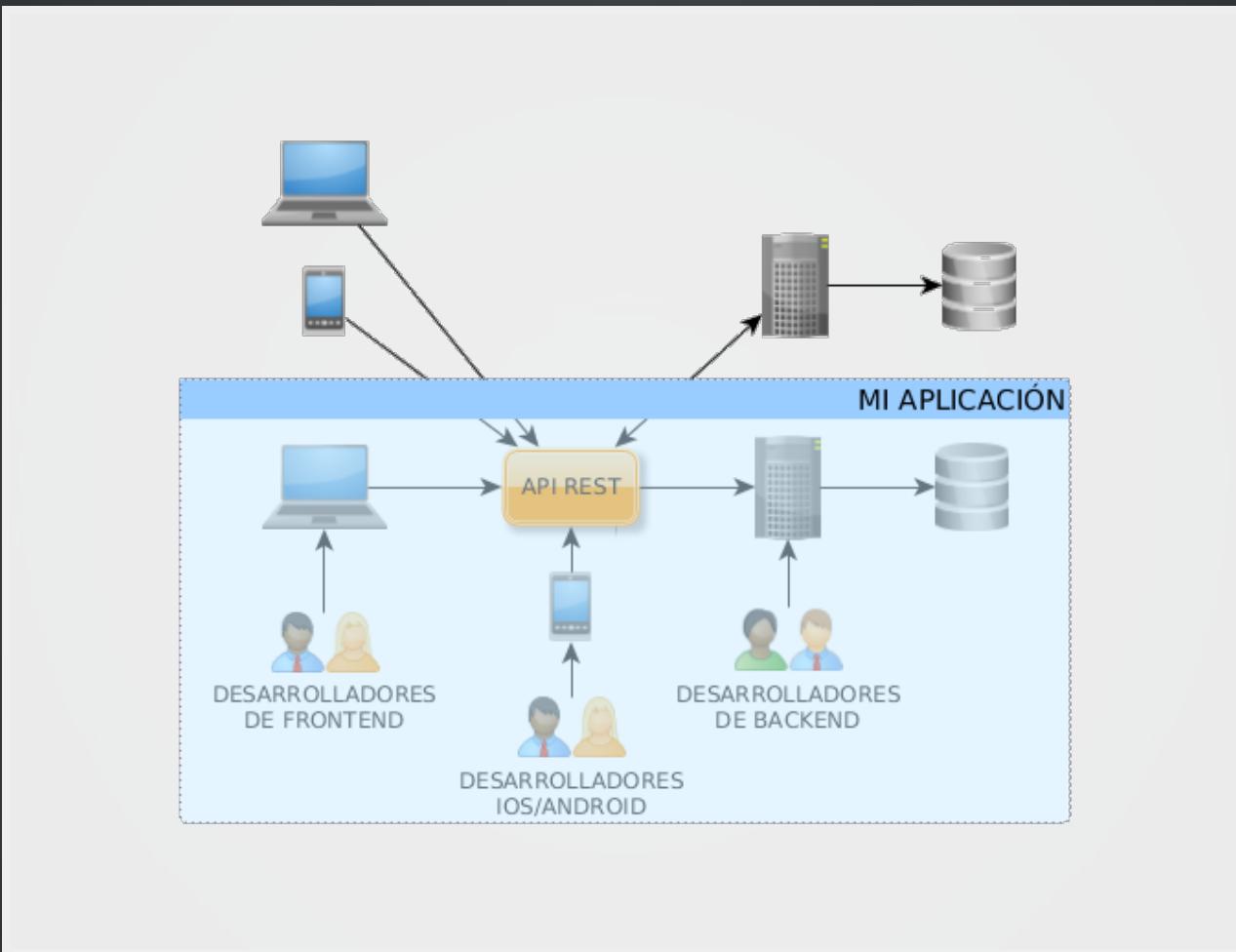
2.4 ¿Y AHORA QUÉ?



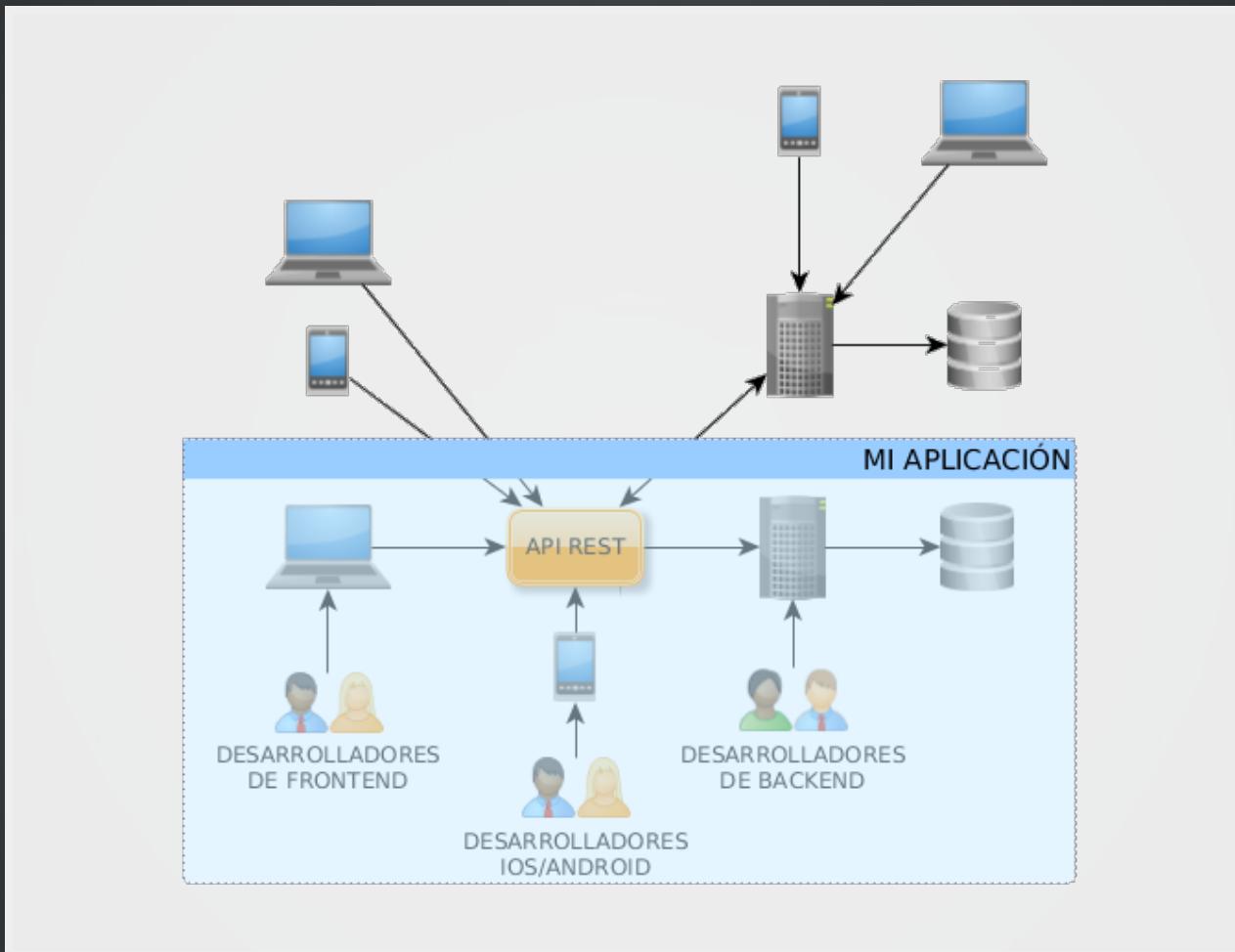
2.5 SERVICIOS EXTERNOS



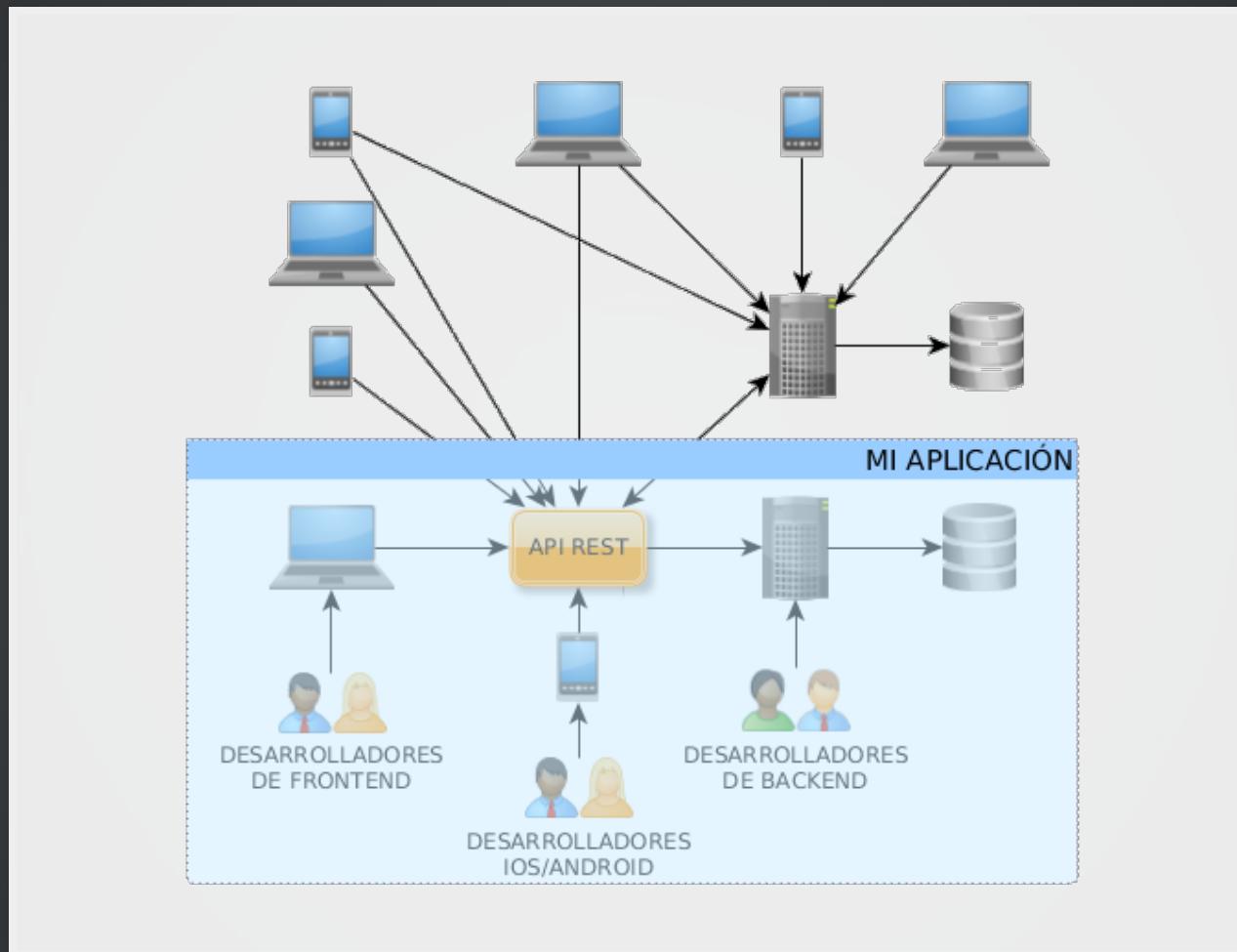
2.6 APPS CLIENTES



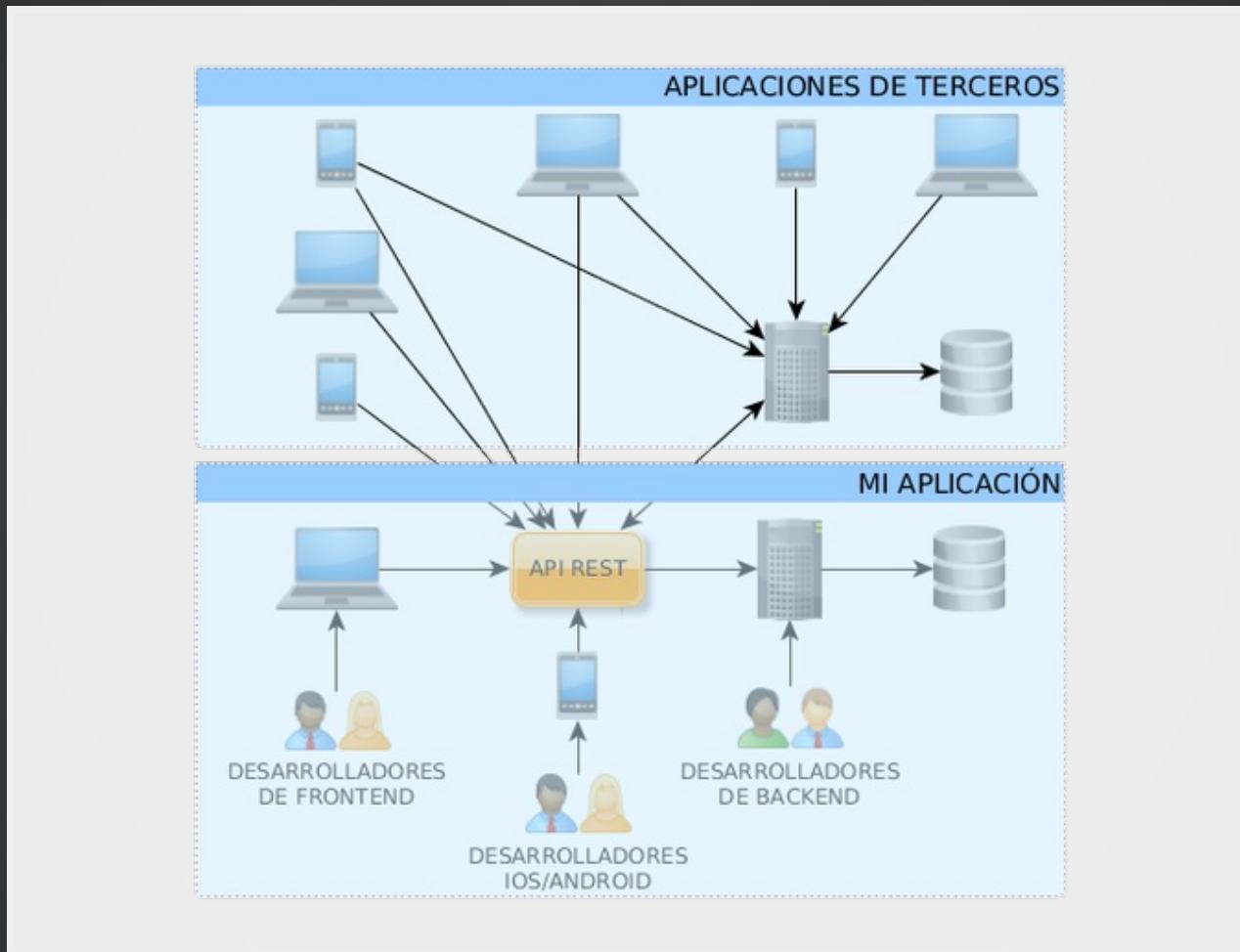
2.7 APPS DE SERVICIOS



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 livianas 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 RESTful indica que implementa dicha arquitectura.
 - Se utiliza como **adjetivo**
 - Se utiliza como por ejemplo: successful = **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 MULTIDISPOSITIVO

4.1 DISTINTOS DISPOSITIVOS



4.2 DISTINTOS OS



4.3 DISTINTOS OS



4.4 DISTINTOS NAVEGADORES



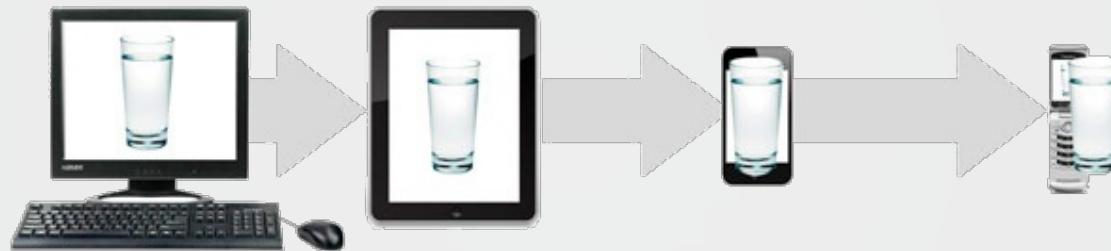
4.5 ¡¡¡SOCORRO!!!



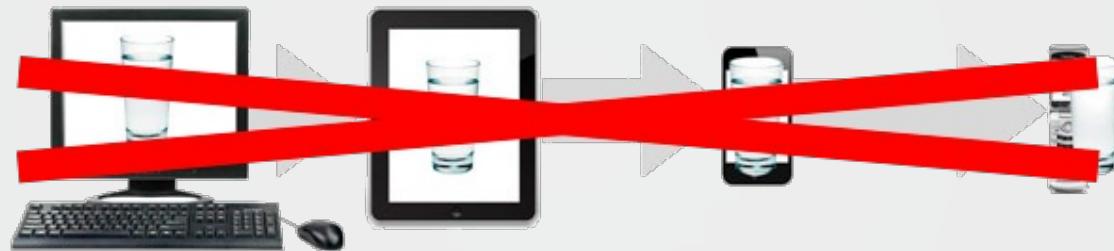
4.6 AL RESCATE



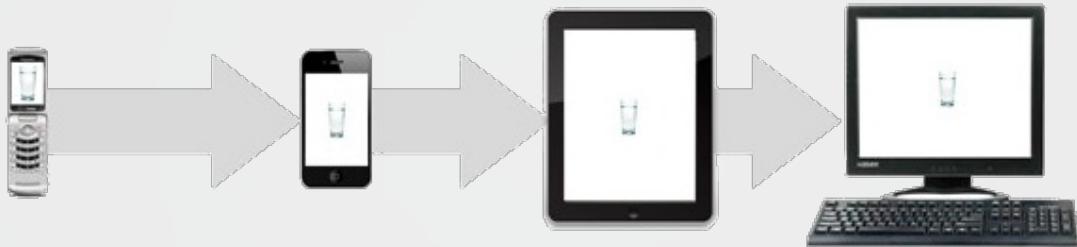
4.7 GRACEFUL DEGRADATION



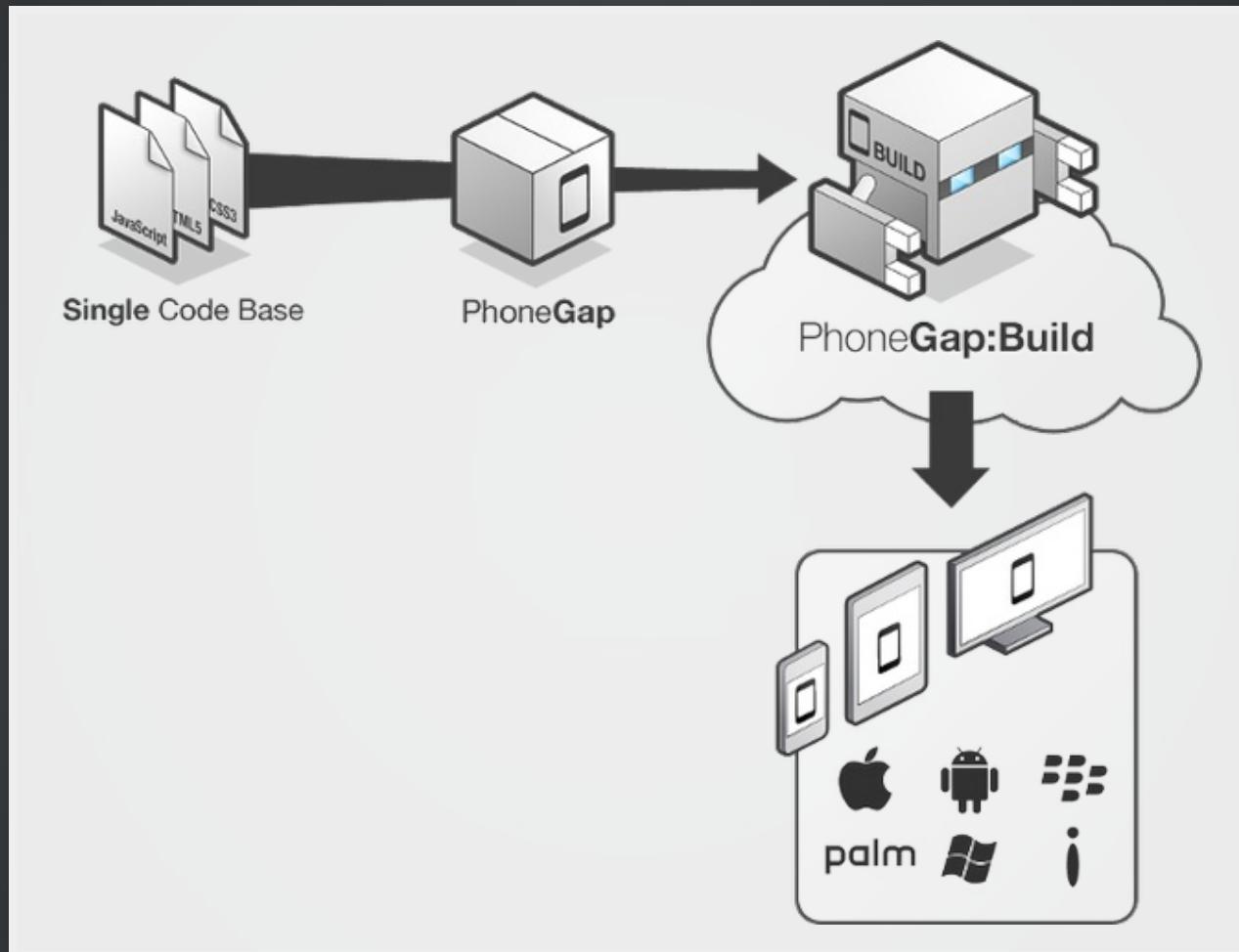
4.8 GRACEFUL DEGRADATION



4.9 PROGRESSIVE ENHANCEMENT



4.10 PHONEGAP



5 LUNGO.JS

5.1 COMPETIDORES

- jQTouch: <http://jqtjs.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?

I 
ESPAÑA

5.5 PUES NO



¡PUES NO!

¡ESTÁBAIS
EQUIVOCADOS!

5.6 ME GUSTA



¿A que
huelen las
nubes?

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

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**

A screenshot of a 'Add Talk' dialog box. The dialog has a blue header bar with the title 'Add Talk'. Below it is a form with five input fields: 'TALK NAME' (placeholder 'Enter the talk name'), 'TALK DATE' (placeholder 'dd/mm/yyyy hh:mm:ss'), 'SPEAKER NAME' (placeholder 'Enter the speaker name'), 'SPEAKER MAIL' (placeholder 'speaker@mail.es'), and 'TALK POINTS' (value '0'). At the bottom are two buttons: a blue 'Add' button on the left and a red 'Cancel' button on the right. The entire dialog is set against a white background with a dark grey border.

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>
```

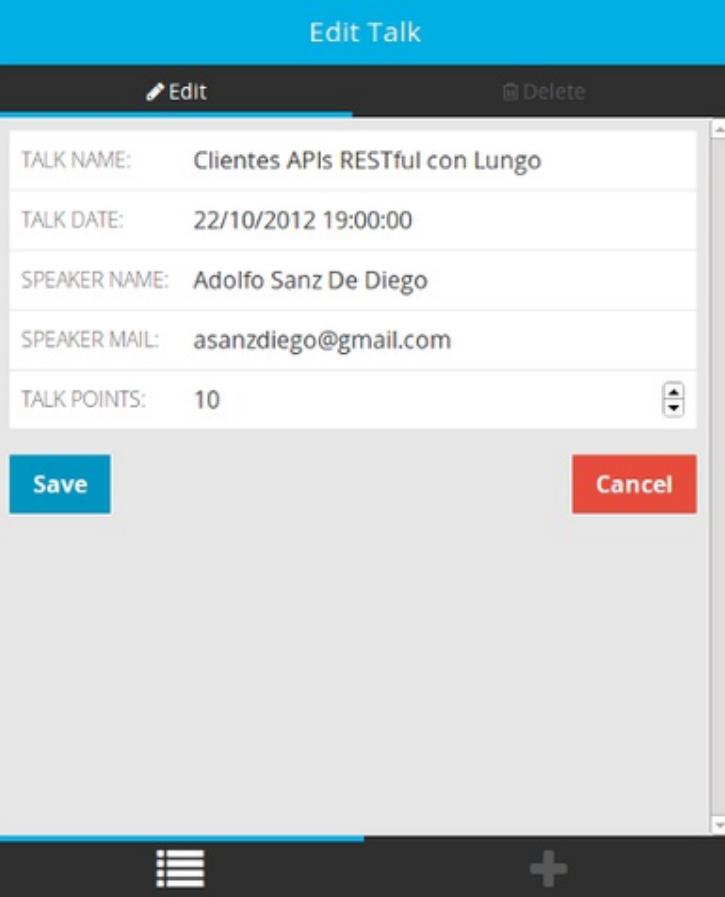
6.8 HTML/TALK/TALK-EDIT.HTML

Edit Talk

[Edit](#) [Delete](#)

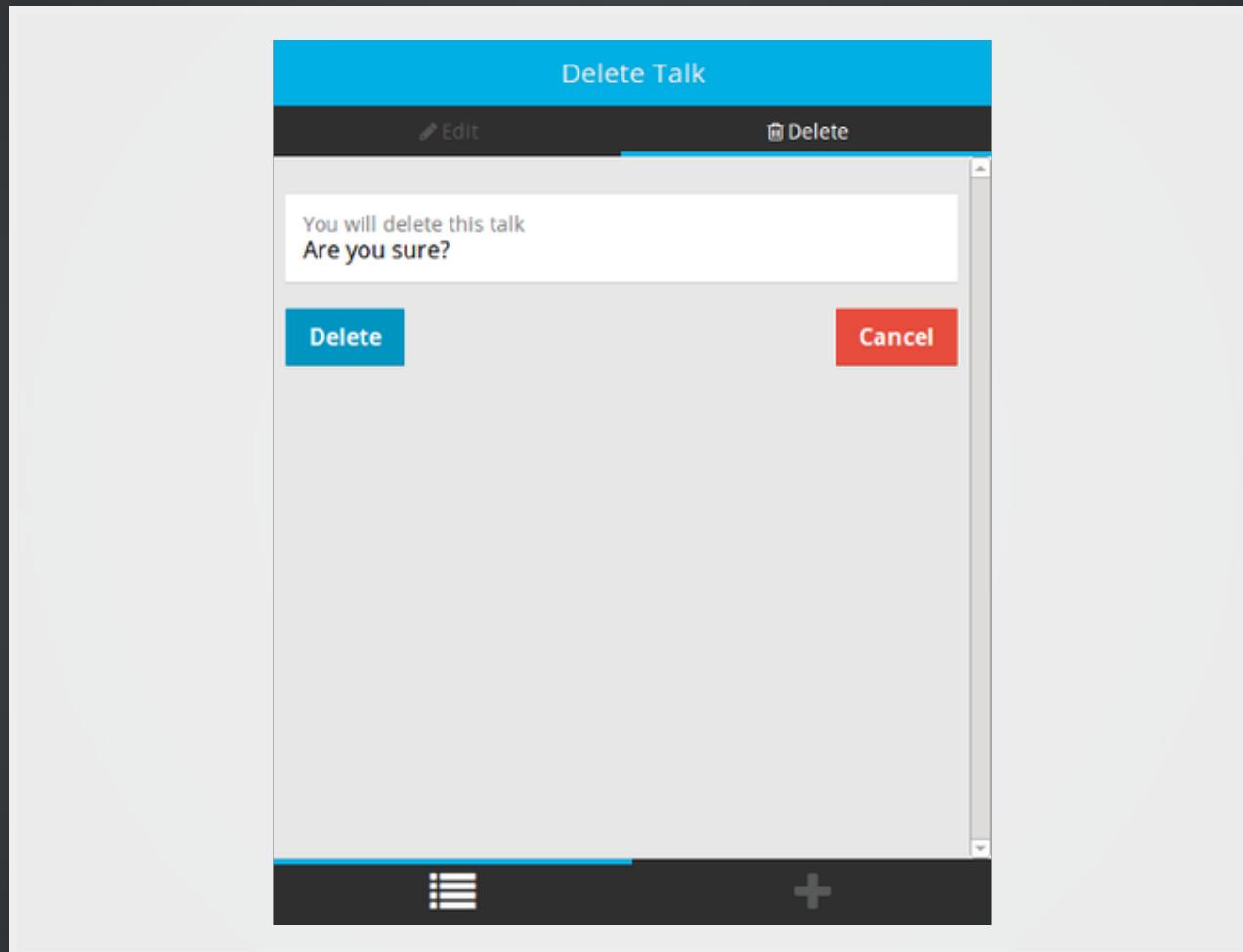
TALK NAME:	Cientes APIs RESTful con Lungo
TALK DATE:	22/10/2012 19:00:00
SPEAKER NAME:	Adolfo Sanz De Diego
SPEAKER MAIL:	asanzdiego@gmail.com
TALK POINTS:	10

[Save](#) [Cancel](#)



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(){

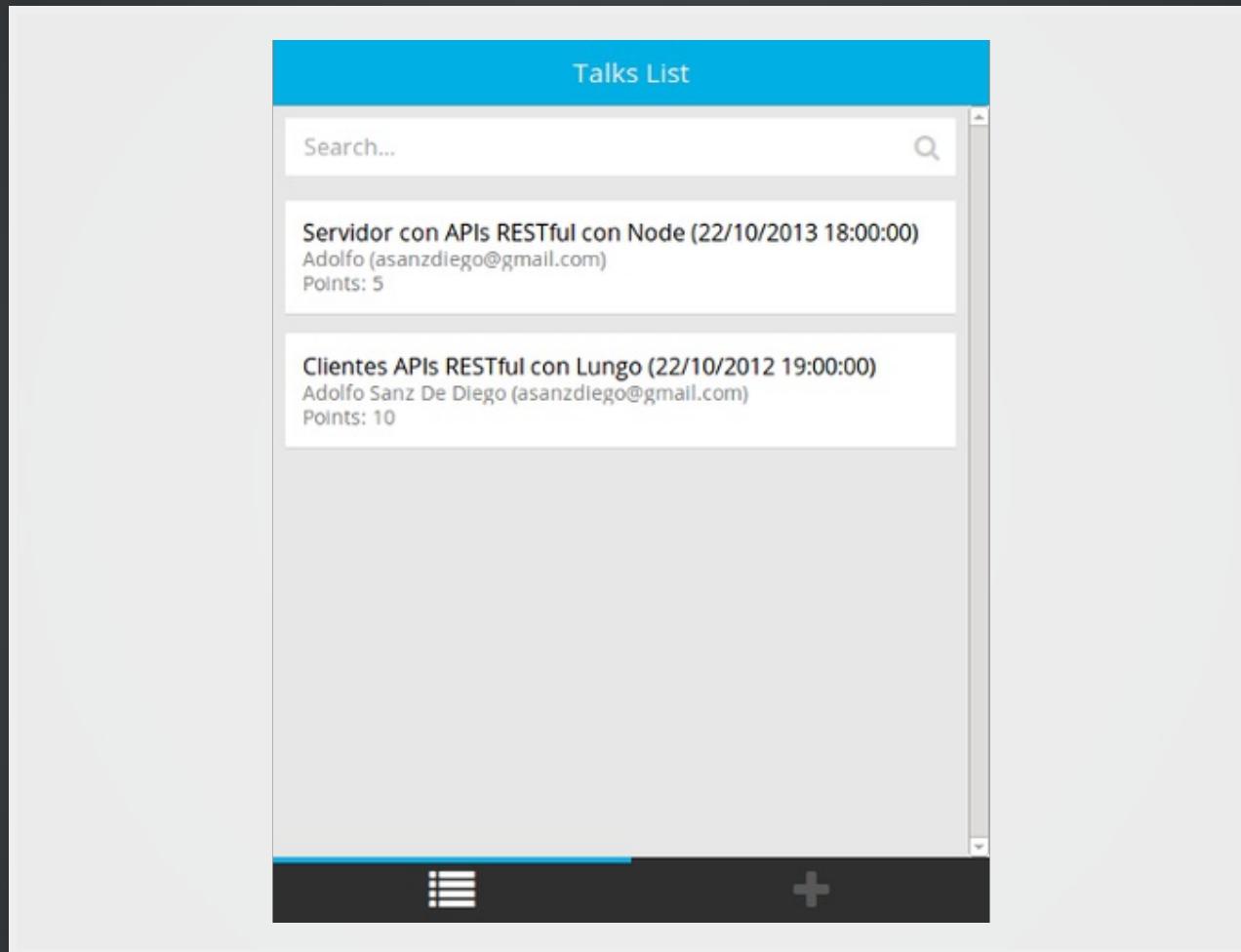
var talkId = util_urlParams["talkId"];

// 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');
  });
}
});
});
```

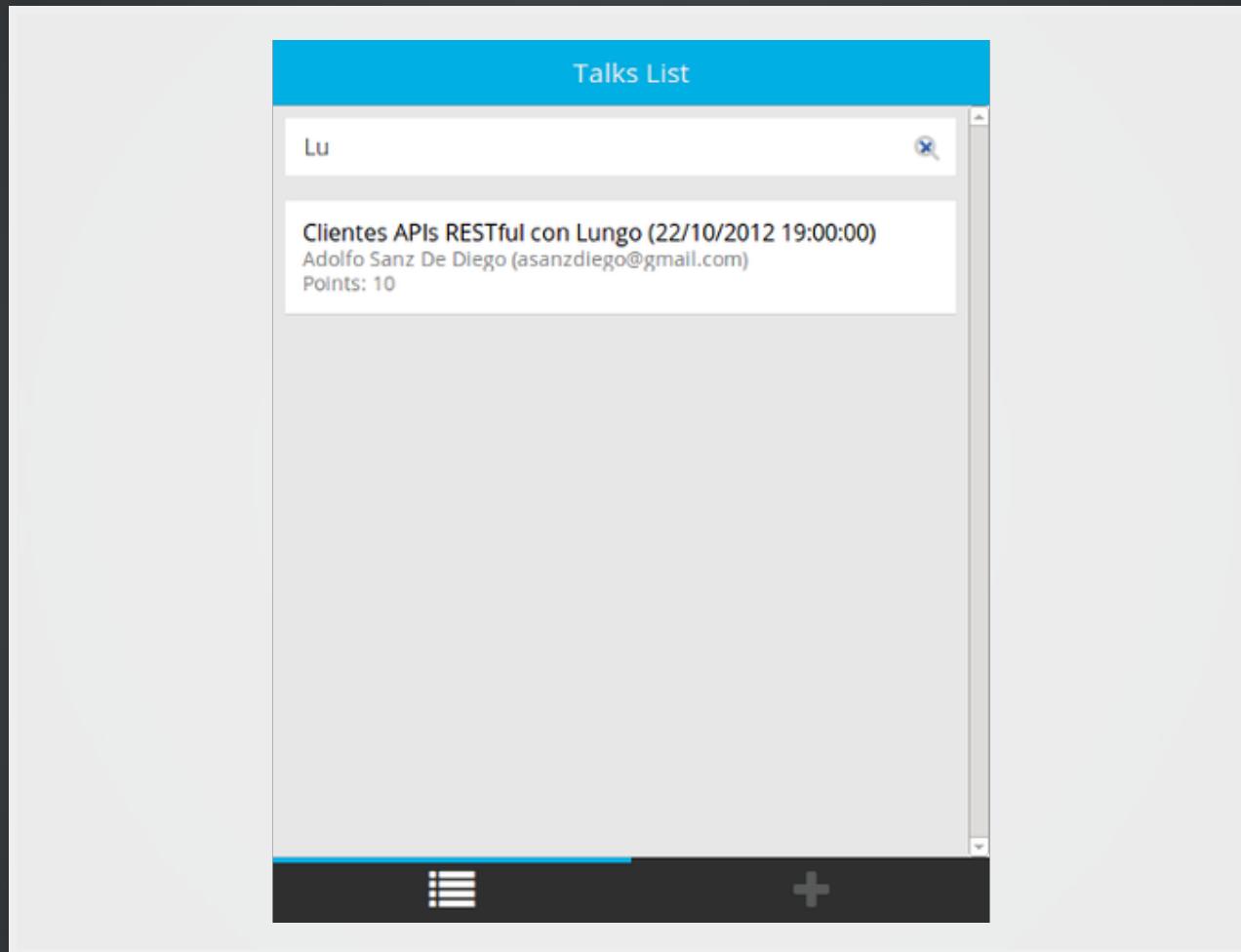
6.13 HTML/TALK/TALK-LIST.HTML

```
<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?