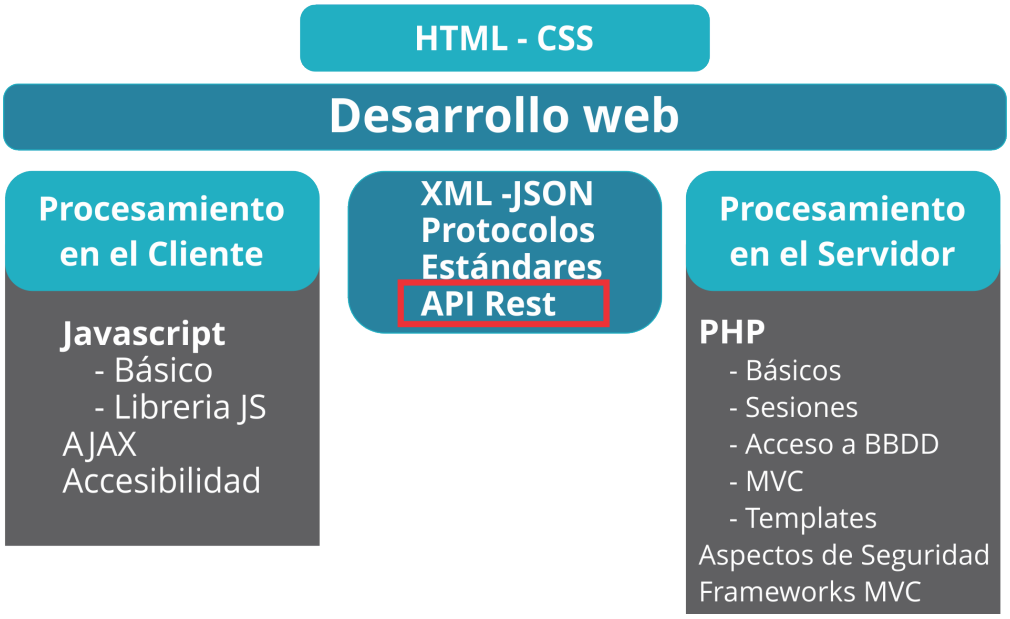
title: Clase 7 Proyecto 2016
Author: Einar Lanfranco, Claudia Banchoff, Matias Pagano, Diego Vilches
description: Notaciones para descripción de datos
keywords: APIs, OSM, REST
css: proyecto.css

Proyecto de Software

Cursada 2016

Hoy seguimos con ...



Temario

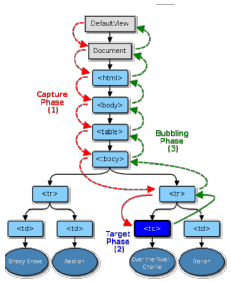
- **Repaso Clase Anterior**
    - DOM
    - AJAX
    - Librerías JS: jQuery
- APIs HTML5
- Trabajamos con mapas...
- API Rest

Repaso - DOM

- Es una **API**, que permite acceder a los contenidos de un documento HTML/XML.
- Proporciona una **interfaz estándar** para trabajar con **eventos**.
- El documento se ve como un **árbol de nodos**.
- **Interfaz Node:** con propiedades y métodos para acceder al árbol de nodos.
- **interfaz Document**: proporciona métodos para acceder y crear otros nodos en el árbol del documento.

Repaso - DOM

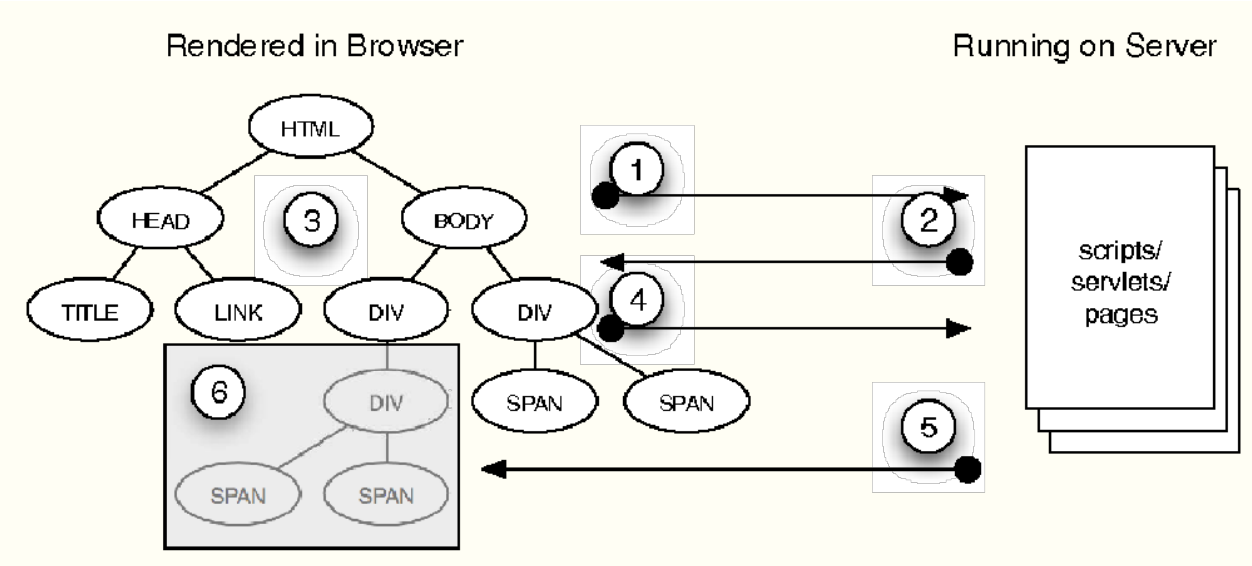
- DOM provee un sistema de eventos genérico que permita registrar **manejadores** de eventos, describir el **flujo de eventos** a través de la estructura del árbol y proveer **información contextual** sobre cada evento.



Repaso AJAX

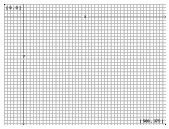
- NO es una tecnología, sino una combinación de varias tecnologías.
- **AJAX incluye:**
    - Presentación basada en estándares usando **HTML** y **CSS**;
    - Exhibición e interacción dinámicas usando **DOM**;
    - Intercambio y manipulación de datos usando **XML** y **XSLT**; (podemos usar otras notaciones también)
    - Recuperación de datos asincrónica usando **XMLHttpRequest**;
    - **JavaScript** como lenguaje de programación.

Repaso AJAX



Volviendo a HTML5: canvas

- Define un contexto usado para incorporar gráficos, imágenes, animaciones, etc.



```
var mi_canvas=document.getElementById("myCanvas");
var context=mi_canvas.getContext("2d");
context.fillStyle="#00ff00";
context.fillRect(0,0,150,75);
```

- Vemos ejemplo-Canvas
- +Info en http://diveintohtml5.info/canvas.html

¿Almacenamiento local?

- **¿Y las cookies?**
    - Limitadas en tamaño.
    - Viajan al servidor.
- **Almacenamiento Local de HTML5: Web storage**
    - Pares clave-valor: almacenados como strings
    - El evento **storage** se dispara cuando hay un cambio.
    - **localStorage vs. sessionStorage**

- Difieren en el alcance y el tiempo de vida.

Almacenamiento local

- **Algunos ejemplos de localStorage y sessionStorage**
    - Vemos ejemplo-storage
    - Otro ejemplo-completo
    - +Ínfo en http://diveintohtml5.info/storage.html

Geolocalización

- **Geolocalización**
    - +Info en http://diveintohtml5.info/geolocation.html
    - La API: http://www.w3.org/TR/geolocation-API/
- ¿Cómo lo hace? ¡Dependerá de cada browser!
    - **En general:**
        - GPS (si está disponible)
        - De acuerdo a las wifi-networks cercanas y la intensidad de la señal
        - De acuerdo a las torres de celular disponibles y la intensidad de la señal
        - IP Address lookup

Ejemplo Firefox: http://www.mozilla.org/en-US/firefox/geolocation/

¿Dijimos que hay muchas APIs?

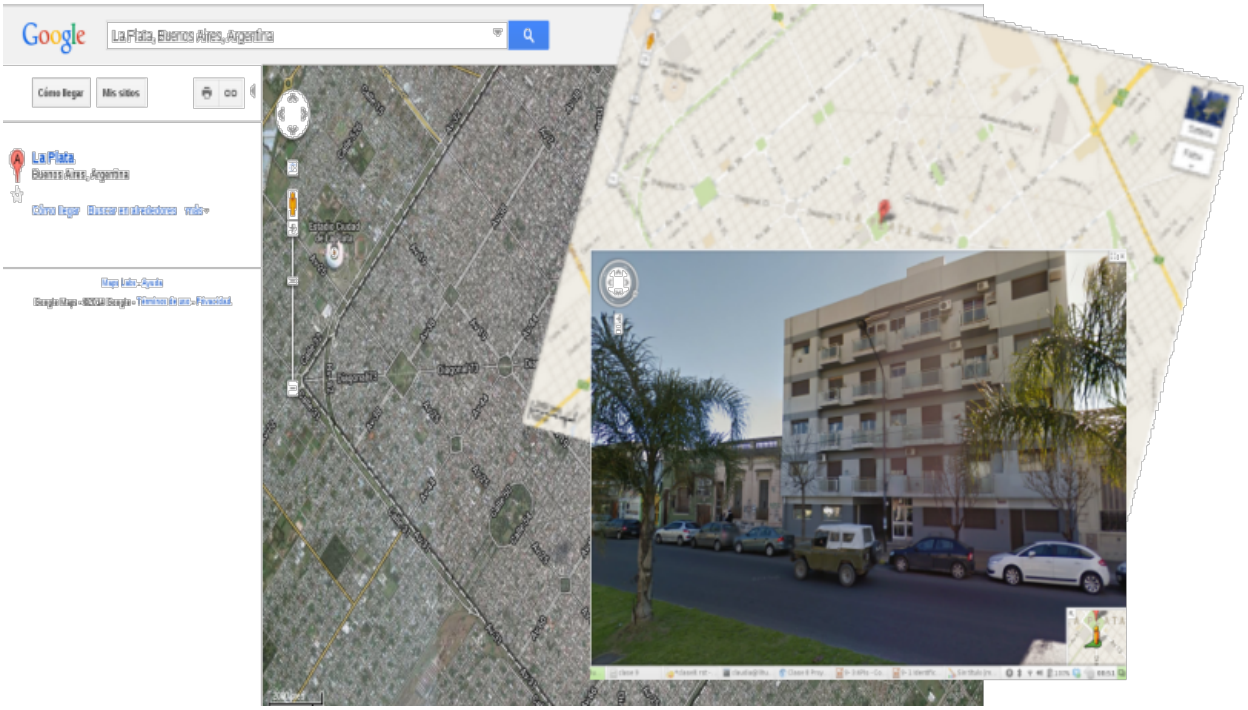
APIs para usar en el cliente

- Trabajo colaborativo
- Videoconferencias
- Mapas
- Autenticación y autorización
- Algunas más en https://developers.google.com/apis-explorer/#p/

¿Usamos mapas?

Cuando pensamos en mapas en la web ...

- Pensamos en Google Maps ...

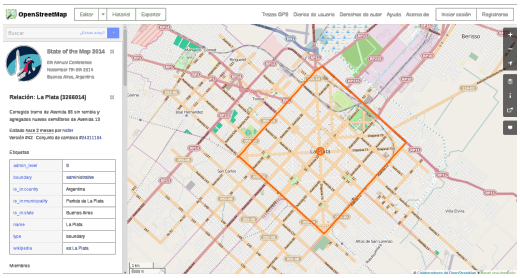


Los mapas de Google

- MUY usados.
- MUY completos.
- Con MUCHA funcionalidad.
- Street view.
- Disponibles.
- **PERO....**
    - NO son libres ....

¿Hay alternativas libres?

- Si: OpenStreetMap



El proyecto OpenStreetMap

- OpenStreetMap es Open Data (un servicio de datos de acceso libre), con licencia Open Data Commons Open Database License (ODbL).
- La cartografía está licenciada bajo la licencia Creative Commons Reconocimiento-CompartirIgual 2.0 (CC-BY-SA).
- http://wiki.openstreetmap.org/wiki/Main_Page
- Mucho proyectos para contribuir: http://wiki.openstreetmap.org/wiki/Develop
- Rutas: http://map.project-osrm.org/
- Usado por varias empresas como Foursquare
- Nosotros... sólo usaremos aspectos básico de la API.

Telegram Bot

Telegram Bot

- Que es Telegram?
    - Aplicación de mensajería instantanea
    - Basado en la cloud
    - Foco en la velocidad y seguridad
    - Se pueden crear grupos de hasta 5000 miembros
    - Broadcast sin límites de usuarios
    - Multiplataforma

Telegram Bot

- **Que es un bot?**

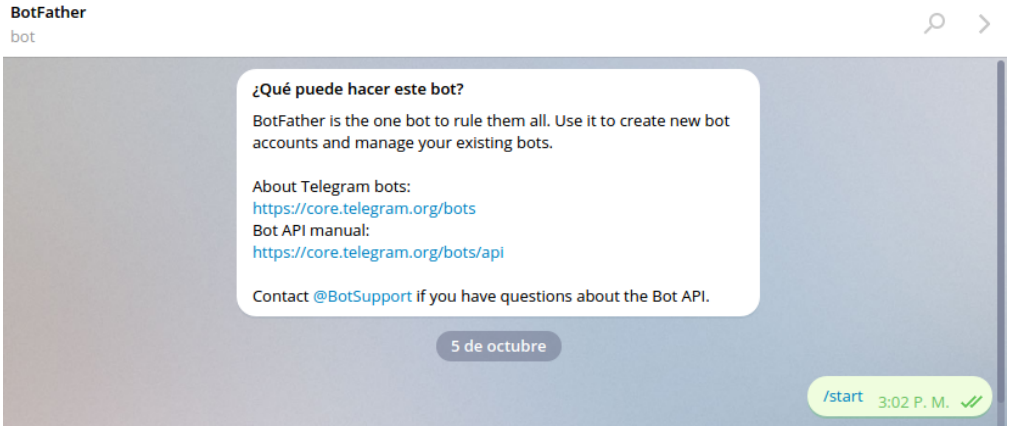
- Es un software que imita el comportamiento humano
- **Se puede utilizar para ...**
    - ... responder preguntas en un chat
    - ... simular un jugador en un videojuego
    - ... etc, etc

Telegram Bot

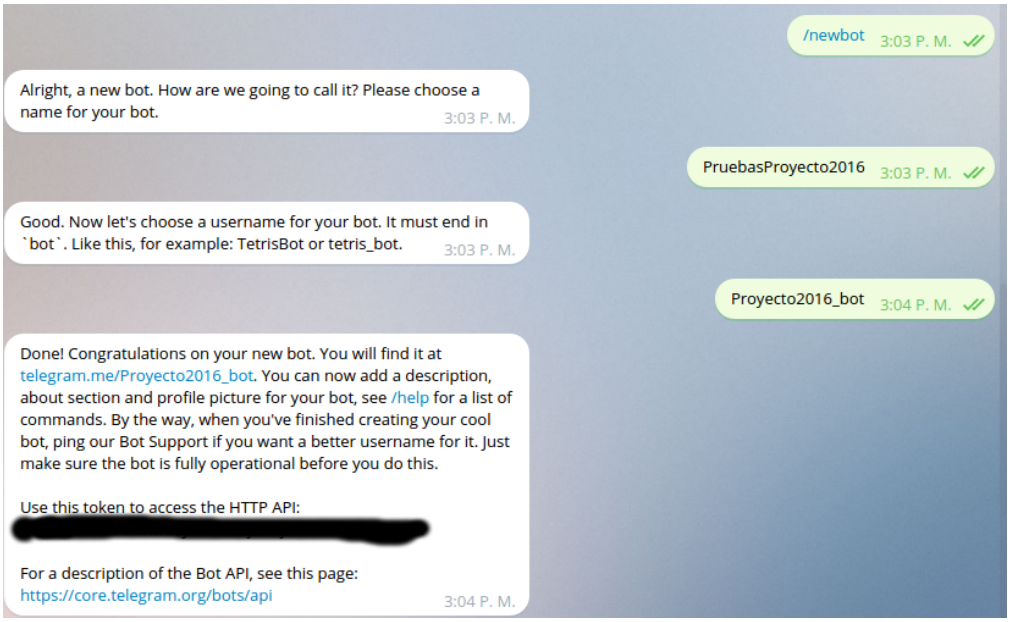
- **Bot API**
    - Corre sobre HTTPs
    - Versión reducida de la API de Telegram
    - No necesita el número de teléfono para enviar mensajes
    - Certificado ssl válido

Telegram Bot

- **Creando nuestro Bot**
    - @BotFather



Telegram Bot



Telegram Bot

- Seteamos el bot con nuestro sistema

https://api.telegram.org/bot123456780:hakfHFT35kvdkfhkffdgdkgh878sfsfskghs/setWebhook?url=https://grupo80.proyecto2016.linti.unlp.edu.ar/pruebasbot.php

- Respuesta Ok

```
{"ok":true,"result":true,"description":"Webhook was set"}
```

Telegram Bot

Respuesta al enviar /menu

```
{"update_id":361480117,
 "message":{"message_id":184,
            "from":{"id":12345678,
                    "first_name":"Diego",
                    "username":"dvilches"},
            "chat":{"id":12345678,
                    "first_name":"Diego",
                    "username":"dvilches",
                    "type":"private"},
            "date":1476120013,
            "text":"\/menu",
            "entities":[{"type":"bot_command","offset":0,"length":14}]}}
```

Telegram Bot

Recibiendo los datos

```php
<?php

$returnArray = true;
$rawData = file_get_contents('php://input');
$response = json_decode($rawData, $returnArray);
$id_del_chat = $response['message']['chat']['id'];


// Obtener comando (y sus posibles parametros)
$regExp = '#^(\/[a-zA-Z0-9\/]+?)(\ .*?)$#i';


$tmp = preg_match($regExp, $response['message']['text'], $aResults);

if (isset($aResults[1])) {
    $cmd = trim($aResults[1]);
    $cmd_params = trim($aResults[2]);
} else {
    $cmd = trim($response['message']['text']);
    $cmd_params = '';
}
```

Telegram Bot

Armando la respuesta

```php
<?php

$msg = array();
$msg['chat_id'] = $response['message']['chat']['id'];
$msg['text'] = null;
$msg['disable_web_page_preview'] = true;
$msg['reply_to_message_id'] = $response['message']['message_id'];
$msg['reply_markup'] = null;
```

Telegram Bot

Armando la respuesta

```php
<?php
switch ($cmd) {
    case '/start':
        $msg['text']  = 'Hola ' . $response['message']['from']['first_name'] . PHP_EOL;
        $msg['text'] .= '¿Como puedo ayudarte? Puedes utilizar el comando /help';
        $msg['reply_to_message_id'] = null;
        break;
    case '/help':
        $msg['text']  = 'Los comandos disponibles son estos:' . PHP_EOL;
        $msg['text'] .= '/start Inicializa el bot';
        $msg['text'] .= '/menú Muestra el menú del día';
        $msg['text'] .= '/help Muestra esta ayuda';
        $msg['reply_to_message_id'] = null;
        break;
    case '/menú':
        $msg['text']  = 'El menú del día es ensalada tropical';
        break;
    default:
        $msg['text']  = 'Lo siento, no es un comando válido.' . PHP_EOL;
        $msg['text'] .= 'Prueba /help para ver la lista de comandos disponibles';
        break;
}
```

Telegram Bot

Enviando la respuesta

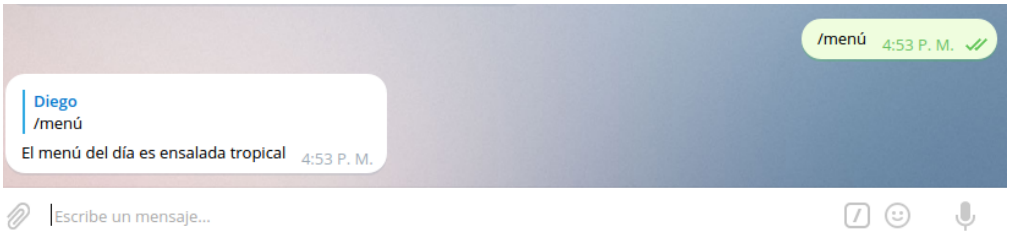
```php
<?php
$url = 'https://api.telegram.org/bot123456780:hakfHFT35kvdkfhkffdgdkgh878sfsfskghs/sendM

$options = array(
'http' => array(
    'header'  => "Content-type: application/x-www-form-urlencoded\r\n",
    'method'  => 'POST',
    'content' => http_build_query($msg)
    )
);

$context  = stream_context_create($options);
$result = file_get_contents($url, false, $context);

exit(0);
```

Telegram Bot

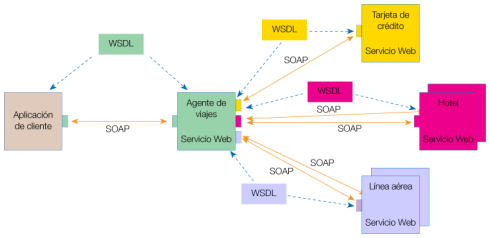


Servicios web

Servicios web

- Un **servicio web** es una tecnología que utiliza un conjunto de protocolos y estándares para intercambiar datos entre aplicaciones.
- Algo importante: lograr **interoperabilidad**.
- **Uso de estándares abiertos**
    - XML-RPC
    - SOAP: Simple Object Access Protocol
    - WSDL: Web Services Description Language
    - ¿Rest?

Servicios web



- Ejemplo típico: http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb

---

¿Qué es REST?

- **REpresentational State Transfer**: arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP.

- REST nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP, por lo que es más simple y convencional que otras alternativas que se han usado en los últimos diez años como SOAP y XML-RPC.
- REST fue definido por Roy Fielding (coautor de la especificación HTTP) en el año 2000.

REST

- Los sistemas que siguen los principios REST se los denomina también RESTful.
- Se basa en HTTP para intercambiar información.
- SIN estado.
- Se piensa en los recursos como una entidad que puede ser accedido públicamente.
- Ejemplo: misrecetas.com/api/v1/receta/chocotorta
- Cada objeto tiene su propia URL y puede ser fácilmente cacheado, copiado y guardado como marcador.

Recordemos que ....

Una petición HTTP consta de:

- Una URL y un método de acceso (GET, POST, PUT,...).
- Cabeceras. Meta-información de la petición.
- Cuerpo del mensaje (opcional).

Métodos HTTP

- GET: Usado para solicitar un recurso al servidor.
- PUT: Usado para modificar un recurso existente en el servidor.
- POST: Usado para crear un nuevo recurso en el servidor.
- DELETE: Usado para eliminar un recurso en el servidor

¿POST = GET + PUT + DELETE?

- Por lo geneneral, las peticiones de tipo PUT y DELETE son realizadas a través de peticiones POST.
- La petición POST se utiliza tanto para crear, borrar o actualizar un recurso.
- Pero hay una diferencia: POST NO es idempotente.

Métodos HTTP

HTTP Method	Idempotent	Safe
OPTIONS	yes	yes
GET	yes	yes
HEAD	yes	yes
PUT	yes	no
POST	no	no
DELETE	yes	no
PATCH	no	no

Mas info ver: rfc7231 y rfc5789

Accediendo a los recursos

- La implementación del recurso decide qué información es visible o no desde el exterior, y qué representaciones de dicho recurso se soportan.
- **Podríamos pensar en:**
    - HTML
    - XML
    - JSON
- Ejemplo: Consultamos feriados?

Accediendo a los recursos

- Probemos con **curl**:

```
curl -X GET https://api.mercadolibre.com/categories/MLA5725
```

- **Otros ejemplos:**
    - API REST de Mercado Libre: http://developers.mercadolibre.com/api-directory/
    - Google Translate (es necesario API_KEY): https://developers.google.com/apis-explorer/#p/translate/v2/

Ventajas / Desventajas

- Separación cliente/servidor
- Simplicidad
- Seguridad
- Uso de estándares
- Escalabilidad
- Cambio de esquema: Usando REST podemos tener varios servidores donde unos no saben que los otros existen.

+Info: http://www.desarrolloweb.com/articulos/ventajas-inconvenientes-apirest-desarrollo.html

---

Generando API REST

- A mano.... o,
- **Muchos frameworks que facilitan el desarrollo:**
    - Slim: http://www.slimframework.com/
    - Symfony,
    - Laravel,
    - ...

Referencias Rest

http://martinfowler.com/articles/richardsonMaturityModel.html
http://restcookbook.com/Miscellaneous/richardsonmaturitymodel/
http://www.restapitutorial.com/lessons/whatisrest.html
http://asiermarques.com/2013/conceptos-sobre-apis-rest/ http://rest.elkstein.org/
http://restfulwebapis.org/rws.html