

# MODERN WEB APPS

mejorando la **productividad** y la **felicidad**



**Desarrollar  
actualmente para la  
web es como...**



**La CANTIDAD de  
herramientas, lenguajes  
y pseudo-lenguajes lo  
hace muy complejo.**

HTML CSS2  
CSS3 HTML5 JavaScript  
jQuery AJAX SASS LESS Stylus  
CoffeeScript CORS AngularJS  
Backbone HandleBars  
HAML

**Boilerplate Abstracciones**  
**Frameworks Testing**  
**Documentación**  
Flujo de trabajo Gestión de  
**dependencias Rendimiento**  
Despliegue Integración continua  
Control de versiones

# **AUTOMATIZAR**

tareas repetitivas para ser eficientes.



Esto no significa ser vagos, sino ser  
**EFICIENTES.**

# EL FLUJO DE TRABAJO ACTUAL EN EL DESARROLLO WEB (FRONT-END)

# SETUP

- Crear una estructura
- Descargar plantillas
- Descargar librerías
- Descargar frameworks

# DE SARROLLO

- Observar Sass / Less / Stylus
- Observar CoffeeScript
- Observar Jade / Haml
- JS / CSS Linting

# CONSTRUCCIÓN

- Test unitarios
- Compilar JS / CSS
- Minimizar y unificar
- Generar imágenes e iconos
- Optimizar el rendimiento
- Desplegar

# ¿CÓMO ENFRENTARNOS A ESTO?





**PRIMERO HAZ QUE FUNCIONE.**  
**DESPUÉS HAZLO** *bien.*  
**Y FINALMENTE HAZLO** *mejor.*





# YEOMAN

# ¿Qué es YEOMAN?



**El término Yeoman se define como un noble “sirviente”.**  
**Él trabaja para nosotros y solo nosotros.**

# ¿Qué es YEOMAN?

**Un conjunto de herramientas, librerías y procesos que nos ayudan a crear aplicaciones web.**

**Agilizar la creación de proyectos y tareas repetitivas.**

**Centrarse en el desarrollo.**

# YEOMAN



YO



GRUNT



BOWER



YO

crea la **estructura** de la aplicación,  
creando una configuración para Grunt,  
añadiendo tareas y definiendo  
dependencias para el tipo de proyecto.



# GRUNT

es utilizado para desplegar, visualizar y testear el proyecto, a partir de las tareas definidas por YO.



## BOWER

es utilizado para **gestionar** las dependencias, por lo que no será necesario descargar manualmente las librerías.

# INSTALACIÓN

```
npm install -g yo
```

Instalará Grunt y Bower de manera automática.

# GENERADORES

<http://yeoman.io/official-generators.html>

<http://yeoman.io/community-generators.html>

YO puede crear muchos tipos de aplicaciones, pero  
necesita de generadores para crear las estructuras.

# GENERADORES

```
npm install -g generator-webapp
```

Este generador creará proyectos que contienen HTML5  
Boilerplate, jQuery, Modernizr y Bootstrap.

# GENERADORES

Ahora que el generador está instalado, pasamos a crear el proyecto.

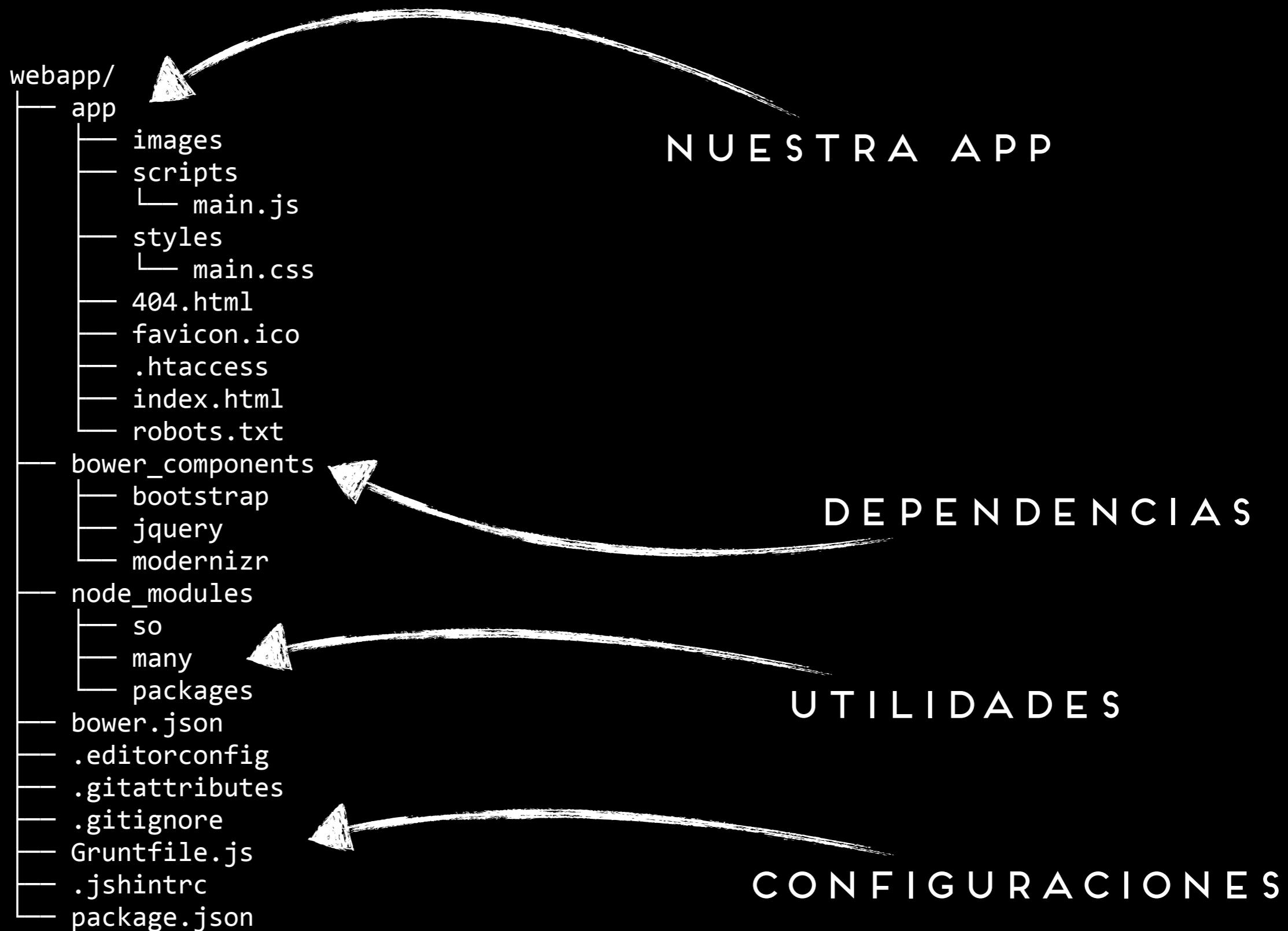
```
mkdir my-yo-project  
cd my-yo-project
```

```
yo webapp
```

```
2. entornos@ubuntu-server: ~/src_git/webapp (ssh)
entornos@ubuntu-server:~/src_git/webapp$ yo webapp

      _-----_
     |       |
     |   (o)  |
     \_ '-' _/ | Welcome to Yeoman,
     /__A__\  | ladies and gentlemen!
     | ~ |
     \_.'_.-' /
          ' Y '


Out of the box I include HTML5 Boilerplate, jQuery, and a Gruntfile.js to build
your app.
[?] What more would you like?
  • Bootstrap
  ○ Sass
  ➤ Modernizr
```



# GENERADORES

## AVANZADOS

Algunos generadores incluyen opciones extra para crear vistas, modelos, controladores, etc...

Un ejemplo es AngularJS

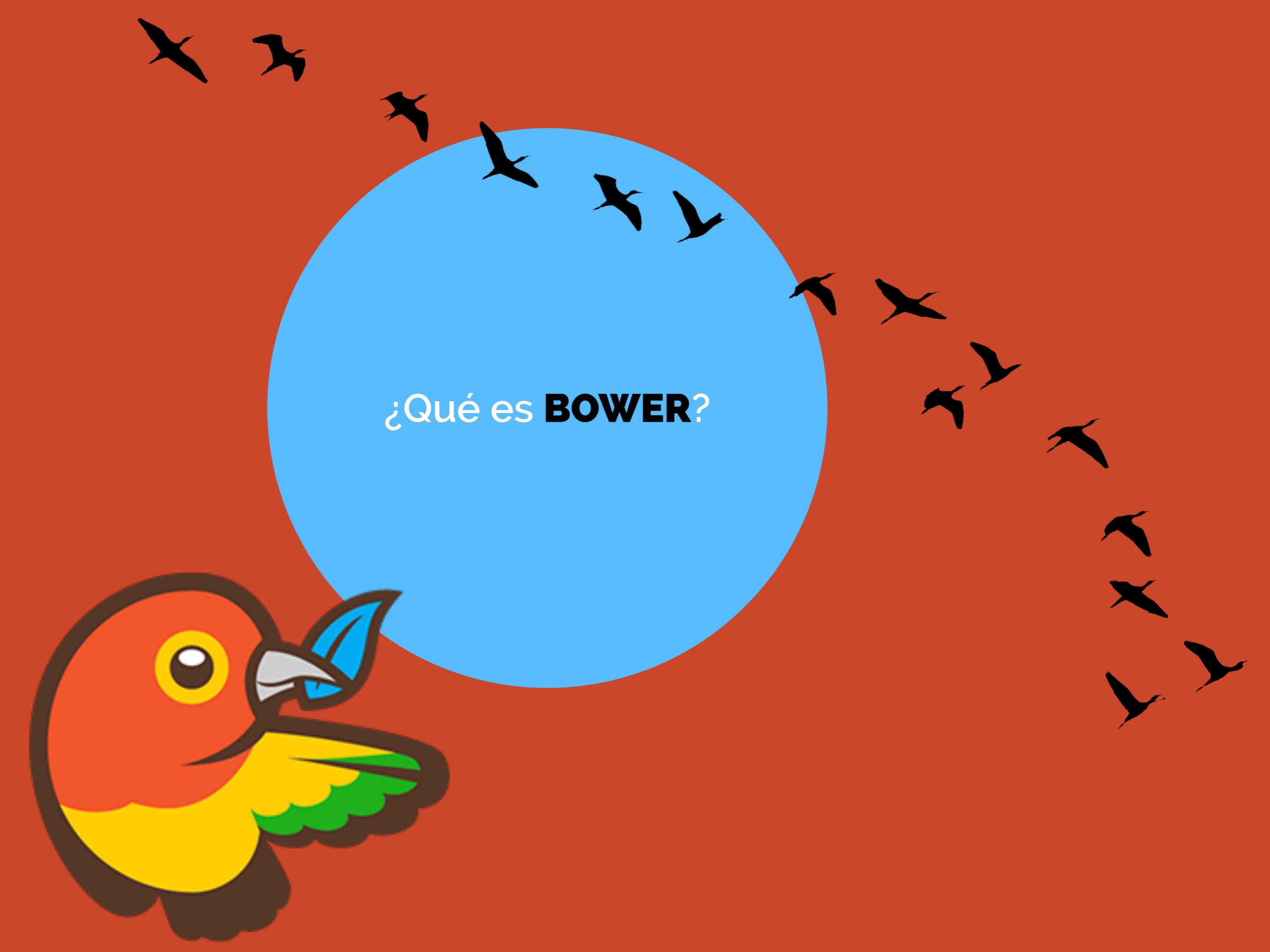
```
npm install -g generator-angular
```

```
yo angular --minsafe
```

```
yo angular:controller myController  
yo angular:directive myDirective  
yo angular:filter myFilter  
yo angular:service myService
```

En este momento, YO  
ha terminado su trabajo,  
creando y configurando  
las herramientas  
necesarias.





¿Qué es **BOWER**?

# BOWER

**Es un gestor de paquetes que nos permite  
controlar las dependencias en nuestros  
proyectos de una manera muy sencilla.**





# BOWER

## The old days

1. ¿Una nueva versión de jQuery?  
Habrá que actualizar.
2. Ir a [jquery.com/download](http://jquery.com/download)
3. Descargar el fichero concreto
4. Copiar el fichero de ~/Downloads
5. Pegar el fichero en app/js/libs
6. Renombrar el fichero
7. Actualizar los enlaces en HTML
8. ...

# BOWER

pone a nuestra disposición una serie de comandos para gestionar las dependencias.



bower list

```
bower check-new      Checking for new versions of the project dependencies..
webapp /home/entornos/src_git/webapp
└── bootstrap#3.0.3 (latest is 3.1.1)
    ├── jquery#1.11.0 (2.1.1-beta1 available)
    └── jquery#1.11.0 (1.11.1-beta1 available, latest is 2.1.1-beta1)
        └── modernizr#2.6.3 (latest is 2.7.2)
```

# BOWER

Para actualizar las dependencias del proyecto, modificamos el fichero bower.json creado por YO.

```
{  
  "name": "webapp",  
  "private": true,  
  "dependencies": {  
    "bootstrap": "~3.1.1",  
    "modernizr": "~2.7.2",  
    "jquery": "~2.1.0"  
  },  
  "devDependencies": {}  
}
```



# BOWER



Indicamos a bower que actualice las dependencias

**bower update**

```
bower check-new    Checking for new versions of the project dependencies..
webapp /home/entornos/src_git/webapp
└── bootstrap#3.1.1
    └── jquery#2.1.0 (2.1.1-beta1 available)
    └── jquery#2.1.0 (2.1.1-beta1 available)
    └── modernizr#2.7.2
```



# BOWER

también nos permite instalar nuevas dependencias de manera muy sencilla.

```
bower install underscore
```

```
bower check-new      Checking for new versions of the project dependencies..
webapp /home/entornos/src_git/webapp
└── bootstrap#3.1.1
    ├── jquery#2.1.0 (2.1.1-beta1 available)
    ├── jquery#2.1.0 (2.1.1-beta1 available)
    └── modernizr#2.7.2
        └── underscore#1.6.0 extraneous
```



¿EXTRAÑO?

# BOWER

Indicando el parámetro --save, la nueva dependencia es añadida al  
ficher bower.json

```
bower install underscore --save
```

```
bower check-new      Checking for new versions of the project dependencies..
webapp /home/entornos/src_git/webapp
└── bootstrap#3.1.1
    ├── jquery#2.1.0 (2.1.1-beta1 available)
    ├── jquery#2.1.0 (2.1.1-beta1 available)
    └── modernizr#2.7.2
        └── underscore#1.6.0
```



# BOWER

Listado de paquetes disponibles

<http://bower.io/search/>



|OK + !!!



En este momento, las dependencias del proyecto  
se encuentran definidas, pero no incluidas en el  
fichero HTML.

Sin RequireJS

`grunt bowerInstall`

Con RequireJS

`grunt bower`

# BOWER



```
<!-- build:js scripts/vendor.js -->
<!-- bower:js -->
<script src="../bower_components/jquery/dist/jquery.js"></script>
<script src="../bower_components/underscore/underscore.js"></script>
<!-- endbower -->
<!-- endbuild -->
```



# GRUNT

Grunt es, en una palabra,  
**automatización.**

Realiza tareas como minimizar y unificar ficheros, realizar test unitarios, Linting de JavaScript, optimizar imágenes, documentación, despliegue, etc.



```
$ grunt
Running "jshint:gruntfile" (jshint) task
>> 1 file lint free.

Running "jshint:src" (jshint) task
>> 1 file lint free.

Running "jshint:test" (jshint) task
>> 1 file lint free.

Running "qunit:files" (qunit) task
Testing test/tiny-pubsub.html....OK
>> 4 assertions passed (23ms)

Running "clean:files" (clean) task
Cleaning "dist"...OK

Running "concat:dist" (concat) task
File "dist/ba-tiny-pubsub.js" created.

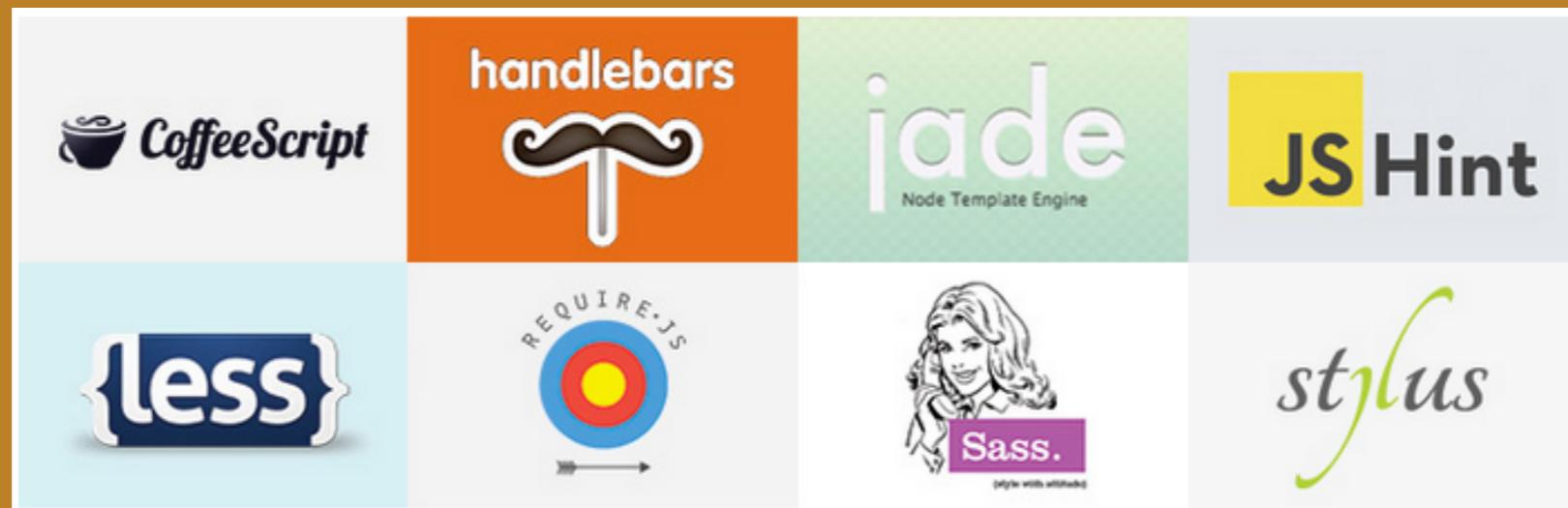
Running "uglify:dist" (uglify) task
File "dist/ba-tiny-pubsub.min.js" created.
Uncompressed size: 389 bytes.
Compressed size: 119 bytes gzipped (185 bytes minified).

Done, without errors.

$ _
```



Existen una gran cantidad de plugins para la realización de tareas concretas.



<http://gruntjs.com/plugins>



# GRUNT

Por defecto, con el generador `webapp`, se incluyen una gran cantidad de plugins y tareas predefinidas.

Autoprefixer

BowerInstall

Concat

CSS min

HTML min

Image min

JSHint

Uglify

Modernizr

...



# GRUNT

La configuración de las tareas se encuentra en el fichero Gruntfile.js.

**Este fichero se divide en:**

- Una función contenedora.
- Configuración de las tareas.
- Carga de plugins externos.
- Tareas personalizadas.



# GRUNT

```
module.exports = function(grunt){  
  grunt.initConfig({  
    uglify: {  
      build: {  
        src: 'app.js',  
        dest: 'build/app.min.js'  
      }  
    },  
    jshint: { all: ['**/*.js'] }  
  });  
  
  grunt.loadNpmTasks('grunt-contrib-uglify');  
  grunt.loadNpmTasks('grunt-contrib-jshint');  
  
  grunt.registerTask('default', ['jshint', 'uglify']);  
};
```



GRUNT

grunt build



# GRUNT

Done, without errors.

Execution Time (2014-04-12 21:49:12 UTC)

concurrent:dist 9.6s  48%

autoprefixer:dist 235ms  1%

cssmin:generated 382ms  2%

uglify:generated 6.9s  34%

modernizr:dist 2.6s  13%

Total 20.2s



# GRUNT

El comando **grunt build** toma el contenido del directorio **app**, y lo convierte en una aplicación.

El directorio **dist** contiene lo imprescindible para hacer funcionar la aplicación.



# GRUNT

```
dist/
├── fonts
│   ├── glyphicon-halflings-regular.eot
│   ├── glyphicon-halflings-regular.svg
│   ├── glyphicon-halflings-regular.ttf
│   └── glyphicon-halflings-regular.woff
├── scripts
│   ├── vendor
│   │   └── d4d78a91.modernizr.js
│   ├── 036656e4.plugins.js
│   ├── b3d37a3c.vendor.js
│   └── b6c3df09.main.js
└── styles
    ├── 71c590b0.vendor.css
    └── a5896f90.main.css
├── 404.html
├── 6df2b309.favicon.ico
└── index.html
└── robots.txt
```

# **D E S A R R O L L O M O D U L A R**

# PROBLEMAS

- Variables globales
- Orden de dependencias
- Sin separación lógica
- Alto acoplamiento

# VARIABLES GLOBALES

```
// app/js/mod1.js  
  
var foo;  
  
foo = function() {  
  console.log("I'm foo");  
}
```

```
// app/js/mod2.js  
  
var foo;  
  
foo = function() {  
  return "I'm foo";  
}
```

?

```
// app/js/app.js  
  
foo();
```

# ORDEN DE DEPENDENCIAS

```
<script src="js/router.js" type="text/javascript"></script>

<script src="js/jquery.min.js" type="text/javascript"></script>

<script src="js/jquery-ui.custom.min.js" type="text/javascript"></script>

<script src="js/bootstrap.min.js" type="text/javascript"></script>

<script src="js/jquery.iceck.min.js" type="text/javascript"></script>

<script src="js/jquery.colorpicker.min.js" type="text/javascript"></script>

<script src="js/jquery.nicescroll.min.js" type="text/javascript"></script>

<script src="js/jquery.fancybox.min.js" type="text/javascript"></script>

<script src="js/select2.min.js" type="text/javascript"></script>

<script src="js/unicorn.js" type="text/javascript"></script>
```

# SEPARACIÓN LÓGICA

```
// app/js/app.js

var APP = APP || {};

APP.data = {
  ...
};

APP.events = {
  ...
};
```

```
// app/js/app.js

APP.controller = {
  ...
};

APP.UI = {
  ...
};
```

# ALTO ACOPLAMIENTO

```
$(document).ready(function(){
    'use strict';

    var $form = $('form');

    $('[data-slug]').on('blur', function(e){
        if(this.value.length === 0) return;

        var $this = $(this),
            id = this.id.replace(/_[a-zA-Z]+$/gi, '_slug'),
            repository = $this.closest('form').data('repository'),
            locale = $this.closest('[data-locale]').data('locale'),
            $target = $('#' + id);

        if ($form.data('action') === 'create' || $target.val().length === 0) {
            $.ajax({
                url: Routing.generate('ajax_slug', {'string': this.value, 'locale': locale, 'repository': repository}),
                dataType: 'json',
                success: function(data, textStatus, jqXHR) {
                    $target.val(data.slug);
                },
                error: function(jqXHR, textStatus, errorThrown) {
                    alert('Se ha producido un error al obtener el slug.');
                }
            });
        }
    });
});
```

# PRINCIPIOS DEL DESARROLLO MODULAR

- Capa componente de la aplicación, debe ocuparse de su cometido.
- Separar responsabilidades reduce la complejidad.



# PATRONES MODULARES

# IIFE

IMMEDIATELY INVOKED FUNCTION EXPRESSION

```
(function($, exports, undefined){  
    exports.APP.data = {  
        ...  
    };  
  
    exports.APP.events = {  
        ...  
    };  
  
    exports.APP.controller = {  
        ...  
    };  
})(jQuery, window);
```

# PATRÓN MÓDULO

```
var snacksModule = (function(){
    var snacks = [],
        nextId = 0,
        Snack = function(options) { /* new snack */ };

    var add = function(options) {};
    var get = function(index) {};
    var update = function(index, props) {};
    var delete = function(index) {};

    return {
        add : add,
        get : get,
        update : update,
        delete : delete
    };
})();
```

# AMD

## ASYNCHRONOUS MODULE DEFINITION

```
define(  
    module_id, /* optional */  
    [dependencies], /* optional */  
    factory  
);
```

# AMD

## ASYNCHRONOUS MODULE DEFINITION

```
define(  
  'user',  
  ['data', 'ajax'],  
  function(data, ajax) {  
    // do something  
  
    // export public API  
    return {  
      signin : function(){},  
      getName : function(){},  
      setName : function(){},  
    };  
  }  
);
```

# AMD

## ASYNCHRONOUS MODULE DEFINITION

```
define(  
  'app',  
  [ 'user' ],  
  function(user) {  
    // do something  
    user.signin('username', 'secretpass');  
  }  
);
```

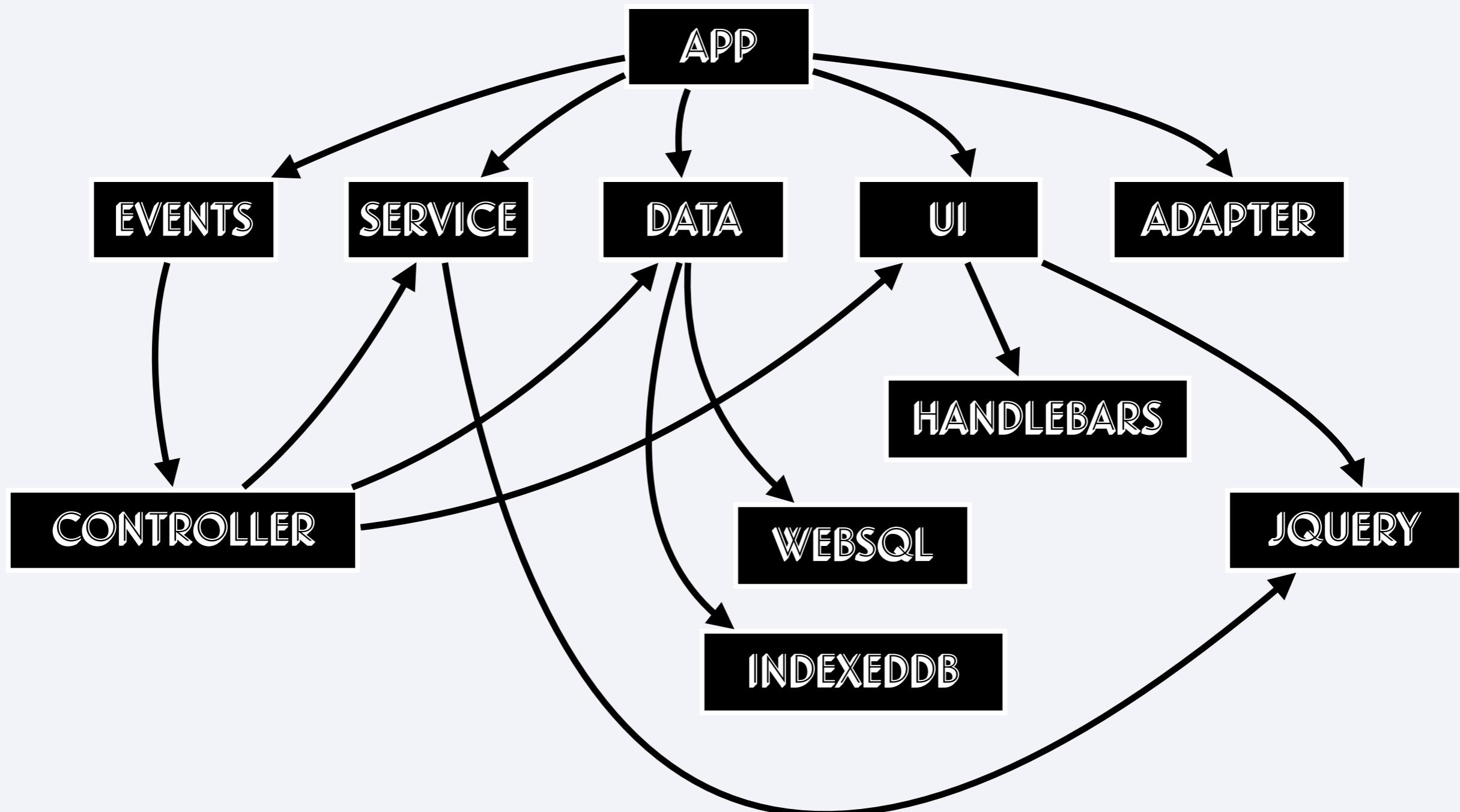


Es un cargador de **ficheros** y **módulos** de  
JavaScript.

Compatible con la especificación **AMD**.

Resuelve **dependencias circulares**.

Admite **plugins**.





## Cómo comenzar a utilizar RequireJS:

```
<script data-main="js/config" src="js/require.js"></script>
```



## Definir módulos:

```
define('controller', ['service', 'data', 'ui', 'quo'], function(Service, DB, UI, $) {
    'use strict';

    console.log('[Controller module initialized]');

    var filterEarthQuakes = function(e) {
        var $this = $(this);
        var filter = JSON.parse(sessionStorage.getItem('filter')) || {};

        filter.name = this.value;
        DB.searchEarthQuakes(filter);
    };

    var showDetail = function(e) {
        var $this = $(this),
            id = this.dataset.id;

        DB.getEarthQuake(id, function(earthquake) {
            UI.showDetail(earthquake);
        });
    };

    var showMap = function(e) {
        UI.showMap(DB.getLatestData());
    };

    return {
        showDetail : showDetail,
        showMap : showMap,
        filterEarthQuakes : filterEarthQuakes
    };
});
```



## Definir módulos:

```
define('service', ['jquery'], function($) {
  'use strict';

  console.log('[Services module initialized]');

  var updateEarthQuakes = function(success) {
    $.getJSON('data/all_day.geojson', null, function(data) {
      success(data.features);
    });
  };

  return {
    updateEarthQuakes : updateEarthQuakes
  };
});
```



## Definir módulos:

```
define('data', ['WebSQL', 'IndexedDB', 'underscore'], function(WebSQL, IndexedDB, _) {
  'use strict';

  console.log('[Data module initialized]');

  var _subscribers = [],
      _latestData = [],
      _lastFilter = {},
      _config,
      _DB;

  var CONFIG_BASE = {
    name : 'earthquake',
    version : '1.0',
    desc : 'EarthQuakes database',
    size : 2 * 1024 * 1024,
    limit : 50
  };

  var init = function(config) {
    _config = _.extend(CONFIG_BASE, config);
    _lastFilter.limit = _config.limit;

    if(Modernizr.websqldatabase) {
      _DB = new WebSQL(config);
    } else {
      _DB = new IndexedDB(config);
    }

    // Show actual earthquakes
    _prepareData(_lastFilter);
  };
});
```



## Configuración: **baseUrl**

Por defecto, RequireJS asigna a **baseURL** el **mismo directorio** donde se encuentra el **HTML** que carga **require.js**

Si el atributo **data-main** ha sido definido, su **path** será el valor que tome **baseUrl**.



Configuración: **baseUrl**

```
require.config({  
  baseUrl : 'app/js'  
});
```



## Configuración: **paths**

Rutas relativas/absolutas a módulos que no se encuentran bajo **baseUrl**.

Podemos indicar rutas completas a dominios externos, en incluso incluir un **fallback**.

En las rutas **NO** se incluye la extensión del fichero.



## Configuración: paths

```
require.config({
  baseUrl : 'app/js',
  paths: {
    jquery: [
      'http://ajax.googleapis.com/ajax/libs/jquery/2.1.0/jquery.min',
      '../vendor/jquery',
    ],
    'jquery.gmap3': '../vendor/jquery.gmap3',
    quo : '../vendor/quo.debug',
    lungo : '../vendor/lungo',
    handlebars: '../vendor/handlebars',
    underscore: '../vendor/underscore',
    async: '../vendor/async',
  }
});
```



## Configuración: **shim**

Cuando los módulos o **librerías** externas no son compatibles con AMD, sigue siendo **possible** incluirlas en nuestro proyecto.

Hay que tener en cuenta el **nombre** de la **variable global** que crea la librería.



## Configuración: paths

```
require.config({
  shim: {
    jquery: {
      exports: 'jQuery'
    },
    quo: {
      exports: '$$'
    },
    lungo : {
      deps : [
        'quo'
      ],
      exports : 'Lungo'
    },
    underscore: {
      exports: '_'
    }
  }
});
```

handlebars



# handlebars

**Librería para utilizar plantillas en JavaScript**

Queremos separar la lógica de la presentación.  
Crear o actualizar la vista con los datos del modelo.

<http://handlebarsjs.com/>

# handlebars

```
var $modules = $('div.module').hide();
var $nav = $('<ul/>', {
  'id' : 'myTabs'
}).addClass('tabs');

var lis = [];
$modules.each(function() {
  var $module = $(this);
  var $title = $module.find('h2').first().text();

  var $li = $('<li/>', {
    'text' : $title
  });
  $li.data('target', $module);

  lis.push($li.get(0));
});

$nav.append(lis).insertBefore($modules.eq(0));
```



NOPE

# handlebars

- 1. Obtener la plantilla, desde al HTML o a través de AJAX (síncrono).**
- 2. Compilar la plantilla.**
- 3. “Ejecutar” con los datos del contexto.**
- 4. Insertar el HTML resultante en la vista.**

# handlebars

## 1. Obtener la plantilla.

IDENTIFICADOR

TIPO

```
<script id="list-tpl" type="text/x-handlebars-template">
{{#each earthquakes}}
  <li class="anchor">{{date}}</li>
{{#each earthquakes}}
  <li class="arrow selectable thumb big" data-id="{{_id}}>
    <span class="magnitude">{{mag}}</span>
    <div>
      <strong>{{title}}</strong>
      <small>{{formatTime time}}</small>
    </div>
  </li>
{{/each}}
{{/each}}
</script>
```

# handlebars

## 1. Obtener la plantilla.

CÓDIGO EN “BRUTO”

```
var list = $("#list-tpl").html();
```

# handlebars

## Sintaxis básica

```
<h1>{{title}}</h1>
```

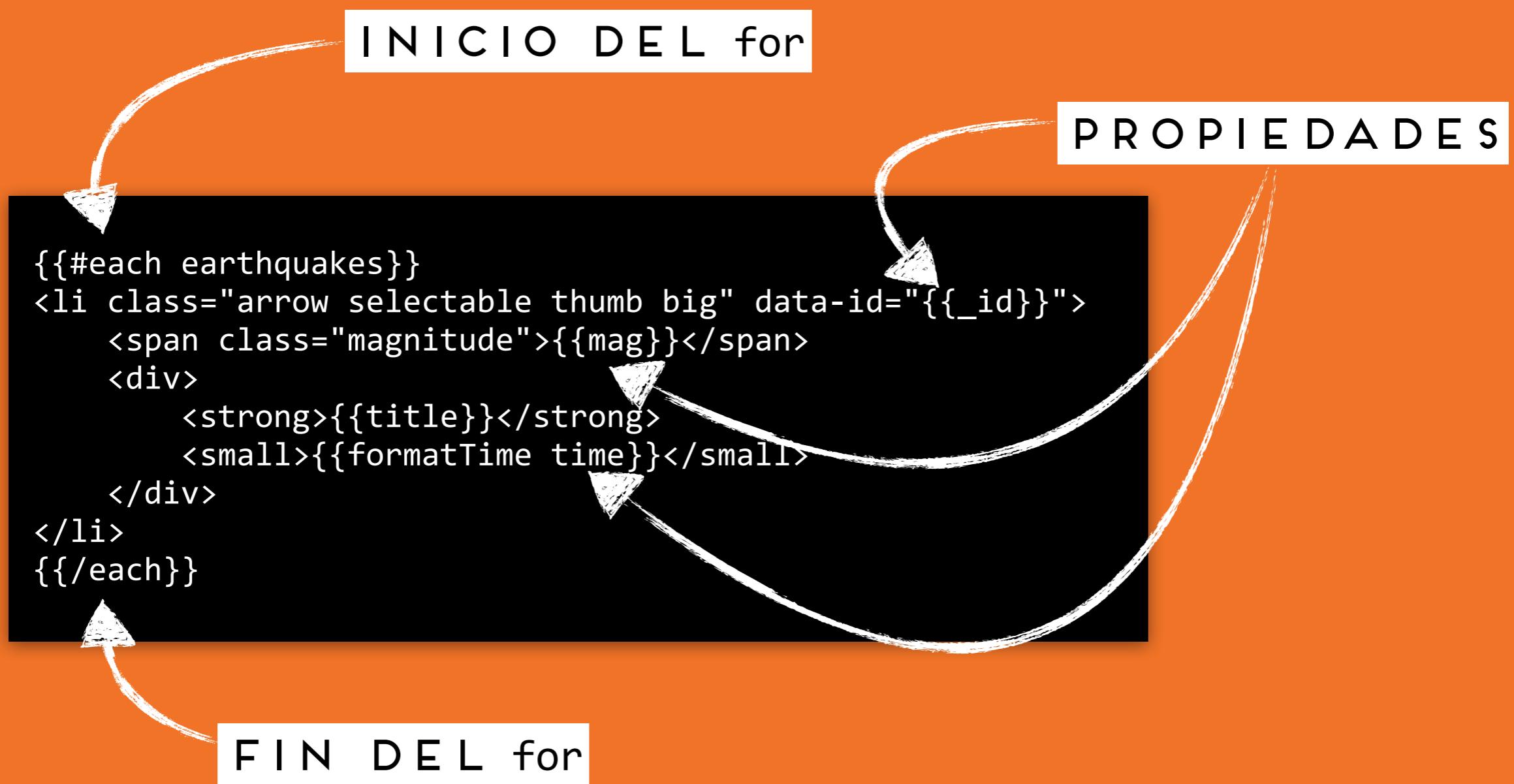
VARIABLE title

VARIABLE article.title

```
<h1>{{article.title}}</h1>
```

# handlebars

## Sintaxis básica



# handlebars

## 2. Compilar la plantilla.

```
var list = $("#list-tpl").html();  
var template = Handlebars.compile(list);
```



FUNCIÓN EJECUTABLE

# handlebars

## 3. “Ejecutar” con los datos del contexto.

```
var data = [
  { _id : 1, title : 'Earthquake', date : '2014-04-05', mag : 3.8 },
  { _id : 2, title : 'Earthquake', date : '2014-04-06', mag : 5.4 },
  { _id : 3, title : 'Earthquake', date : '2014-04-07', mag : 2.2 }
];

var html = template(data);
```



HTML FINAL

# handlebars



YEAH

```
<li class="arrow selectable thumb big" data-id="1">
  <span class="magnitude">3.8</span>
  <div>
    <strong>Earthquake</strong>
    <small>April 5th, 2014</small>
  </div>
</li>
<li class="arrow selectable thumb big" data-id="2">
  <span class="magnitude">5.4</span>
  <div>
    <strong>Earthquake</strong>
    <small>April 6th, 2014</small>
  </div>
</li>
<li class="arrow selectable thumb big" data-id="3">
  <span class="magnitude">2.2</span>
  <div>
    <strong>Earthquake</strong>
    <small>April 7th, 2014</small>
  </div>
</li>
```

# handlebars

## 4. Insertar el HTML resultante en la vista.

```
var html = template(data);
$("ul#earthquake-list").html(html);
```

# handlebars

## Expresiones

```
 {{#each expression}}  
   <tag>{{property}}</tag>  
 {{/each}}
```

Itera sobre cada elemento de la expresión, y genera la plantilla asociada.

El elemento sobre el que itera pasa a ser el nuevo contexto.

# handlebars

## Expresiones

`{{@index}}`

Número de iteración actual.

`{{@key}}`

Nombre de la propiedad actual.

`{{@first}}`

Indica si es la primera iteración.

`{{@last}}`

Indica si es la última iteración.

# handlebars

## Expresiones

```
{{{#if author}}
  <h1>{{firstName}} {{lastName}}</h1>
{{else}}
  <h1>Unknown Author</h1>
{{/if}}}
```

Renderiza el primer bloque si se cumple la condición (no es `false`, `undefined`, `null`, “” o `[ ]`).

Bloque `else` en caso de no cumplirse.

# handlebars

## Expresiones

```
{{#unless license}}
  <h3>This entry does not have a license!</h3>
{{/unless}}
```

Caso contrario al **if**, se renderiza el bloque si la condición es falsa.

# handlebars

## Helpers

Permiten crear funciones que modifiquen los datos desde la propia plantilla.

Se registran a través de  
`Handlebars.registerHelper()`.

Los helpers reciben el contexto actual en `this`.

# handlebars

## Helpers

```
Handlebars.registerHelper('formatTime', function(time) {  
  var dateTime = new Date(time);  
  return dateTime.toTimeString()  
    .replace(/.*(\d{2}:\d{2}:\d{2}).*/ , '$1');  
});
```

```
{{#each earthquakes}}  
<li class="arrow selectable thumb big" data-id="{{_id}}>  
  <small>{{formatTime time}}</small>  
</li>  

```

# handlebars

## **Handlebars + RequireJS**

Las plantillas de Handlebars pueden ser utilizadas como un módulo especial de RequireJS.

Podemos separar las plantillas en ficheros.

Obtenemos la plantilla como una dependencia del módulo.

# handlebars

## Handlebars + RequireJS

```
define("UI", ["module", "text!tpl/module.html"],  
       function(module, html){  
           //the html variable will be the text  
           //of the some/module.html file  
       }  
);
```

PLUGIN TEXT

RUTA A LA PLANTILLA

# handlebars

## **Handlebars + RequireJS**

```
require.config({  
    baseUrl: "app/scripts",  
    paths: {  
        "tpl": "../tpl"  
    }  
});
```

# **FRONTEND TESTING**

# PROBLEMAS

**Necesitamos probar nuestros desarrollos en frontend, y eso supone:**

Comprobar la lógica y el comportamiento.  
Enfrentarse a asincronismo y funciones de tiempo.  
Particularidades de cada navegador.

Diferentes tecnologías de testing.

# TESTING



+



+





mocha

- NodeJS y navegador.
- Tests asíncronos.
- % de cobertura.
- Identifica ejecuciones lentas.
- Permite utilizar cualquier librería de aserciones.
- Eventos para ejecutar tareas antes y después de los tests.

<http://visionmedia.github.io/mocha/>



- BDD + TDD
- NodeJS y navegador.
- Tres tipos de aserciones:
  - ✓ should
  - ✓ expect
  - ✓ assert

<http://chaijs.com/>

# Sinon.JS

Librería que nos permite espiar, modificar y  
engañar en las llamadas a métodos de  
JavaScript.

<http://sinonjs.org/>



+



```
// Hello [name]!
var hello = function(name) {
    return "Hello " + name + "!";
};
```



+



```
describe("Hello testing", function(){
  it("should say hello to you", function(){
    var resp = hello("Arkaitz");
    resp.should.be.a('string');
    resp.should.equal('Hello Arkaitz!');
    resp.have.length(14);
  })
});
```



+



```
describe("Hello testing", function(){
  it("should say hello to you", function(){
    var resp = hello("Arkaitz");
    expect(resp).to.be.a('string');
    expect(resp).to.equal('Hello Arkaitz!');
    expect(resp).to.have.length(14);
  })
});
```



+



```
describe("Hello testing", function(){
  it("should say hello to you", function(){
    var resp = hello("Arkaitz");
    assert.typeOf(resp, 'string');
    assert.equal(resp, 'Hello Arkaitz!');
    assert.lengthOf(resp, 14);
  })
});
```



+



**Al trabajar con RequireJS, necesitamos cargar los módulos antes de ejecutar los tests.**

```
<script src="bower_components/requirejs/require.js">
</script>
```

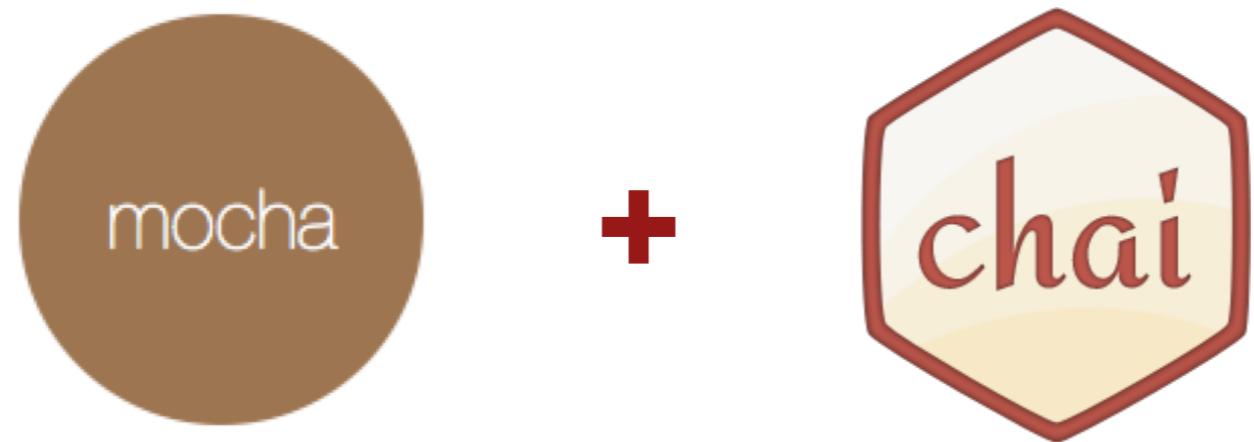
```
// tests.js
require.config({
  baseUrl: '../path/to/scripts',
  nodeRequire: require
});
```



```
describe('Test', function () {
  var mod;

  beforeEach(function(done){
    require(['module'], function(module){
      mod = module;
      done();
    });
  });

  describe('Test methods', function () {
    it('Should return 1', function () {
      assert.equal(1, mod.test(1));
    });
  });
});
```



Mocha Spec Runner - Google Chrome

Mocha Spec Runner localhost/test/

passes: 1 failures: 2 duration: 0.07s 100%

## FizzBuzz

### FizzBuzz main method

- ✓ Should return n when the number is not multiple of 3 or 5
- ✗ Should return Fizz when the number is multiple of 3
  - AssertionError: expected 'Fizz' to equal 3
- ✗ Should return Buzz when the number is multiple of 5
  - AssertionError: expected 'Buzz' to equal 5

# Sinon.JS

## **Spy**

Crea una función con el mismo comportamiento que la original, pero añade la funcionalidad de “espionaje”.

Identifica los parámetros de la llamada, el valor de retorno, el valor de `this` y el lanzamiento de excepciones.

# Sinon.JS

## Spy

```
var callback = sinon.spy();
PubSub.subscribe("message", callback);

PubSub.publishSync("message");

assertTrue(callback.called);
```

# Sinon.JS

## Spy

```
var spy = sinon.spy(subscribe);
subscribe("message");

assertTrue(callback.called);
```

# Sinon.JS

## Spy

```
var spy = sinon.spy(PubSub, 'subscribe');  
PubSub.subscribe("message");  
  
assertTrue(callback.called);
```



+



+

Sinon.JS

## Spy

```
describe('Test', function () {
  describe('Test methods', function () {
    it('Spies', function () {
      var callback = sinon.spy();
      callMethod('param', callback);
      assert.isTrue(callback.called);
    });
  });
});
```



+



+

Sinon.JS

## Spy

```
describe('Test', function () {
  beforeEach(function(){
    var spy = sinon.spy(method);
  });

  describe('Test methods', function () {
    it('Spies', function () {
      expect(spy.callCount).to.equal(0);
      expect(method()).to.equal('something');
      expect(spy.callCount).to.equal(1);
    });
  });
});
```



+



+

Sinon.JS

## Spy

```
describe('Test', function () {
  beforeEach(function(){
    sinon.spy($, 'ajax');
  });

  afterEach(function(){
    $.ajax.restore();
  });

  describe('Ajax Spy', function (done) {
    it('Spies', function () {
      $.ajax({'http://domain/path', success : function(){}});
      expect($.ajax.calledOnce).to.be.true;
      done();
    });
  });
});
```

# Sinon.JS

## **Stub**

Crea una función que modifica el comportamiento original de la función.

Útil para simular tratamiento de excepciones o simular respuestas AJAX.



+



+

Sinon.JS

## Stub

```
beforeEach(function() {  
  sinon.stub($, 'ajax').yieldsTo('success', {  
    count: '100',  
    message: 'Object from Spy!'  
  });  
});  
  
// Restore jQuery's ajax method to its  
// original state  
afterEach(function() {  
  $.ajax.restore();  
})
```



+



+

**Sinon.JS**

## Stub

```
it('should make an ajax call', function(done) {  
  object.makeAjax();  
  expect($.ajax.calledOnce).to.be.true;  
  done();  
});  
  
it('should update the count', function(done) {  
  object.makeAjax(function(data){  
    expect(data).to.equal(100);  
    done();  
  });  
});
```

# Sinon.JS

## Fake Timers

Es una implementación síncrona del método `setTimeout`.

Nos permite simular retardos en la ejecución de las funciones.

# Sinon.JS

## Fake Timers

```
var clock = sinon.useFakeTimers();  
  
var hidden =  
  $("<div class='hidden">Text</div>")  
    .appendTo(document.body).fadeIn("slow");  
  
clock.tick(650);          // slow = 600ms  
hidden.css("opacity") === 1; // true  
  
clock.restore();
```



+



+

Sinon.JS

## Fake Timers

```
describe('Test', function () {
  beforeEach(function(){
    clock = sinon.useFakeTimers();
  });

  afterEach(function(){
    clock.restore();
  });

  describe('Fake timer', function () {
    it('Clock', function () {
      var $hidden = $("<div class='hidden'>Text</div>")
        .appendTo(document.body).fadeIn("slow");

      clock.tick(650);
      expect($hidden.css("opacity")).to.be.equal(1);
    });
  });
});
```

# Sinon.JS

## Fake Server

Permite crear un servidor ficticio que responda a peticiones web.

# Sinon.JS

## Fake Server

```
var server = sinon.fakeServer.create();

server.respondWith("GET", "/twitter/api/user.json", [
  200,
  {"Content-Type": "application/json"},  

  '[{"id": 0, "tweet": "Hello World"}]'
]);

$.get("/twitter/api/user.json", function (data) {
  console.log(data); // [{"id":0,"tweet":"Hello World"}]
});
server.respond();

server.restore();
```



Sinon.JS

## Fake Server

```
describe("getTweets - Server", function () {
  var server, fakeData = [ /* ... */ ];

  before(function () {
    server = sinon.fakeServer.create();
    server.respondWith(
      "GET",
      "/data/tweets.json",
      [200, { "Content-Type": "application/json" }, JSON.stringify(fakeData)]
    );
  });

  it("should $.ajax and invoke callback", function (done) {
    twitter.getTweets("arkaitzgarro", function (tweets) {
      expect(tweets.length).to.be(5);
      done();
    });
    server.respond();
  });

  after(function () { server.restore(); });
});
```