

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

**El tiempo es un factor clave a la
hora de ser**

PRODUCTIVOS



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.

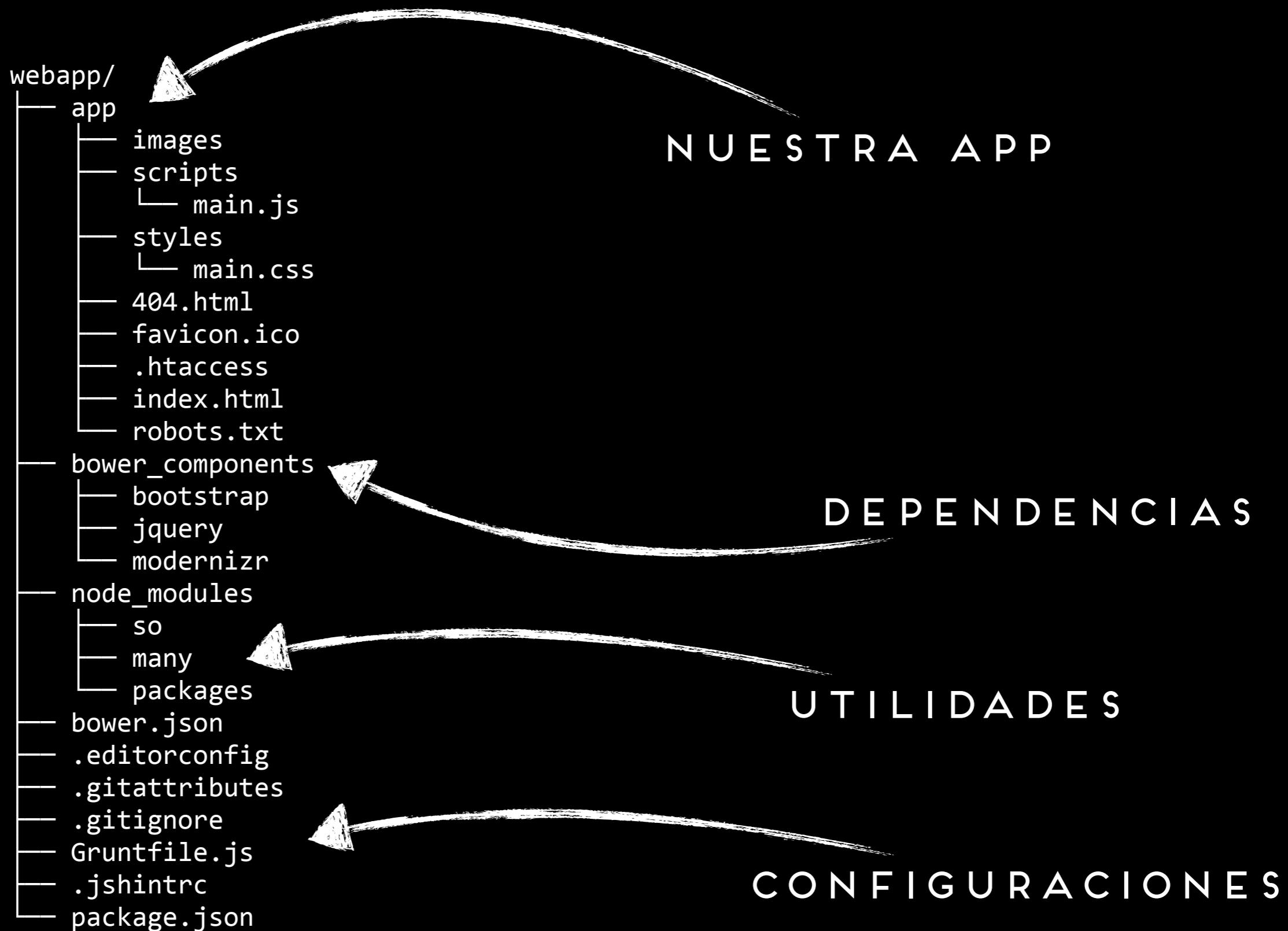
```
mkdir my-yo-project  
cd my-yo-project
```

```
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

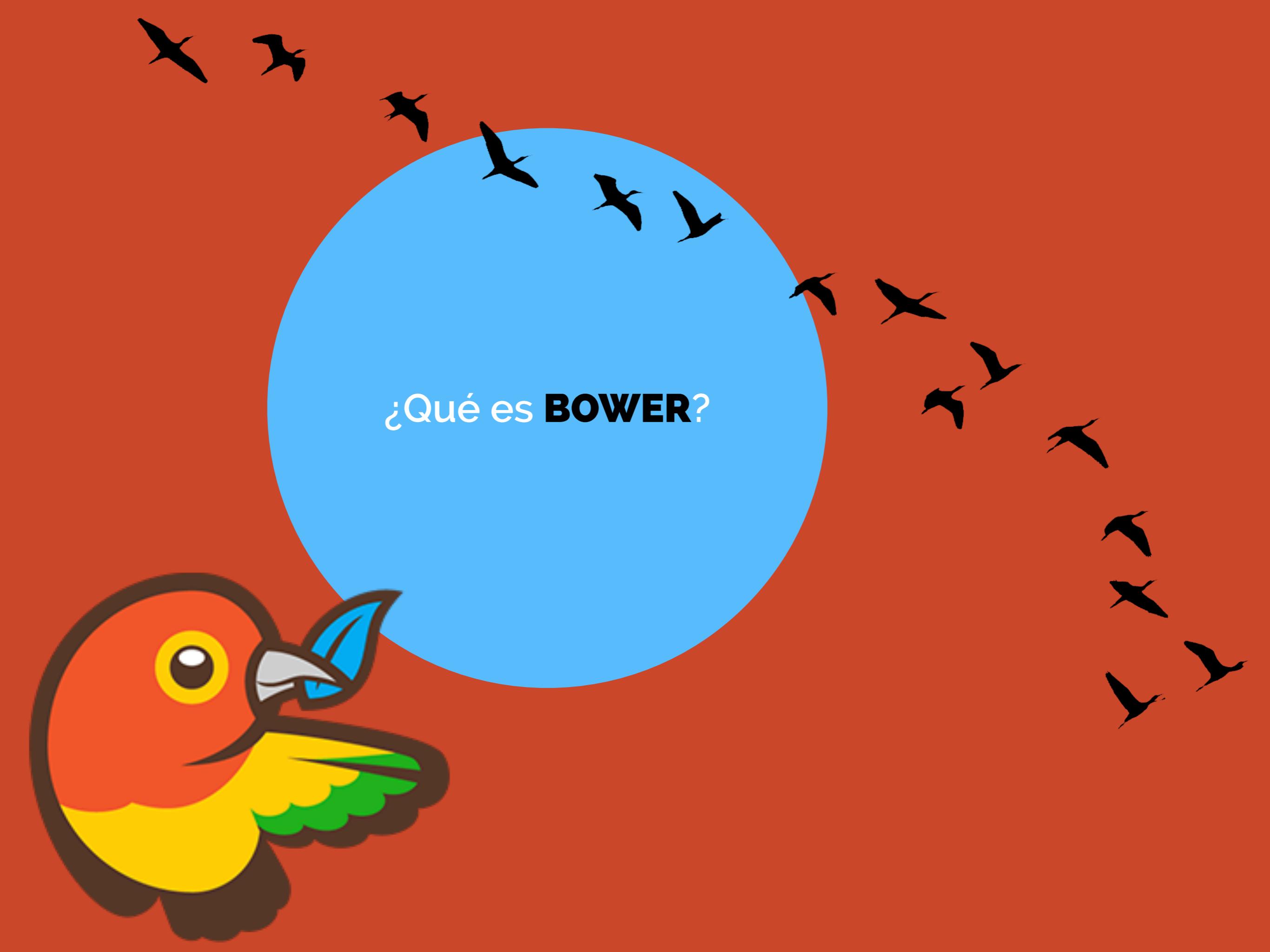
```
npm install -g generator-angular
```

```
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
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 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.

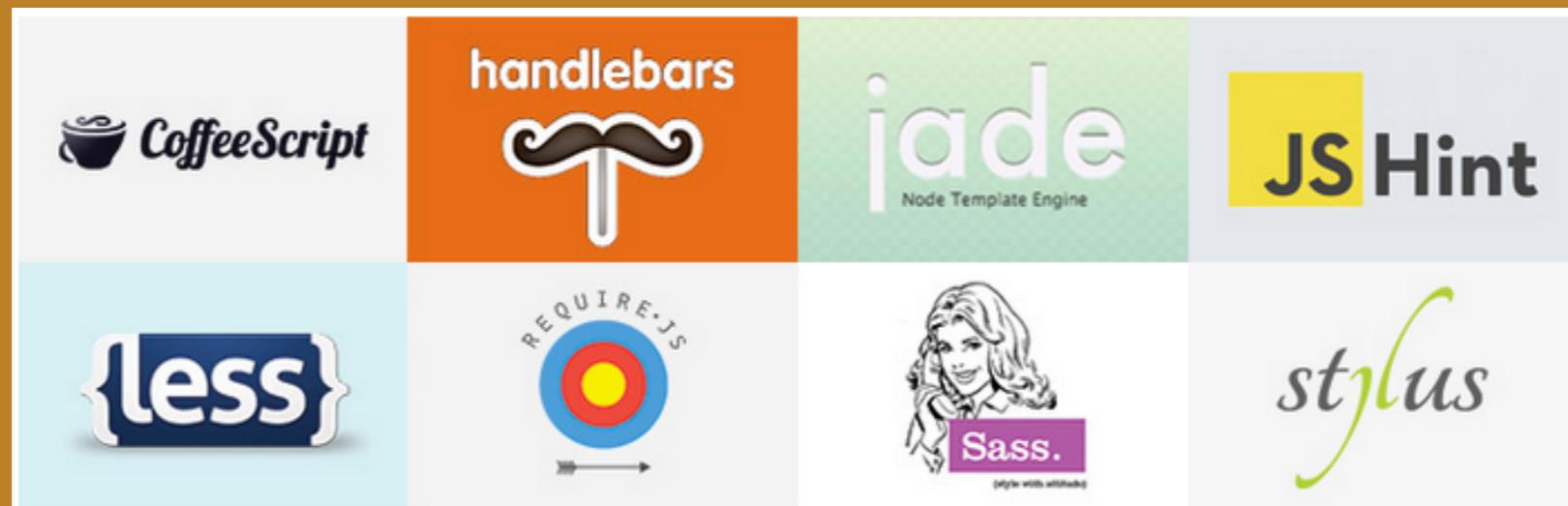
Running "uglify:dist" (uglify) task
File "dist/ba-tiny-pubsub.min.js" created.
Uncompressed size: 389 bytes.
Compressed size: 119 bytes gzipped (185 bytes minified).

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.ivecheck.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');  
  }  
);
```

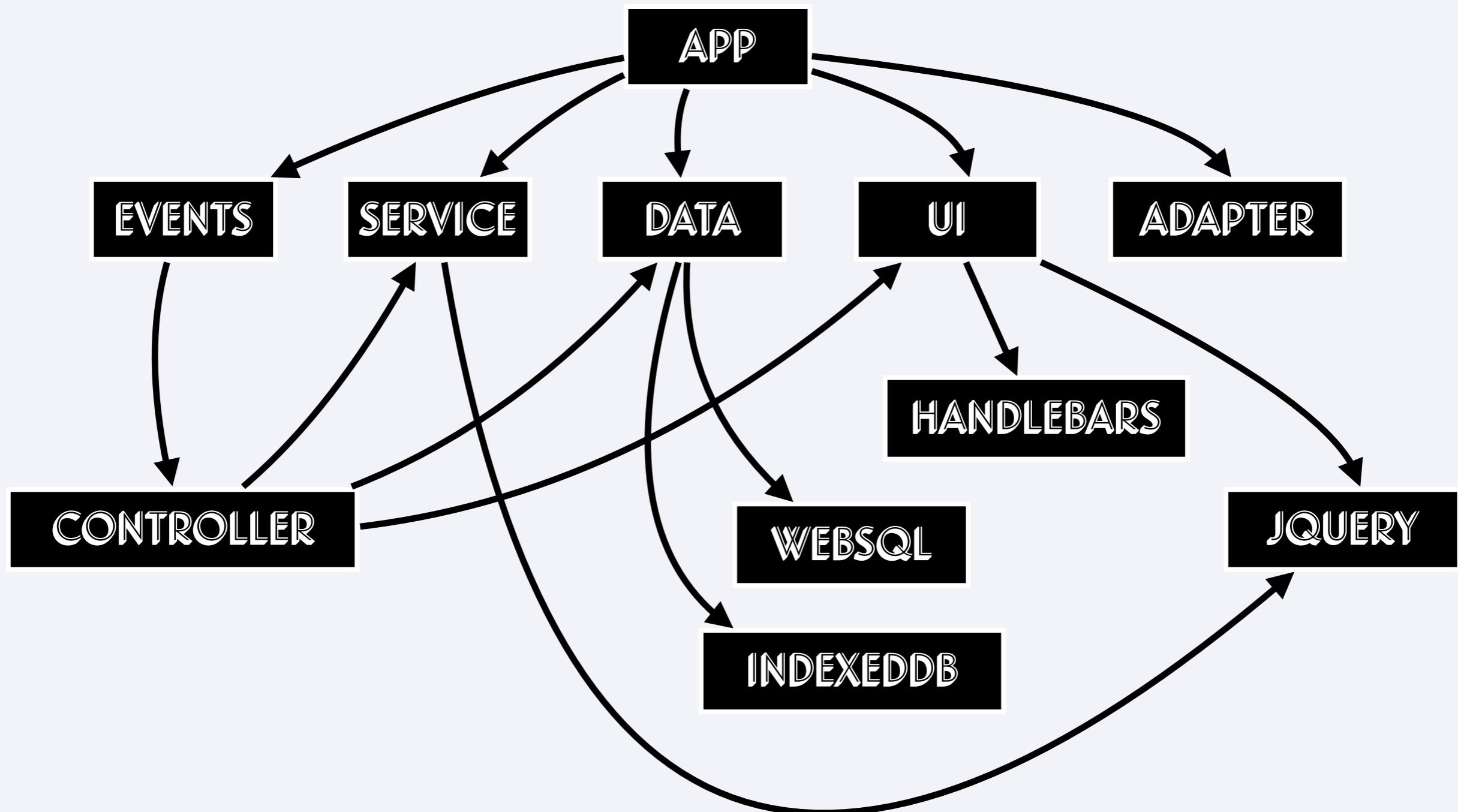


Es un cargador de **ficheros** y **módulos** de
JavaScript.

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: '_'
    }
  }
});
```