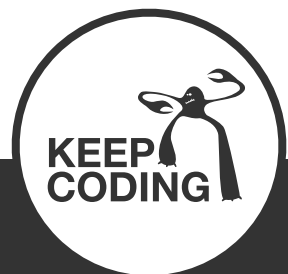




Grunt



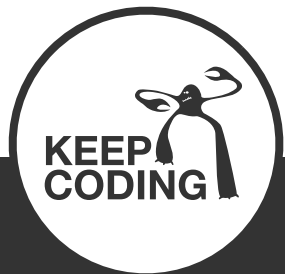
# Introducción



# ■ GRUNT: The JavaScript TaskRunner

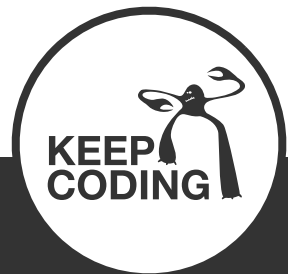


Grunt permite automatizar diferentes tipos de tareas típicas del desarrollo.





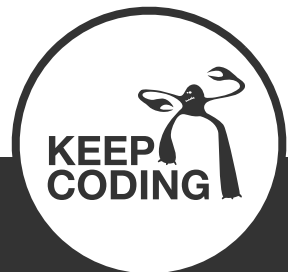
# Instalando Grunt



# ■ Instalando la Command Line Interface de Grunt

Para instalar la CLI de Grunt, usaremos NPM:

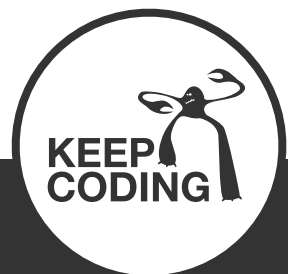
```
$ sudo npm install -g grunt-cli
```





Además de Grunt, necesitamos instalar los plugins para automatizar las tareas que queramos.

Lo más mejor es crear un archivo `packages.json` donde pondremos la información de dependencias (plugins) a instalar.





# ■ package.json

```
{  
  "name": "my-project",  
  "version": "0.1.0",  
  "devDependencies": {  
    "grunt": "~0.4.5",  
    "grunt-contrib-connect": "1.0.1",  
    "load-grunt-tasks": "3.4.1",  
    "time-grunt": "1.3.0"  
  }  
}
```



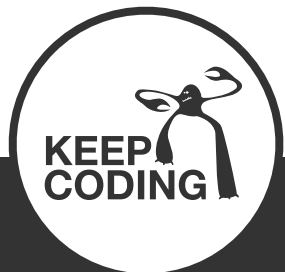


# ■ Instalando las dependencias

Con nuestro archivo package.json, debemos instalar las dependencias usando NPM:

```
$ npm install
```

Esto nos creará un directorio node\_modules en nuestro proyecto con los módulos indicados en el package.json

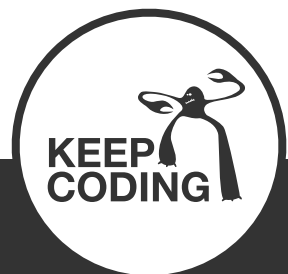


# ■ Configurando Grunt

Para configurar Grunt, debemos crear en nuestro proyecto un archivo **Gruntfile.js** con un diccionario de configuración de Grunt.

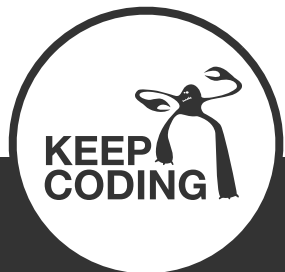


# Configurando Grunt



# ■ Gruntfile.js

```
module.exports = function (grunt) {  
  
    // Load grunt tasks automatically  
    require('load-grunt-tasks')(grunt);  
  
    // Time how long tasks take.  
    require('time-grunt')(grunt);  
  
    var appConfig = {...}  
  
    grunt.initConfig({  
        appConfig: appConfig,  
        connect: {  
            options: {  
                port: 9000,  
                hostname: '0.0.0.0',  
                livereload: true  
            },  
            server: {  
                options: {  
                    open: true,  
                    base: '.'  
                }  
            }  
        }  
    });  
  
    // default task  
    grunt.registerTask('default', ['connect']);  
};
```

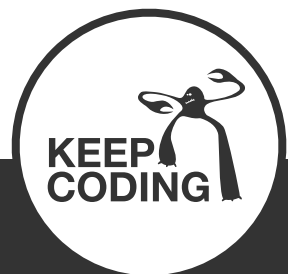


# ■ Gruntfile.js

```
var appConfig = {  
  env: env,  
  app: require('./bower.json').appPath || 'app',  
  dist: 'dist',  
  // Change this to '0.0.0.0' to access the server from outside.  
  hostname: '0.0.0.0',  
  ports: {  
    server: grunt.option('serverPort') || 9000,  
    liveReload: grunt.option('liveReloadPort') || 35729  
  }  
};
```



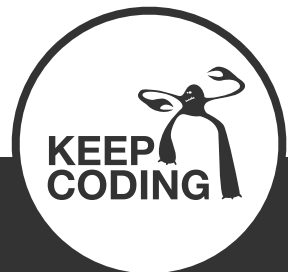
# Ejecutando Grunt



# ■ Ejecutando grunt

Ya tenemos todo listo, ahora tan sólo tenemos que ejecutar Grunt desde el path del proyecto:

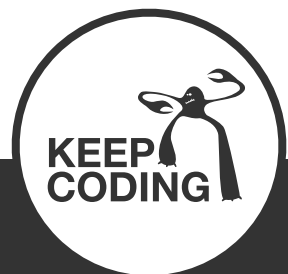
```
$ grunt
```





# Plugins

<http://gruntjs.com/plugins>



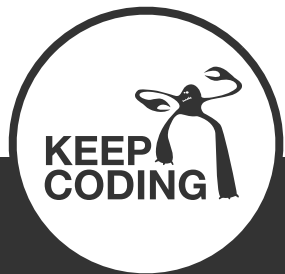
# ■ load-grunt-tasks

Permite cargar de forma automática todos los plugins de grunt oficiales (“grunt-\*”).



# ■ load-grunt-tasks

```
// Load grunt tasks automatically  
require('load-grunt-tasks')(grunt);
```



# ■ grunt-contrib-connect

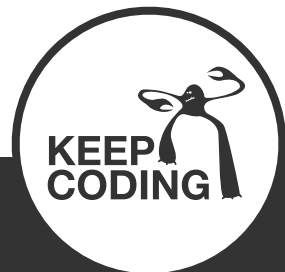
Nos permite levantar un servidor web en local, con soporte “livereload”.

Permite servir multiples directorios en una sola ruta.



# ■ grunt-contrib-connect

```
connect: {
  options: {
    port: appConfig.ports.server,
    hostname: appConfig.hostname,
    livereload: appConfig.ports.liveReload
  },
  livereload: {
    options: {
      open: true,
      base: ['.tmp', '.', 'app']
    }
  },
  dist: {
    options: {
      open: true,
      base: ['.tmp', 'dist']
    }
  }
},
```

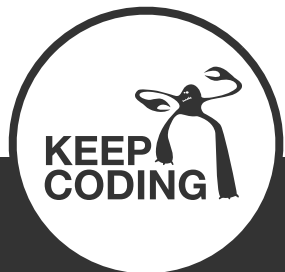


# ■ grunt-contrib-watch



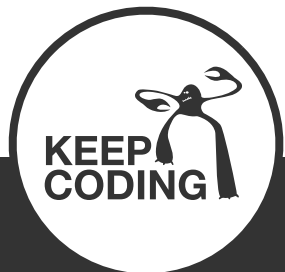
Nos permite configurar acciones a realizar cuando cambian archivos que tienen cierto patrón.

Sirve para compilar archivos LESS o SASS cuando se modifican o para concatenar y ofuscar archivos Javascript.



# ■ grunt-contrib-watch

```
watch: {
  sass: {
    files: [
      '<%= appConfig.app %>/styles/**/*.{scss,sass}'
    ],
    tasks: ['sass', 'autoprefixer']
  },
  livereload: {
    options: {
      livereload: '<%= connect.options.livereload %>'
    },
    files: [
      '.tmp/*.html',
      '<%= appConfig.app %>/**/*.html',
      '.tmp/styles/{,*/}*.css',
      '<%= appConfig.app %>/images/{,*/}*.{png,jpg,jpeg,gif,webp,svg}'
    ]
  }
},
```

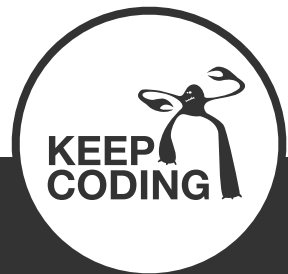




# ■ grunt-contrib-wiredep

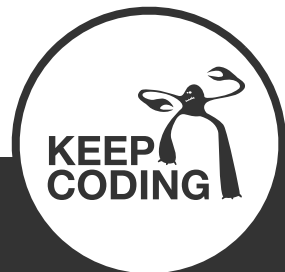


Nos permite enlazar dependencias bower en nuestro código de forma automática.



# ■ grunt-contrib-wiredep

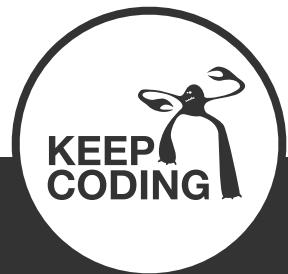
```
wiredep: {
  app: {
    src: [
      '<%= appConfig.app %>/index.html'
    ],
    ignorePath: /\.\.\\/,
  },
  sass: {
    src: ['<%= appConfig.app %>/styles/{,*/}*.{scss,sass}']
  }
},
```



# ■ grunt-include-source

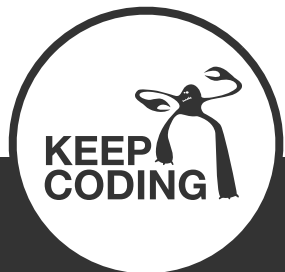


Nos permite enlazar recursos en nuestro código de forma automática.



# ■ grunt-include-source

```
includeSource: {
  options: {
    basePath: ['<%= appConfig.app %>', '.tmp'],
    baseUrl: '',
  },
  server: {
    files: {
      '.tmp/index.html': '<%= appConfig.app %>/index.html'
    }
  },
  dist: {
    files: {
      '<%= appConfig.dist %>/index.html': '<%= appConfig.app %>/index.html'
    }
  }
},
```

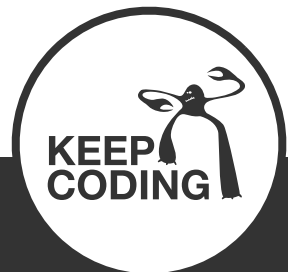


# ■ grunt-contrib-jshint



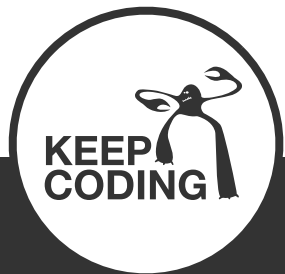
Analiza código Javascript y da feedback de errores y malas prácticas.

<http://jshint.com>



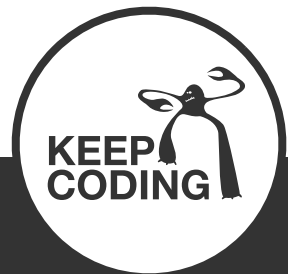
# ■ grunt-contrib-jshint

```
jshint: {
  options: {
    jshintrc: '.jshintrc',
    reporter: require('jshint-stylish')
  },
  all: {
    src: [
      'Gruntfile.js',
      '<%= appConfig.app %>/{,**/}*.*js'
    ]
  }
},
```



# ■ grunt-contrib-sass

Compila archivos SASS a CSS





# ■ grunt-contrib-sass

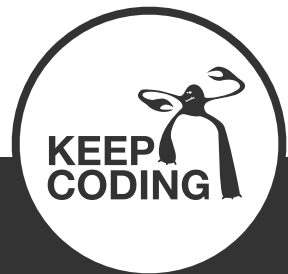
```
sass: {  
  options: {  
    sourceMap: true  
  },  
  all: {  
    files: [{  
      expand: true,  
      cwd: 'app',  
      src: ['styles/main.scss'],  
      dest: '.tmp',  
      ext: '.css'  
    }]  
  }  
},
```



# ■ grunt-contrib-copy

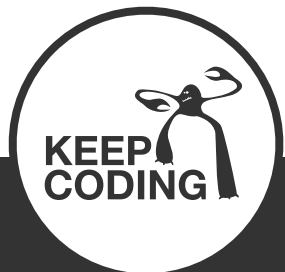


Copia ficheros y directorios



# ■ grunt-contrib-copy

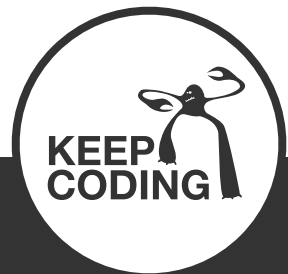
```
copy: {
  dist: {
    files: [{
      expand: true,
      dot: true,
      cwd: '<%= appConfig.app %>',
      dest: '<%= appConfig.dist %>',
      src: [
        '*.ico,png,txt',
        '.htaccess',
        '.htpasswd',
        '*.html',
        'images/{,*/}* ',
        'fonts/*',
        '**/*.js'
      ]
    }, {
      expand: true,
      cwd: '.tmp/images',
      dest: '<%= appConfig.dist %>/images',
      src: ['generated/*']
    }
  ]
},
```



# ■ grunt-contrib-clean

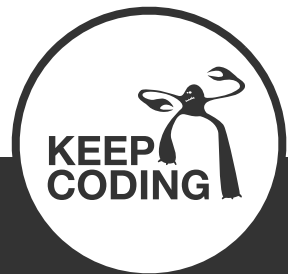


Elimina ficheros y directorios



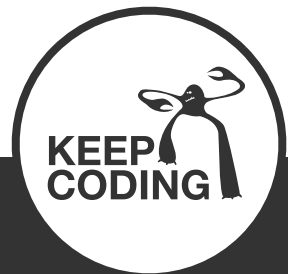
# ■ grunt-contrib-clean

```
clean: {
  dist: {
    files: [{
      dot: true,
      src: [
        '.sass-cache',
        '.tmp',
        '<%= appConfig.dist %>/{,*/}*'
      ]
    }]
  }
},
```



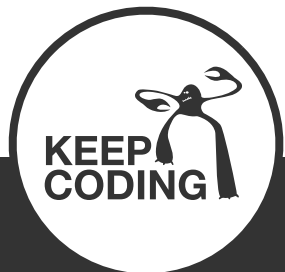
# ■ grunt-autoprefixer

Pone automáticamente prefijos a instrucciones CSS



# ■ grunt-autoprefixer

```
autoprefixer: {  
  options: {  
    browsers: ['last 1 version']  
  },  
  dist: {  
    files: [{  
      expand: true,  
      cwd: '.tmp/styles/',  
      src: '**/*.css',  
      dest: '.tmp/styles/'  
    }]  
  }  
},
```

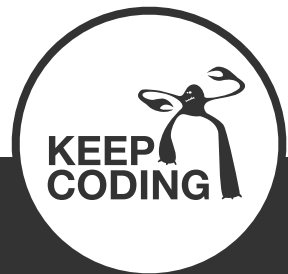




# ■ grunt-contrib-filerev

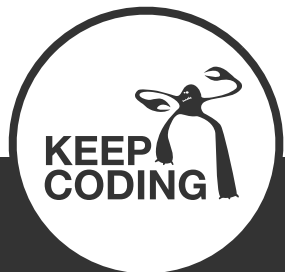


Versiona los recursos



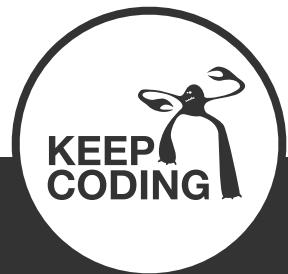
# ■ grunt-filerev

```
// Renames files for browser caching purposes
filerev: {
  dist: {
    src: [
      '<%= appConfig.dist %>/{,**/}* .js',
      '<%= appConfig.dist %>/styles/{,*/*}.css',
      '<%= appConfig.dist %>/images/**/*.{png,jpg,jpeg,gif,webp,svg}',
      '<%= appConfig.dist %>/styles/fonts/*'
    ]
  }
},
```





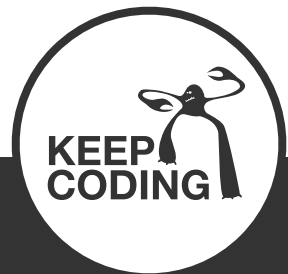
Ofusca y minimiza archivos Javascript



# ■ grunt-contrib-concat



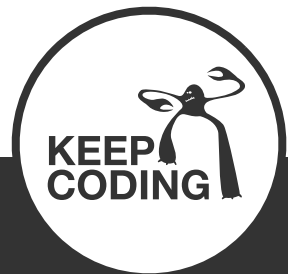
Concatena archivos



# ■ grunt-contrib-cssmin

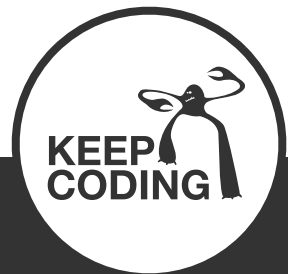


Minimiza y comprime CSS



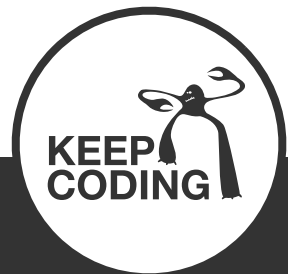
# ■ grunt-contrib-htmlmin

Minimiza HTML



# ■ grunt-contrib-imagemin

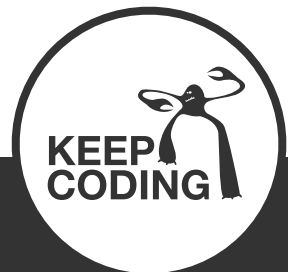
Reduce el peso de imágenes



# ■ grunt-contrib-usemin

Coordina los siguientes plugins

- concat
- uglify
- cssmin
- filerev





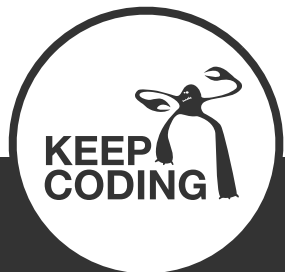
# ■ grunt-contrib-usemin

```
// Reads HTML for usemin blocks to enable smart builds that automatically
// concat, minify and revision files. Creates configurations in memory so
// additional tasks can operate on them
useminPrepare: {
  html: '<%= appConfig.dist %>/index.html',
  options: {
    dest: '<%= appConfig.dist %>',
    flow: {
      html: {
        steps: {
          js: ['concat', 'uglifyjs'],
          css: ['cssmin']
        },
        post: {}
      }
    }
  }
},
```



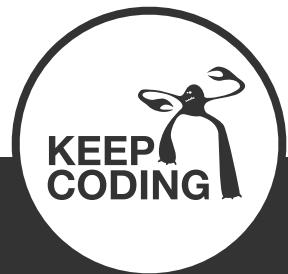
# ■ grunt-contrib-usemin

```
// Performs rewrites based on filerev and the useminPrepare configuration
usemin: {
  html: ['<%= appConfig.dist %>/{,**/}*.*html'],
  css: ['<%= appConfig.dist %>/styles/{,*/*}*.css'],
  options: {
    assetsDirs: ['<%= appConfig.dist %>', '<%= appConfig.dist %>/images']
  }
},
```



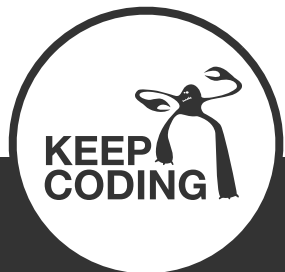


## Plugin para Karma Test Runner



# ■ grunt build task

```
grunt.registerTask('build', [  
  'clean:dist',  
  'wiredep',  
  'includeSource:dist',  
  'useminPrepare',  
  'sass',  
  'autoprefixer',  
  'concat:generated',  
  'copy:dist',  
  'cssmin',  
  'uglify',  
  'filerev',  
  'usemin',  
  'htmlmin'  
]);
```



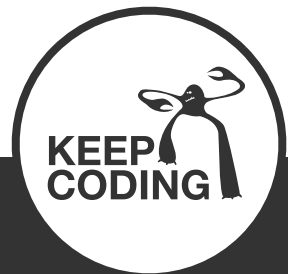
# ■ grunt serve task

```
grunt.registerTask('serve', 'Compile then start a connect web server', function(target) {  
  if (target === 'dist') {  
    return grunt.task.run(['build', 'connect:dist:keepalive']);  
  }  
  
  grunt.task.run([  
    'clean:server',  
    'wiredep',  
    'includeSource:server',  
    'sass',  
    'autoprefixer',  
    'connect:livereload',  
    'watch'  
  ]);  
});
```



# ■ grunt task

```
grunt.registerTask('default', [  
    'jshint',  
    'build'  
]);
```





GRACIAS  
[www.keepcoding.io](http://www.keepcoding.io)

