

# BOWER e GRUNT

AUTOMAÇÃO NO DESENVOLVIMENTO WEB

GABRIEL MANZANO

ALFAMA WEB



# GABRIEL MANZANO

- FORMADO EM TECNOLOGIA EM APLICAÇÕES WEB
- MBA EM ADMINISTRAÇÃO DE BANCO DE DADOS ORACLE
- TRABALHO COM INTERNET HÁ MAIS DE 12 ANOS
- SÓCIO/DIRETOR DE DESENVOLVIMENTO NA ALFAMA WEB

GABRIEL@ALFAMAWEB.COM.BR

FACEBOOK.COM/CAPELOZA



# INSTALANDO O NODEJS

**Link oficial:** <http://nodejs.org>

## **Instalação:**

- Clique em "Install", baixe-o e execute o arquivo.
- Instalação simples, NNF.

**INSTALL**



# Bower

A package manager for  
the web

# O que é?

**O Bower permite gerenciar dependências client-side de uma maneira bastante simples. Em vez de ter que entrar no site de cada projeto que você quer usar e baixar os arquivos necessários, você pode automatizar este processo criando um arquivo de manifesto.**

Link oficial: <http://bower.io>

# Porque utilizar?

**Vamos supor que queria usar uma biblioteca como o “Jquery”**

**Antes:**

- Buscar a última versão do Jquery.
- Baixar o arquivo zipado, extrair e salvar na pasta do seu projeto.
- Se tiver alguma dependência, faça novamente com os outros projetos.
- Depois de alguns dias, verifique se o seu plugin está atualizado.

# Porque utilizar?

## Depois:

- Pelo terminal, ache a pasta do seu projeto e...

```
 bower install jquery --save
```

# Iniciando

Criamos o nosso arquivo de configuração do Bower com o comando

 `bower init`

E vamos responder algumas perguntas...



```
G:\Servidor\www\aw-workshop-grunt-bower>bower init
? name: Workshop Grunt e Bower
? version: 1.0.0
? description: Descrição do nosso projeto.
? main file:
? what types of modules does this package expose?: node
? keywords: workshop, grunt, bower
? authors: Gabriel Manzano <gabriel@alfamaweb.com.br>
? license: MIT
? homepage: (https://github.com/manzano/aw-workshop-grunt-bower) http://www.alfa
? homepage: http://www.alfamaweb.com.br
? add commonly ignored files to ignore list?: No
? would you like to mark this package as private which prevents it from being ac
? would you like to mark this package as private which prevents it from being ac
cidentally published to the registry?: Yes
```

```
{
  name: 'Workshop Grunt e Bower',
  version: '1.0.0',
  homepage: 'http://www.alfamaweb.com.br',
  authors: [
    'Gabriel Manzano <gabriel@alfamaweb.com.br>'
  ],
  description: 'Descrição do nosso projeto.',
  moduleType: [
    'node'
  ],
  keywords: [
    'workshop',
    'grunt',
    'bower'
  ],
  license: 'MIT',
  private: true
}
```

```
? Looks good?: Yes
```

```
G:\Servidor\www\aw-workshop-grunt-bower>
```

# Instalando um pacote

- Pelo terminal, ache a pasta do seu projeto e...

```
 bower install jquery
```

# Instalando um pacote

- Pelo terminal, ache a pasta do seu projeto e...

```
 bower install jquery --save-dev
```

# Instalando um pacote

- Podemos instalar um pacote de diversas maneiras...

```
>- bower install jquery#1.2.3
```

```
>- bower install manzano/aw-workshop-grunt-bower
```

```
>- bower install git://github.com/manzano/aw.git
```

```
>- bower install http://exemplo.com/script.jsv
```

# Desinstalando um pacote

- Pelo terminal, ache a pasta do seu projeto e...



`bower uninstall jquery --save-de`

# Buscando um pacote

- Pelo terminal, ache a pasta do seu projeto e...

 **bower search projeto**

# Atualizando um pacote

- Pelo terminal, ache a pasta do seu projeto e...

```
 bower update
```

```
 bower update jquery
```

# Atualizando um pacote

Há inúmeras maneiras diferentes que você pode usar para especificar quais versões de cada componente você pode usar.

- Versão Específica - 3.1.1
- Somente Versões de Ajuste - ~3.1.1
- Versões Menores - ^3.1.1
- Versões Mais Atualizadas - \*

Mais opções em: <https://github.com/npm/node-semver#ranges>



# Novo projeto

- Fiz uma cópia do meu projeto, sem as dependências, e agora?

 **bower install**

- Você precisa estar na pasta do projeto e com o bower.json.

# Alterando a pasta

- Não queremos mais usar a pasta **bower\_components**, o que faremos?
- Criamos um arquivo `.bowerrc` no diretório do nosso projeto com o conteúdo:

```
{  
  "directory": "meus_componentes/"  
}
```

# Alterando a pasta

- Depois...

 `bower install`

# Done!



# Bower

A package manager for  
the web



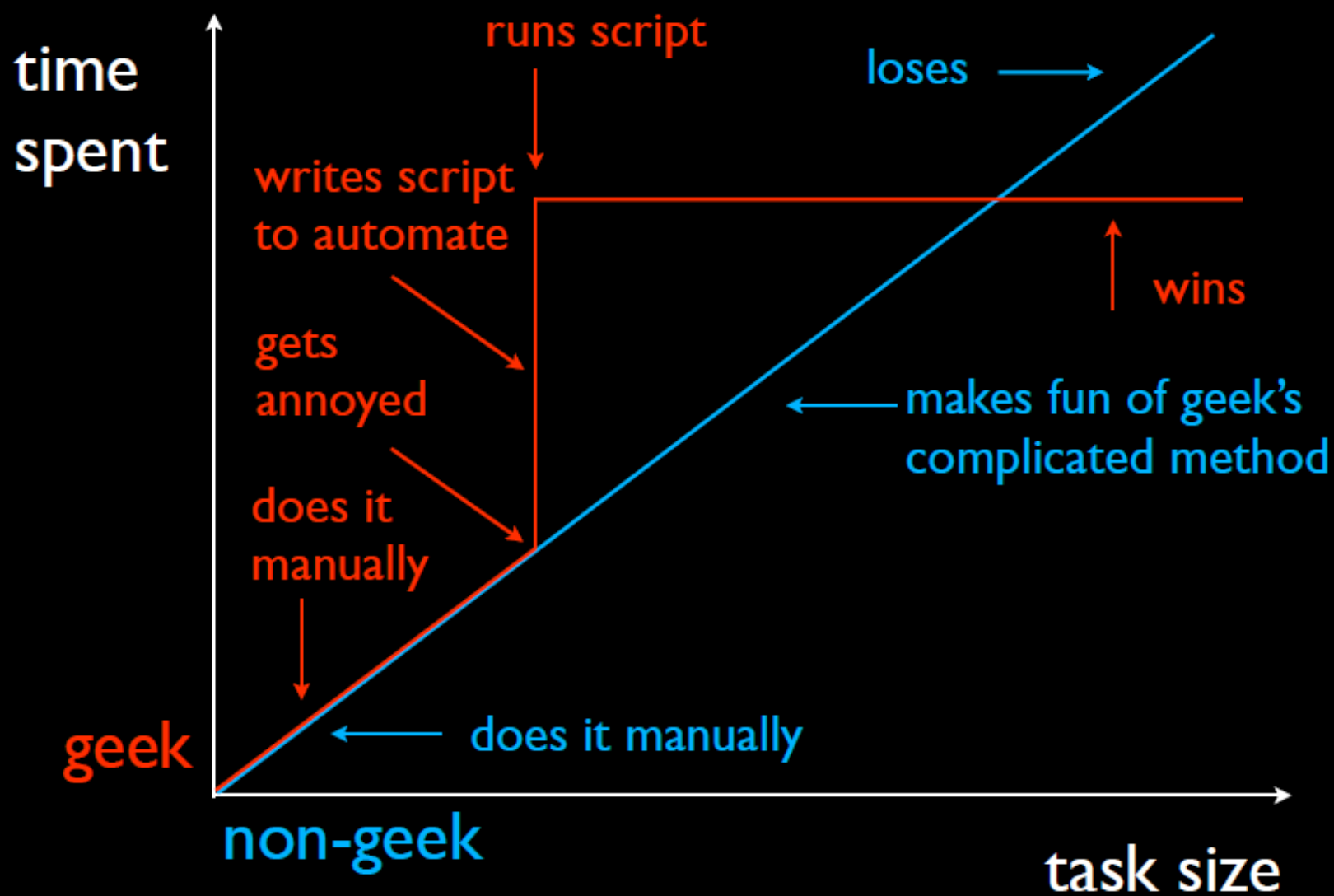
# GRUNT

The JavaScript Task Runner

# O QUE É?

**O Grunt é uma aplicação de linha de comando que tem como objetivo automatizar tarefas.**

Link oficial: <http://gruntjs.com>



# INSTALANDO

**Certifique-se que o Nodejs já está instalado.**  
Pelo terminal...

```
 npm install -g grunt-cli
```

Se você já instalou o Grunt alguma vez, você pode remove-lo no futuro.

```
 npm uninstall -g grunt
```



# NPM INIT

Criamos o nosso arquivo de configuração do NPM com o comando

 `npm init`

E vamos responder algumas perguntas...

```
Press ^C at any time to quit.
name: (aw-workshop-grunt-bower) Workshop
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository: (https://github.com/manzano/aw-workshop-grunt-bower.git)
keywords:
author:
license: (ISC)
About to write to C:\Servidor\www\aw-workshop-grunt-bower\package.json:
```

```
{
  "name": "Workshop",
  "version": "1.0.0",
  "description": "aw-workshop-grunt-bower\r =====",
  "main": "index.js",
  "dependencies": {
    "grunt": "^0.4.5"
  },
  "devDependencies": {},
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "https://github.com/manzano/aw-workshop-grunt-bower.git"
  },
  "author": "",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/manzano/aw-workshop-grunt-bower/issues"
  },
  "homepage": "https://github.com/manzano/aw-workshop-grunt-bower"
}
```

```
Is this ok? (yes) yes
```

# IMPORTANTE

```
>- npm install grunt --save-dev
```

**DOIS ARQUIVOS SÃO IMPORTANTES.**

- **package.json**

Variáveis que você pode usar em seu script.

Dependências do nosso projeto de automação.

- **Gruntfile.js**

# IMPORTANTE

DOIS ARQUIVOS FORAM CRIADOS E SÃO IMPORTANTES.

- **package.json**

Variáveis que você pode usar em seu script.

Dependências do nosso projeto de automação.

- **Gruntfile.js**

```
>- npm install grunt --save-dev
```

# O GRUNTFILE.JS

**Vamos criar o arquivo Gruntfile.js com o conteúdo:**

```
module.exports = function(grunt) {  
  
  grunt.initConfig({  
    pkg: grunt.file.readJSON('package.json'),  
  
  });  
  
};
```

# ESTRUTURA

```
module.exports = function(grunt) {  
  
  // Configurações do nosso projeto  
  grunt.initConfig({  
    pkg: grunt.file.readJSON('package.json'),  
    uglify: {  
      options: {  
        banner: '/*! <%= pkg.name %> <%= grunt.template.today("yyyy-mm-dd") %> */\n'  
      },  
      build: {  
        src: 'desenvolvimento/js/<%= pkg.name %>.js',  
        dest: 'producao/js/<%= pkg.name %>.min.js'  
      }  
    }  
  });  
  
  // Carregando os plugins  
  grunt.loadNpmTasks('grunt-contrib-uglify');  
  
  // Tasks  
  // Padrão é obrigatória  
  grunt.registerTask('default', ['uglify']);  
  
};
```

# VARIÁVEIS

**Utilizando variáveis no Gruntfile.js:**

```
pkg: grunt.file.readJSON('package.json'),
```

```
var variavel = {  
    foo: 'Valor 1',  
    bar: 'Valor 2'  
};
```

Uso: <%= variavel.foo %>

# PLUGINS

**Em nosso workshop iremos utilizar os plugins abaixo:**

- Copy
- Concat
- Unglify
- Watch
- Cssmin
- Imagemin
- Htmlmin
- Load-grunt-taks



# COPY

**Página:** <https://github.com/gruntjs/grunt-contrib-copy>

```
 npm install grunt-contrib-copy --save-dev
```

```
 grunt.loadNpmTasks('grunt-contrib-copy');
```

Configura o plugin no `grunt.initConfig()`;

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

# UGLIFY

**Página:** <https://github.com/gruntjs/grunt-contrib-uglify>

```
 npm install grunt-contrib-uglify --save-dev
```

```
 grunt.loadNpmTasks('grunt-contrib-uglify');
```

Configura o plugin no `grunt.initConfig()`;

```
 grunt.registerTask('default', ['copy', 'uglify']);
```

# CONCAT

**Página:** <https://github.com/gruntjs/grunt-contrib-concat>

```
 npm install grunt-contrib-concat --save-dev
```

```
 grunt.loadNpmTasks('grunt-contrib-concat');
```

Configura o plugin no `grunt.initConfig()`;

```
 grunt.registerTask('default', ['copy', 'concat', 'uglify']);
```


# CSSMIN

**Página:** <https://github.com/gruntjs/grunt-contrib-cssmin>

```
 npm install grunt-contrib-cssmin --save-dev
```

```
 grunt.loadNpmTasks('grunt-contrib-cssmin');
```

Configura o plugin no `grunt.initConfig()`;

```
 grunt.registerTask('default', ['copy', 'concat',  
    'cssmin', 'uglify']);
```


# IMAGEMIN

**Página:** <https://github.com/gruntjs/grunt-contrib-imagemin>

```
 npm install grunt-contrib-imagemin --save-dev
```

```
 grunt.loadNpmTasks('grunt-contrib-imagemin');
```

Configura o plugin no `grunt.initConfig()`;

```
 grunt.registerTask('default', ['copy', 'concat',  
    'cssmin', 'imagemin', 'uglify']);
```


# HTMLMIN

**Página:** <https://github.com/gruntjs/grunt-contrib-htmlmin>

```
 npm install grunt-contrib-htmlmin --save-dev
```

```
 grunt.loadNpmTasks('grunt-contrib-htmlmin');
```

Configura o plugin no `grunt.initConfig()`;

```
 grunt.registerTask('default', ['copy', 'concat',  
    'cssmin', 'imagemin', 'htmlmin', 'uglify']);
```

# WATCH

**Página:** <https://github.com/gruntjs/grunt-contrib-watch>

```
 npm install grunt-contrib-watch --save-dev
```

```
 grunt.loadNpmTasks('grunt-contrib-watch');
```

Configura o plugin no `grunt.initConfig()`;

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

# LOAD-GRUNT-TASKS

Página: <https://github.com/sindresorhus/load-grunt-tasks>

```
 npm install load-grunt-tasks --save-dev
```

 ~~grunt.loadNpmTasks('xxxxxxx');~~

~~Configura o plugin no grunt.initConfig();~~

 require('load-grunt-tasks')(grunt);



# OBRIGADO!

Gabriel Manzano

[gabriel@alfamaweb.com.br](mailto:gabriel@alfamaweb.com.br)

[alfamaweb.com.br](http://alfamaweb.com.br)