



**DIO**  
CEZAR  
**GO**

# **Automatizadores para Web**



# AGENDA

---

- o que são automatizadores
- grunt
- npm scripts
- gulp
- plugins do gulp

# O QUE SÃO?

- *task runners* são automatizadores de tarefas;
- ajudam bastante nas tarefas manuais do dia a dia;
- de uma forma bem resumida... um projeto web precisa passar por algumas otimizações antes de ir para produção;
- como por exemplo:
  - minificação do CSS;
  - otimização de imagens;
  - unificação e/ou ofuscação de JavaScripts;

# O QUE SÃO?

- essas otimizações são extremamente importantes para obter um melhor ranking de SEO do site em motores de busca;
- entre algumas métricas, a velocidade de carregamento conta muito;
- essas ações ajudam neste processo;

# O QUE SÃO?

- problema:
  - é impossível escrever nossos códigos já de forma minificada;
  - não recebe as imagens já otimizadas;
- solução:
  - utilizar alguma ferramenta que faça isso de forma automática por você;

# O QUE SÃO?

- outro fator: compilação de arquivos;
- utilização de pré-processadores em CSS, como SASS ou LESS, ou ainda um transpiler de JavaScript como Babel ou CoffeScript;
- nestes casos, você de alguma magia que transforme os arquivos originais nos arquivos compilados;

# GRUNT

---

- esse foi o primeiro task runner a ganhar destaque na comunidade JavaScript;
- ele é simples e as tarefas são definidas em um arquivo de configuração;
- essa simplicidade pode incomodar quando se precisa de um fluxo mais complexo;
- possui vários plugins → <https://gruntjs.com/plugins>
- pode ser encontrado em → <https://gruntjs.com/>

# NPM SCRIPTS

---

- como já vimos com o NPM podemos criar scripts customizados para realizar determinadas ações;
- podemos utilizar webpack e browserify para otimização de algumas tarefas pelos próprios scripts npm;



# OUTRAS ALTERNATIVAS

---

- <http://broccolijs.com/> e <http://brunch.io/> são algumas dessas opções, além de opções muito recente, que pouquíssimas pessoas usam ou já ouviram falar, como o <https://github.com/lukeed/taskr>.

# **GULP**

---

- o **gulp** não é uma ferramenta nova, porém ganhou destaque nos últimos anos;
- acabou passando o Grunt, porém não tem a mesma quantidade de plugins;
- existem 2 grandes diferenciais entre o Grunt e o gulp: a definição de suas tasks e como elas são executadas;
  - as tarefas são executadas em modo async e em memória, tornando ele mais rápido que o grunt;
- além disso, as tarefas são definidas como código, o que permite que você programe suas tarefas;

# GULP

## INSTALANDO O GULP

```
$npm install -g gulp
```

## VERIFICAR A VERSÃO INSTALADA

```
$gulp -v
```

# GULP

---

- toda a automação que criaremos ser reunida em um único arquivo chamado *gulpfile.js*
- a localização do arquivo pode variar de acordo com seu projeto, mas você poderá informar as localizações dos arquivos que deseja automatizar;
- normalmente o gulpfile é colocado na raiz do projeto;

# GULP

---

- agora precisamos definir quais serão os plugins a serem utilizados com o gulp;
- você pode buscar os plugins e realizar a instalação como consta na página de instruções;
- e ainda será necessário instalar o próprio gulp no seu projeto;

## INSTALANDO PLUGINS PARA O GULP

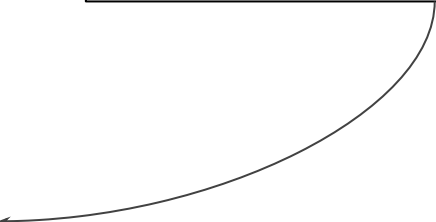
```
$npm install gulp gulp-concat gulp-cssmin gulp-imagemin  
gulp-sass gulp-svgto gulp-uglify --save-dev
```

# ESTRUTURA DO ARQUIVO GULP

---

```
Gulp = {  
  self: null,  
  sass: null,  
  init: function(){  
    Gulp.self = require('gulp');  
    Gulp.sass = require('gulp-sass');  
    Gulp.default();  
  },  
  default: function(){  
    Gulp.self.task('default', ['scss']);  
  },  
  scss: function(){  
    Gulp.self.task('scss', function(){  
      Gulp.self.src('style.scss')  
        .pipe(Gulp.sass())  
        .pipe(Gulp.self.dest('style.css'));  
    });  
  }  
}; Gulp.init();
```

JavaScript, então podemos organizar  
em um objeto literal



# UTILIZANDO SVG OPT

---

```
Gulp = {
  self: null,
  sass: null,
  init: function(){
    Gulp.self = require('gulp');
    Gulp.sass = require('gulp-sass');
    Gulp.default();
  },
  default: function(){
    Gulp.self.task('default', ['svgopt']);
  },
  svgopt: function(){
    Gulp.self.task('svgopt', function(){
      console.log('[SVG0] Otimizando arquivos SVG');
      Gulp.self.src('./assets/images/**/*.svg')
        .pipe(Gulp.svggo())
        .pipe(Gulp.self.dest('./assets/images'));
    });
  }
}; Gulp.init();
```

# UTILIZANDO SASS

---

```
Gulp = {
  self: null,
  sass: null,
  init: function(){
    Gulp.self = require('gulp');
    Gulp.sass = require('gulp-sass');
    Gulp.default();
  },
  default: function(){
    Gulp.self.task('default', ['scss']);
  },
  scss: function(){
    Gulp.self.src('./assets/scss/pages/**/*.scss')
      .pipe(Gulp.sass())
      .pipe(Gulp.self.dest('./assets/css/src/pages'));
  }
}; Gulp.init();
```



# UTILIZANDO CSSMIN

---

```
Gulp = {
  self: null,
  sass: null,
  init: function(){
    Gulp.self = require('gulp');
    Gulp.sass = require('gulp-sass');
    Gulp.default();
  },
  default: function(){
    Gulp.self.task('default', ['css']);
  },
  css: function(){
    Gulp.self.src('./assets/css/src/**/*.css')
      .pipe(Gulp.cssmin())
      .pipe(Gulp.self.dest('./assets/css/dist'));
  }
}; Gulp.init();
```

# UTILIZANDO UGLIFY

---

```
Gulp = {
  self: null,
  sass: null,
  init: function(){
    Gulp.self = require('gulp');
    Gulp.sass = require('gulp-sass');
    Gulp.default();
  },
  default: function(){
    Gulp.self.task('default', ['js']);
  },
  js: function(){
    Gulp.self.src(['./js/src/**/*.js', '!js/dist/**'])
      .pipe(Gulp.uglify())
      .pipe(Gulp.self.dest('./js/dist'))
  }
}; Gulp.init();
```

# UTILIZANDO CONCAT

---

```
Gulp = {  
  self: null,  
  sass: null,  
  init: function(){  
    Gulp.self = require('gulp');  
    Gulp.sass = require('gulp-sass');  
    Gulp.default();  
  },  
  default: function(){  
    Gulp.self.task('default', ['jslibs']);  
  },  
  jslibs: function(){  
    Gulp.self.src(['./js/src/jquery/*.js', './js/src/libs/**/*.js'])  
      .pipe(Gulp.uglify())  
      .pipe(Gulp.concat('all.js'))  
      .pipe(Gulp.self.dest('./js/src/all/'))  
  }  
}; Gulp.init();
```

# UTILIZANDO WATCH

---

com o watch consegue inspecionar permanentemente qualquer alteração em seus arquivos, e a cada nova alteração invocar uma funcionalidade definida no gulp

```
Gulp.self.task('watch', function(){  
  Gulp.self.watch('./js/**/*.js', ['js', 'jslibs']);  
  Gulp.self.watch('./assets/css/src/**', ['css']);  
  Gulp.self.watch('./assets/scss/**', ['scss']);  
  Gulp.self.watch('./assets/images/**/*.svg', ['svgo']);  
});
```

# EXERCÍCIO

---

- crie a seguintes automatizações:
  - minificação dos arquivos js;
  - minificação dos arquivos css;
  - watch para modificação nestes arquivos;

# **MATERIAIS COMPLEMENTARES**

---

- <https://goo.gl/qX8L69>
- <https://goo.gl/AtK6pG>